Research Report: A Multi-modal AI System for Enhanced Breast Cancer Risk Prediction and Detection

Abstract

Breast cancer remains a significant global health challenge, with early and precise risk assessment being paramount for effective intervention and improved patient prognosis. This research details the development and architecture of a novel multi-modal Breast Cancer Risk Prediction Tool. The system is engineered to provide a holistic risk profile by synergistically integrating patient-derived questionnaire data, advanced mammographic image analysis, and molecular-level gene expression data. Methodologically, it employs a diverse suite of machine learning models for quantitative image and genetic analysis, including traditional feature-based approaches (HOG, LBP, SIFT-BoVW), deep learning architectures (ResNet), and logistic regression for gene data. Furthermore, an innovative aspect involves the exploration of reinforcement learning using **Stable Baselines3 and Gymnasium** for **automating the hyperparameter tuning** of these individual models, aiming to enhance their performance. A Large Language Model (Llama 3 via Groq API) is integrated for the automated generation of comprehensive, narrative medical reports. The system is architected with a Python Flask backend, a web-based frontend for user interaction, and an SQLite database for robust data persistence and management. This research presents a comprehensive decision support framework aimed at augmenting clinical assessment in breast cancer risk stratification.

Keywords: Breast Cancer, Risk Prediction, Multi-modal Data Integration, Machine Learning, Deep Learning, Mammography Analysis, Gene Expression, Reinforcement Learning, Hyperparameter Tuning, Large Language Model, Medical Report Generation, Decision Support System.

1. Introduction

1.1 Background and Significance

Breast cancer represents a formidable global health challenge, standing as a leading cause of cancer-related morbidity and mortality among women worldwide. The profound impact of this disease necessitates continuous advancements in diagnostic and prognostic methodologies. Crucially, the efficacy of treatment and patient survival rates are significantly enhanced by early detection and accurate risk assessment. Identifying individuals at an elevated risk allows for the implementation of tailored screening strategies, the adoption of proactive preventive measures, and the initiation of timely therapeutic interventions. This proactive approach plays a pivotal role in reducing the overall burden of the disease on both individuals and healthcare systems.[1] The fundamental premise of this tool is rooted in the established clinical benefit of early detection, suggesting that its ultimate utility is intrinsically linked to its capacity for identifying at-risk individuals earlier and more accurately. This also sets the stage for the ethical considerations later in this report, as the potential benefit is high, but so is the responsibility associated with such a powerful diagnostic aid.

1.2 Problem Statement and Gaps in Existing Approaches

Traditional breast cancer risk assessment often relies on a limited array of data sources, typically including demographic information, family history, and basic clinical examination findings. While these approaches offer valuable insights, they frequently fail to capture the full spectrum of complex risk factors, potentially leading to suboptimal risk stratification. Mammography, a cornerstone of breast cancer screening, provides critical visual information but can be further enhanced through sophisticated computational analysis. Concurrently, molecular markers derived from gene expression profiles offer deeper biological insights into an individual's predisposition and disease characteristics. There is a pressing need for integrated systems capable of assimilating these diverse data modalities to provide a more accurate and comprehensive risk profile. Moreover, a significant challenge in developing high-performing machine learning models lies in optimizing their performance through extensive hyperparameter tuning. While various automated methods exist, such as grid search, random search, and Bayesian optimization, the application of reinforcement learning to this specific problem within the context of multi-modal medical data remains an active area of exploration.[1] The inherent incompleteness of existing, often unimodal, methods necessitates the proposed multi-modal approach. This system aims to bridge these information gaps, establishing a direct link between data integration and improved accuracy. The specific mention of reinforcement learning for hyperparameter tuning as an "active area of exploration" further signals its novelty and positions this project at the forefront of current research.

1.3 Proposed Multi-modal Solution and Core Contributions

This research proposes a multi-modal "Breast Cancer Risk Prediction Tool" designed as an advanced clinical decision support system. The primary contributions of this work are multifaceted and aim to overcome the limitations identified in existing approaches:

- **Multi-modal Integration:** The system establishes a unified framework capable of processing and integrating disparate data types, including patient questionnaires, mammographic images, and gene expression data. This comprehensive approach is a core architectural and methodological contribution, allowing for a more complete understanding of an individual's risk profile.[1]
- **Advanced Machine Learning:** The tool employs a comprehensive suite of machine learning techniques tailored for both image and genetic analysis. For mammographic images, it utilizes Histogram of Oriented Gradients (HOG), Local Binary Patterns (LBP), Scale-Invariant Feature Transform (SIFT) with Bag of Visual Words (BoVW), and Residual Networks (ResNet). For gene expression data, logistic regression is applied. The diversity of models reflects a robust strategy for capturing various features and patterns within the data.[1]
- **Reinforcement Learning for Hyperparameter Tuning:** A significant technical contribution involves the exploration of a Reinforcement Learning (RL) approach. Using the Stable Baselines3 library and custom Gymnasium environments, the system automates and optimizes the hyperparameter tuning process for individual image and gene expression analysis models. This methodology is designed to enhance model performance by intelligently navigating complex hyperparameter spaces.[1]
- **Automated LLM-Powered Reporting:** The system incorporates a state-of-the-art Large Language Model (Llama 3, accessed via the Groq API) to automatically generate comprehensive, narrative medical reports. This capability facilitates clinical interpretation by transforming complex analytical outputs into coherent, natural language summaries, which can significantly streamline clinical workflows.[1]
- **Comprehensive System Architecture:** A robust and user-friendly application has been developed, featuring a Python Flask backend, a web-based interface for user interaction, and an SQLite database for efficient data handling and persistent storage. This architectural design underscores the practical and deployable nature of the prototype.[1]

These enumerated contributions collectively demonstrate a holistic approach to a complex medical challenge, combining diverse artificial intelligence paradigms to address both predictive accuracy and clinical interpretability. The combination of traditional machine learning, deep learning, reinforcement learning, and large language models is not merely a collection of techniques but a strategic layering designed to achieve robust, optimized, and interpretable results. This multi-faceted approach is intended to overcome the inherent limitations of single-modality or single-technique systems.

1.4 Report Structure

This report is organized to provide a comprehensive overview of the Breast Cancer Risk Prediction Tool. Section 2, Materials and Methods, details the data modalities, system architecture, data preprocessing techniques, and the development and tuning of machine learning models. Section 3 describes the System Implementation and Workflow, outlining the operational aspects. Section 4 presents the key Results and System Capabilities, showcasing the prototype's functionalities. Section 5 provides a Discussion of the findings, advantages, limitations, and clinical implications. Section 6 offers a Conclusion summarizing the research. Section 7 outlines Future Work and Directions, identifying avenues for further development. Finally, Section 8 briefly touches upon Ethical Considerations inherent in the development and deployment of such a tool.[1]

2. Materials and Methods

2.1 Data Sources and Acquisition

The system integrates three primary data modalities to construct a comprehensive patient risk profile:

- **Patient Questionnaire Data:** Clinical and demographic information is collected through a structured web-based form, implemented via the index.html frontend. This includes essential patient details such as demographics, family history of breast cancer, personal medical history, and relevant lifestyle factors. This data provides crucial clinical context for the risk assessment.[1]
- **Mammographic Image Data:** Digital mammogram images are uploaded by the user, and the system is designed to process standard image formats. This image data is fundamental for visual feature extraction and deep learning-based analysis. The primary dataset utilized for training and evaluating the image

analysis models is the Curated Breast Imaging Subset of DDSM (CBIS-DDSM). This dataset is accessible via The Cancer Imaging Archive (TCIA) and is further referenced through Google Dataset Search and the University of Central Florida's Complex Adaptive Systems Laboratory.[1] CBIS-DDSM is a highly regarded, updated, and standardized version of the Digital Database for Screening Mammography. It comprises 10,239 scanned film mammography studies from 6,671 subjects, totaling approximately 163.6 GB of data. The dataset includes cases categorized as normal, benign, and malignant, all with verified pathology information, detailed Region of Interest (ROI) segmentations, and standardized train/test splits. These characteristics make CBIS-DDSM an ideal choice for robust model training and evaluation in a medical imaging context.[1] Metadata within the RRP-Breast_Cancer_Detection.ipynb Jupyter Notebook indicates the use of Kaggle dataset IDs 17860 and 1115384, which may represent supplementary data, pre-processed versions of CBIS-DDSM, or data used for specific sub-tasks or initial experiments within the notebook's workflow.[1] The selection of CBIS-DDSM for imaging data represents a strategic choice for scientific rigor, providing a strong foundation for training the image models and enhancing their potential for generalizability.
- **Gene Expression Data:** Gene expression profiles are accepted by the system as CSV or TSV files, providing molecular-level information crucial for deeper biological insights. For the development and demonstration of the gene expression module, the GSE1000_series_matrix.txt file, a GEO Series Matrix file, was utilized. The parsing of this data is handled by a dedicated load_geo_matrix function within the system.[1] The project's long-term objective is to utilize gene expression data derived from patient cohorts, ideally linked to corresponding imaging and clinical data. This integration would enable the identification of molecular subtypes (e.g., Luminal A, Luminal B, HER2-enriched, Basal-like) and prognostic gene signatures, as detailed in seminal works such as Bao and Davidson (2008).[1] The current use of "example labels" for the gene expression data, while demonstrating the capability of integration, reveals a current limitation that necessitates explicit attention. While the architecture supports gene expression data, the model's performance in a real clinical scenario for breast cancer risk prediction cannot be inferred from its current state. This highlights a critical distinction between proof-of-concept and clinical readiness, emphasizing the need for rigorous clinical data acquisition and validation for this specific module in future work.

2.2 System Architecture

The application employs a robust client-server architecture, designed for modularity, efficiency, and scalability.

Technological Framework:

- **Backend:** The core logic of the system is developed in Python, leveraging the Flask web framework. The primary backend script, server.py, manages requests, orchestrates data processing, and interacts with the database and machine learning models.[1] Flask provides a lightweight and flexible foundation for the web server.
- **Frontend:** The user interface is built using standard web technologies: HTML for structure (found in the templates/ directory), CSS for styling (static/style.css), and JavaScript for interactive elements (static/script.js). This combination provides a user-friendly and responsive web interface for patient data input and report display.[1]
- **Database:** An SQLite database, bcrrp_data.db, is utilized for robust data persistence and management. Database operations, including storing patient records, questionnaire responses, prediction outcomes, and generated reports, are managed via db_utils.py.[1] While suitable for a prototype or single-user deployment, SQLite's efficiency for querying and managing structured metadata is a key advantage.
- **Environment Management:** Sensitive information, such as API keys (e.g., for the Groq API), is securely managed using a .env file. This adheres to standard security practices for handling credentials in development and deployment environments.[1]
- **Development Environment:** The initial model development and experimentation were conducted within a Jupyter Notebook environment, specifically RRP-Breast_Cancer_Detection.ipynb. This choice facilitated an iterative and exploratory development process for the machine learning components.[1]

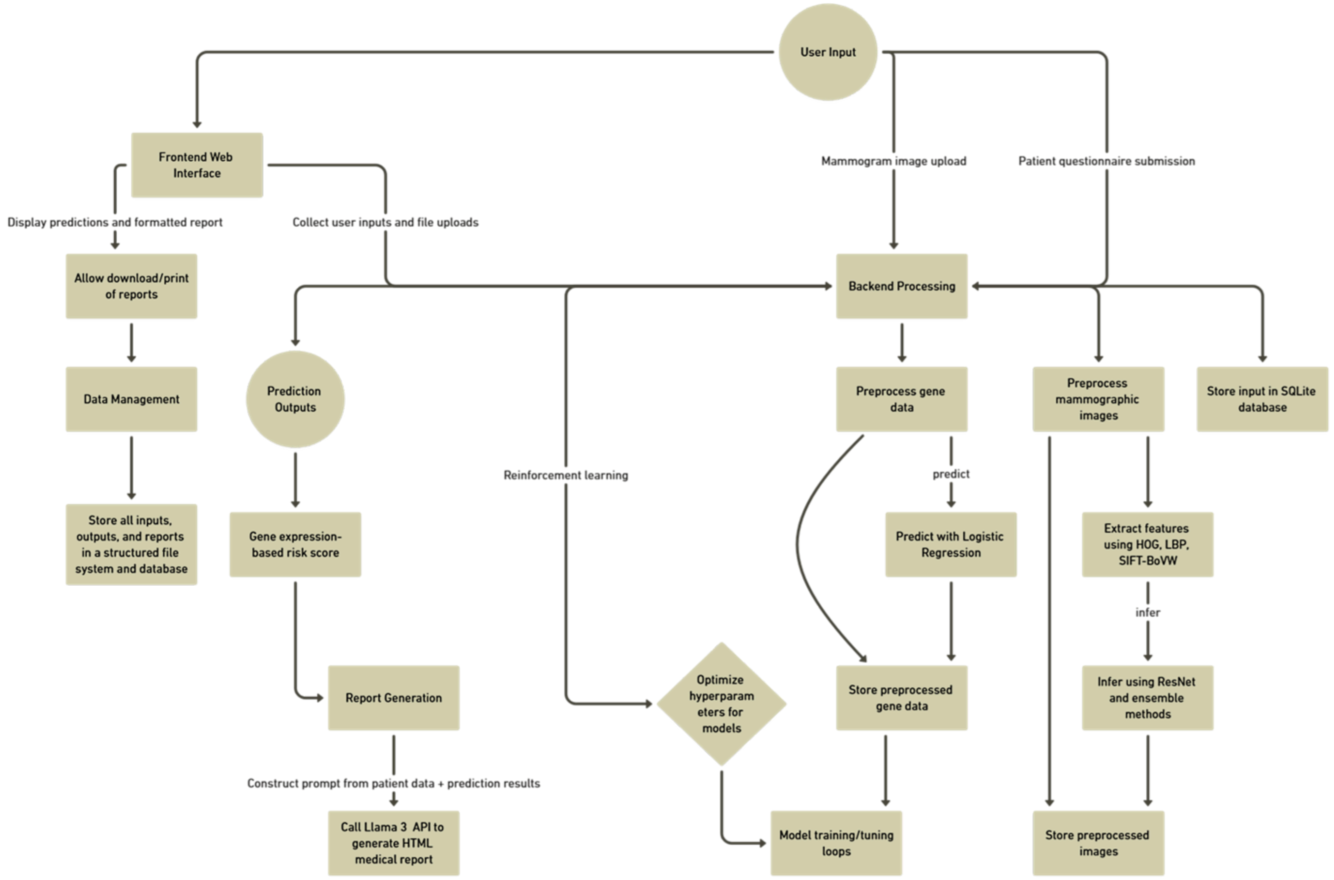
Directory Structure Insights: The provided directory structure (Image 1, Image 2) offers a visual representation of the system's modularity and the organization of its components.

- The Output_Files/ directory contains various subdirectories for image preprocessing outputs, such as Hog_Images, LBP_Images, SIFT_Images, AHistogram_Images, Negative_Images, and merged_images. This organization clearly indicates the extensive preprocessing and visualization capabilities built into the system. The models subdirectory within Output_Files suggests that intermediate or final processed images might also be stored for specific models or for debugging and understanding model inputs.

- The RL_Models/ directory explicitly confirms the successful implementation and persistence of reinforcement learning-tuned models and their training artifacts. It contains gene_model_best_rl, gene_model_logs, gene_expression_rl_agent.zip, and gene_expression_model_tuned.joblib, demonstrating that RL tuning is not an afterthought but a central, integrated component of the model optimization pipeline, suggesting a commitment to performance enhancement beyond initial training.
- The presence of static/ and templates/ directories directly confirms the structure of the frontend components.
- The models/ directory houses the trained machine learning models, saved in .h5 format for Keras/TensorFlow models and .joblib for Scikit-learn models. This includes models for HOG, LBP, SIFT, ResNet, and gene expression, along with their preprocessing variants (AHE, Negative, N) and the SIFT K-Means model. This organization directly supports the "Advanced Machine Learning" contribution by providing a centralized repository for all trained predictive components.
- Core Python scripts such as server.py, inference_app.py, train_gene_model.py, rl_hyperparameter_tuning.py, llm_utils.py, db_utils.py, create_sift_kmeans.py, and requirements.txt are clearly visible. These files directly map to the functionalities described in the research report, underscoring the system's modular and well-defined architecture.[1] The
  pycache directories further indicate active Python development.

The directory structure is not merely a list of files but a concrete manifestation of the system's modularity, the complexity of its preprocessing, the distinct phases of model development (training, tuning), and the persistence of various artifacts. This modularity is a critical engineering strength, facilitating maintenance, debugging, and future expansion. For instance, the dedicated RL_Models folder implies that RL tuning is a central, integrated component of the model optimization pipeline.

System Architecture Diagram


User Input
Frontend Web Interface
Mammogram image upload
Patient questionnaire submission
Display predictions and formatted report
Collect user inputs and file uploads
Allow download/print of reports
Backend Processing
Data Management
Prediction Outputs
Preprocess gene data
Preprocess mammographic images
Store input in SQLite database
Reinforcement learning
predict
Store all inputs, outputs, and reports in a structured file system and database
Gene expression-based risk score
Predict with Logistic Regression
Extract features using HOG, LBP, SIFT-BoVW
infer
Report Generation
Optimize hyperparameters for models
Store preprocessed gene data
Infer using ResNet and ensemble methods
Construct prompt from patient data + prediction results
Call Llama 3 API to generate HTML medical report
Model training/tuning loops
Store preprocessed images

2.3 Data Preprocessing

Effective data preprocessing is a critical step in preparing raw data for consumption by machine learning models, ensuring consistency, quality, and optimal feature representation.

2.3.1 Mammographic Image Preprocessing (inference_app.py)

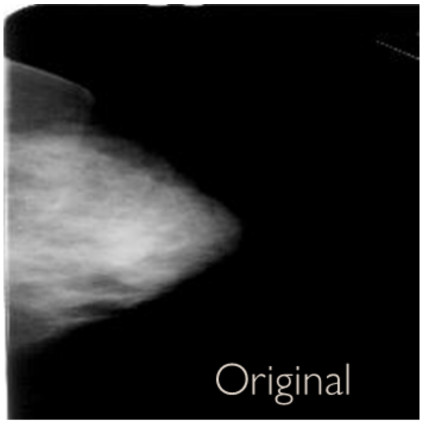
Mammographic image preprocessing is primarily performed using the OpenCV (cv2) and Scikit-image libraries.[1] These steps are crucial for standardizing images and enhancing features relevant for analysis.

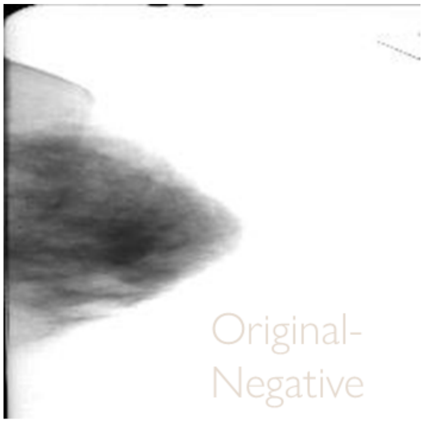
- **Resizing:** Images are uniformly resized to standardized dimensions. FEATURE_IMG_SIZE is used for feature extraction methods like HOG, LBP, and SIFT, while RESNET_IMG_SIZE is tailored for deep learning models such as ResNet.

This ensures consistent input dimensions across different model architectures.[1]

- **Contrast Enhancement:** Adaptive Histogram Equalization (AHE) is applied as a variant preprocessing step. AHE enhances local contrast, making subtle features and abnormalities within the mammogram more discernible. This technique leads to the development of specific model variants, denoted by *AHE_model.h5, which are trained on these contrast-enhanced images.[1] The use of AHE helps improve the visibility of important image structures, which can be crucial for detection algorithms.[4]
- **Image Inversion:** Negative images are generated and utilized as another preprocessing variant. This transformation creates an alternative representation of the mammogram, which might offer complementary information or be beneficial for certain feature detectors or deep learning models. Models trained on these inverted images are denoted by *N_model.h5.[1]
- **Normalization:** Pixel values are typically scaled, for instance, to a range of or standardized (zero mean, unit variance), as required by the specific machine learning models. This is a standard practice to prevent features with larger numerical ranges from disproportionately influencing model training.[1]

The application of multiple preprocessing variants, such as AHE and image inversion, for the same base image highlights a deliberate strategy to enhance model robustness and capture diverse visual information. Different preprocessing techniques emphasize distinct image characteristics (e.g., AHE for local contrast, inversion for intensity reversal). Training models on these variants allows them to learn from diverse representations, suggesting an implicit ensemble at the preprocessing level. This approach contributes to the overall robustness of the image analysis pipeline, especially when combined with the explicit ensemble methods applied later in the system.





2.3.2 Gene Expression Data Preprocessing (train_gene_model.py, inference_app.py)

Preprocessing of gene expression data is managed using the Pandas and NumPy libraries, ensuring the data is in a suitable format for machine learning analysis.[1]

- **Data Transposition and Cleaning:** The raw gene expression data is structured such that samples are represented as rows and genes as columns. This involves transposing the data if necessary, followed by numeric conversion and handling of NaN (Not a Number) values. This structuring ensures the data aligns with the input requirements of tabular machine learning models.[1]
- **Feature Scaling:** StandardScaler from Scikit-learn is applied to normalize gene expression values. This process scales the data to have a mean of zero and a standard deviation of one, which is crucial for many machine learning algorithms, particularly those sensitive to the magnitude of feature values. The trained scaler is persisted using Joblib (models/gene_expression_scaler.joblib).[1]
- **Feature Alignment:** During the inference phase, uploaded gene expression data columns are meticulously aligned with the expected features. These expected features are stored in models/gene_feature_names.joblib, which was generated during the training phase. This alignment is vital to ensure that the model receives features in the same order and format as it was trained on, thereby preventing errors during prediction.[1]

The persistence of the scaler and feature names (gene_expression_scaler.joblib, gene_feature_names.joblib) is a critical aspect of the system's robust deployment strategy. It ensures that the exact same scaling transformation and feature order applied during model training are consistently applied to new, unseen data during inference. Without this consistency, even a perfectly trained model would yield incorrect predictions if the input data's scale or feature order differed. This practice ensures the reliability and reproducibility of predictions in a production-like environment, demonstrating a mature approach to model deployment.

2.4 Machine Learning Model Development and Tuning

The system incorporates a sophisticated approach to machine learning, developing distinct models for image analysis and gene expression analysis. All trained models and scalers are systematically saved using Joblib or Keras's H5 format within the models/ directory, reflecting a structured and organized approach to model management. A key innovative aspect of this project is the exploration of a Reinforcement Learning (RL) approach for hyperparameter tuning, which is applied to these models.[1]

2.4.1 Image Analysis Models

A diverse set of models are employed for mammographic image analysis, indicating a comprehensive and experimental approach, likely derived from extensive exploration within the RRP-Breast_Cancer_Detection.ipynb Jupyter Notebook. These models are implemented using a combination of TensorFlow/Keras and Scikit-image.[1]

Traditional Feature-Based Models: These models focus on extracting interpretable, hand-crafted features from images.

- **Histogram of Oriented Gradients (HOG):** HOG models, including Hog_model.h5, HogAHE_model.h5, and HogN_model.h5, extract features using Scikit-image. HOG is a powerful feature descriptor that counts occurrences of gradient orientation in localized portions of an image. It divides an image into small cells, computes a histogram of gradient directions within each cell, and then normalizes local contrast across overlapping blocks. This process effectively captures edge and shape information, which is crucial for identifying anatomical structures and abnormalities in medical images. HOG is robust to geometric transformations and lighting variations, making it particularly effective for analyzing the wide variability encountered in real-world medical images.[1]
- **Local Binary Patterns (LBP):** LBP models, such as LBP_model.h5, LBPAHE_model.h5, and LBPN_model.h5, also extract features using Scikit-image. LBP is a highly efficient texture descriptor that characterizes the local texture of an image by comparing a pixel's intensity to its neighbors. The resulting binary patterns are then used to construct histograms that represent the texture of regions. This approach is effective for texture classification and has been successfully applied to mammographic images to characterize regional texture patterns of masses, aiming to reduce false-positive detections.[1]
- **Scale-Invariant Feature Transform (SIFT) with Bag of Visual Words (BoVW):** SIFT-based models, including Sift_model.h5, SiftAHE_model.h5, and SiftN_model.h5, utilize SIFT descriptors. SIFT is a robust feature detector that identifies keypoints in an image that are invariant to scale, rotation, translation, illumination, and blur.[9] These SIFT descriptors are then clustered using a K-Means model (
sift_kmeans_model.joblib, trained by create_sift_kmeans.py) to create a visual

vocabulary, or "codebook." Images are then represented as histograms of these "visual words," effectively compressing their descriptions into a set of visual word IDs. This method allows for the representation of similar features across different images using a shared vocabulary.[1]

Deep Learning Models:

- **Residual Networks (ResNet):** ResNet models, specifically resnet_model.h5, resnetAHE_model.h5, and resnetN_model.h5, are implemented using TensorFlow/Keras. ResNet architectures are highly effective for image classification tasks, particularly in medical imaging, due to their ability to learn complex, hierarchical features automatically. These models were likely pre-trained on large image datasets (e.g., ImageNet) and then fine-tuned on mammography datasets, or trained from scratch if a sufficiently large specialized dataset was available.[1] Studies have demonstrated that ResNet-50 can achieve high performance, with reported accuracies of 96% and F1 scores of 94.66% on Digital Database for Screening Mammography (DDSM) datasets for breast cancer detection and classification.[11]

Training Data (Image Models): The primary dataset for training these image models is the Curated Breast Imaging Subset of DDSM (CBIS-DDSM), a well-curated and standardized resource for mammography research.[1]

Training Process (Image Models): The training process for image models typically involves several stages: initial image preprocessing, feature extraction (for HOG, LBP, SIFT), definition of the model architecture (especially for ResNet), training using appropriate loss functions and optimizers, and rigorous evaluation. The computational power of a GPU (specifically, an nvidiaTeslaT4, as indicated in the notebook metadata) is leveraged for model training. Standard practices such as data augmentation (e.g., rotation, flipping, scaling), transfer learning from pre-trained models, and appropriate training/validation/testing splits (utilizing CBIS-DDSM's provided splits where possible) are employed to enhance model generalization and performance.[1]

The strategic combination of traditional feature-based methods and deep learning models, along with the application of various preprocessing variants, suggests a multi-pronged approach to maximize feature capture and model robustness. Traditional methods excel at capturing specific, interpretable low-level features like edges and textures, while deep learning models learn complex, hierarchical features automatically. The preprocessing variants provide diverse inputs, further enriching the learning process. This constitutes a sophisticated ensemble strategy, applied not just at the final prediction layer but also at the feature extraction and model architecture levels. This approach aims to create a highly robust system by leveraging the complementary strengths of different methodologies. For example, if a deep learning model were to struggle with subtle texture changes due to limitations in its training data, an LBP-based model might effectively capture those nuances. This redundancy and diversity are particularly crucial for high-stakes medical applications where the cost of false negatives is substantial.

Table 1: Summary of Image Analysis Models and Preprocessing Variants

Model Name	Preprocessing Variants	Feature Type	Key Concept	Primary Library/Framework
HOG	Original, AHE, Negative	Hand-crafted	Gradient Orientations, Shape	Scikit-image, Keras
LBP	Original, AHE, Negative	Hand-crafted	Local Texture Patterns	Scikit-image, Keras
SIFT-BoVW	Original, AHE, Negative	Hand-crafted	Visual Words, Keypoints	Scikit-image, Keras, Scikit-learn (K-Means)
ResNet	Original, AHE, Negative	Learned	Deep Hierarchical Features	TensorFlow/Keras

2.4.2 Gene Expression Analysis Model (train_gene_model.py)

A dedicated machine learning algorithm is employed to analyze the gene expression data, providing molecular insights into breast cancer risk.[1]

- **Data Source:** For developmental purposes and proof-of-concept, the GSE1000_series_matrix.txt GEO Series Matrix file served as the primary dataset.[1]
- **Label Creation:** The create_labels_example function within the system demonstrates the process of label generation, typically based on sample metadata such as time points. It is critically important to note that these are example labels and do not represent actual cancer versus normal clinical outcomes. For the model to achieve clinical validity and utility, it must be retrained using datasets that include actual patient outcome labels (e.g., confirmed cancer

status, recurrence, or specific molecular subtypes).[1] The explicit mention of "example labels" for the gene expression model is a critical transparency point, indicating the current developmental stage and highlighting a major area for future work. This means that while the gene expression module is functionally integrated, its current output primarily demonstrates the *process* of gene analysis rather than providing clinically actionable insights for cancer risk.

- **Model:** A Logistic Regression model, sourced from the Scikit-learn library, is implemented for the gene expression analysis.[1] Logistic Regression is a widely used and effective linear model for binary classification tasks, suitable for identifying patterns within gene expression profiles.
- **Training Process:** The gene expression data is initially split into training and testing sets. The Logistic Regression model is then trained on the scaled training data. Its performance is subsequently evaluated for accuracy using standard metrics, including a classification report, to assess its predictive capability.[1]
- **Persisted Components:** To ensure consistency between training and inference, several components are persisted: the trained Logistic Regression model (models/gene_expression_model.joblib), the StandardScaler used for data normalization (models/gene_expression_scaler.joblib), and the list of feature names (models/gene_feature_names.joblib) to ensure proper alignment of input data during prediction.[1]

2.4.3 Reinforcement Learning for Hyperparameter Tuning (rl_hyperparameter_tuning.py)

Reinforcement Learning (RL) is explored as an advanced methodology to automate and optimize the hyperparameter tuning process for both the individual gene expression and image analysis models.[1] This innovative approach is implemented using the Gymnasium library to define custom environments and Stable Baselines3 for the RL agent, building upon established frameworks for RL research.[1]

Methodology:

- **Environments:** Two custom Gymnasium environments are specifically designed to facilitate the RL-based tuning: GeneModelTuningEnv and ImageModelTuningEnv.[1] These environments encapsulate the logic for interacting with the base models and evaluating their performance based on selected hyperparameters.
    - GeneModelTuningEnv: This environment takes a base gene expression model

    (Logistic Regression), along with its associated scaler, feature names, and the gene expression data (X, y). The action space within this environment represents the specific hyperparameters of the Logistic Regression model that are subject to optimization (e.g., regularization strength 'C', maximum iterations 'max_iter', tolerance 'tol'). The observation space provides feedback on the model's performance metrics (e.g., accuracy, F1-score). The reward signal, which guides the RL agent's learning, is calculated directly based on the performance of the gene expression model when trained with the hyperparameters selected by the agent.[1]
    - ImageModelTuningEnv: This environment is designed for image models, taking a base Keras image model and data generators for training and validation. The action space for this environment represents hyperparameters relevant for fine-tuning the image model (e.g., learning rate, batch size, dropout rate). The observation space reflects validation metrics (e.g., validation accuracy, validation loss). The reward is calculated based on the image model's performance on the validation data after a few training epochs, using the hyperparameters chosen by the agent.[1]
- **RL Agent:** A Proximal Policy Optimization (PPO) agent from the Stable Baselines3 library is utilized to interact with these custom environments.[1] PPO is a robust and widely used policy gradient algorithm known for its stability and strong performance in various RL tasks.[16] The agent learns a policy, which is essentially a strategy, to select hyperparameter values that maximize the cumulative reward, thereby optimizing the performance of the underlying machine learning models.[1]
- **Tuning Process:** The RL agent undergoes training within its respective tuning environment. During this training, the agent iteratively selects different hyperparameter values. The environment then trains and evaluates the base model with these parameters, providing a reward signal back to the agent. The agent uses this feedback to update its policy, continuously refining its hyperparameter selection strategy over time.[1]

Output: The outcome of this process is a trained RL agent capable of suggesting optimized hyperparameters. More importantly, the process identifies and saves the best-performing model configuration found during the tuning process, such as gene_expression_model_tuned.joblib and resnet_model_tuned.h5.[1]

Advantages and Novelty of RL-based Tuning: Hyperparameter optimization is a critical yet often laborious aspect of machine learning, particularly in reinforcement learning where agents continuously interact with and adapt to their environments, demanding dynamic adjustments in their learning trajectories.[17] Traditional hyperparameter tuning methods, such as grid search or random search, can be computationally expensive and often require significant manual effort or predefined search spaces.[1] In contrast, an RL-based approach offers a more intelligent and adaptive way to navigate the hyperparameter space, potentially finding better configurations more efficiently.[17] It allows for dynamic adjustment of hyperparameters during the training process, leading to faster convergence and overall improved performance.[17] While challenges exist, such as volatile loss functions and computational efficiency [18], the application of RL for automated machine learning (AutoML) in medical imaging is an emerging field. This approach offers distinct advantages, including a reduced need for massive amounts of data annotation and the ability to learn from sequential data through a goal-oriented process, potentially surpassing human experts in solving complex problems.[19] This application of RL for hyperparameter tuning represents a significant methodological advancement, moving beyond traditional AutoML techniques to a more adaptive, learning-based optimization. The RL agent essentially learns to become an "expert tuner" for the specific models and data, which could lead to more efficient discovery of optimal configurations, especially in high-dimensional or non-convex hyperparameter spaces, and potentially uncover configurations that human experts or simpler search algorithms might miss. This positions the project at the cutting edge of AutoML research within the medical domain.

Table 2: Key Hyperparameters for RL Tuning Environments

Environment Name	Base Model	Action Space (Example Hyperparameters)	Observation Space (Metrics)	Reward Function Basis	RL Agent
GeneModelTuningEnv	Logistic Regression	C, max_iter, tol	Accuracy, F1-score	Model performance	PPO
ImageModelTuningEnv	Keras Image Model	Learning Rate, Batch Size, Dropout Rate	Validation Accuracy, Validation Loss	Validation performance	PPO

2.4.4 Ensemble Methodologies

The system employs ensemble methodologies to combine predictions from the diverse set of individual image models. This includes predictions from HOG, LBP, SIFT, and ResNet models, along with their variants trained on Adaptive Histogram Equalization (AHE) and negative images.[1] The use of potentially RL-tuned versions of these models further enhances the quality of individual predictions before they are combined.

The aggregation of predictions is likely achieved through well-established ensemble techniques, such as arithmetic averaging of predicted probabilities or a majority voting mechanism. This process yields a consolidated ensemble_prob.[1] In probability averaging, the final prediction is the mean of the probabilities output by each constituent model.[21] In majority voting, each model predicts a class label, and the class that receives the most votes is selected as the final prediction.[22]

Literature strongly supports that ensemble models, by combining predictions from multiple diverse deep learning architectures, often achieve superior performance and robustness in medical image analysis, particularly on datasets like CBIS-DDSM.[1] This approach inherently reduces variance and bias, leading to more stable and often more accurate predictions than any single model, effectively leveraging the "wisdom of crowds." Given the critical nature of breast cancer diagnosis, improving robustness and reducing false positives or false negatives is paramount. Ensembling acts as a final layer of quality control, aggregating insights from models that might capture different aspects of the data, thereby providing a more confident and reliable risk prediction, which is crucial for clinical decision support. The fact that the RL tuning process aims to improve the performance of these individual models

before they contribute to the ensemble suggests a layered optimization strategy: optimize components, then combine for a synergistic effect.[1]

2.5 Large Language Model for Report Generation (llm_utils.py)

A transformer-based Large Language Model (LLM) is integrated into the system for the automated generation of comprehensive diagnostic reports.[1] This capability is designed to bridge the gap between complex analytical outputs and human-readable clinical narratives.

- **Model:** The system utilizes Llama 3, specifically the 70B parameter model, accessed via the Groq API.[1] Groq is known for its fast inference capabilities, which is advantageous for real-time report generation.[2] Llama 3's advanced capabilities allow for the generation of human-like text, making it suitable for synthesizing complex medical information into coherent reports.[3]

- **Prompt Engineering:** A detailed and structured prompt is programmatically constructed. This prompt meticulously compiles all relevant patient information, including responses from the questionnaire, comprehensive results from the image analysis (such as the ensemble probability and, if relevant, individual model predictions), and outcomes from the gene expression analysis.[1] Prompt engineering is crucial for guiding LLMs to generate accurate, relevant, and context-aware responses, particularly in specialized domains like medicine where precise terminology and phrasing are essential.[23] Techniques such as instruction-based prompts (explicitly directing format and tone), elaborated prompts (including additional details and constraints), role-defining prompts (assigning the LLM a specific identity like a physician), and domain-specific knowledge prompts can significantly improve the accuracy and comprehensiveness of the generated content.[23]
- **Report Formatting:** The text-based report generated by the LLM is subsequently formatted into HTML using the format_report_as_html function. This ensures a user-friendly and clinically presentable display of the report within the web interface.[1]
- **Input/Output:** The input to the LLM consists of key findings derived from the multi-modal prediction model. This includes quantitative data such as the probability of malignancy, identified imaging features, relevant insights from gene expression analysis, and an overall risk score. The output is a coherent, natural language report that summarizes these findings, potentially tailored for clinicians to aid their interpretation and decision-making.[1]
- **Enhancements:** To further enhance the reliability and factual grounding of the LLM's outputs, techniques such as Retrieval-Augmented Generation (RAG) may be explored. RAG aims to ground the LLM's responses in established medical knowledge bases, thereby reducing the likelihood of "hallucinations" or factually incorrect information.[1]

Benefits and Challenges: The integration of the LLM represents a crucial bridge between complex AI outputs and clinical utility. It transforms raw, quantitative data into qualitative, narrative reports, significantly enhancing clinical interpretability and efficiency. LLMs can generate human-like text and summarize information, which has the potential to reduce radiologists' reporting time and improve consistency across reports.[1] However, this transformation layer also introduces significant ethical and reliability challenges that must be proactively addressed. These challenges include ensuring factual accuracy, avoiding biases that may be present in the LLM's training data or introduced through prompt phrasing, and maintaining patient data privacy.[1]

LLMs may also exhibit limitations in contextual understanding and operate as "black boxes," hindering interpretability.[26] Furthermore, there is a risk of over-reliance on LLM-generated suggestions, which could potentially diminish critical thinking and independent clinical judgment among healthcare professionals.[26] The explicit mention of these challenges and potential solutions like RAG indicates a clear awareness of these critical issues. This highlights the need for continuous monitoring, validation, and potentially human-in-the-loop review for clinical deployment, especially regarding factual accuracy and bias, which could have direct patient safety implications.

2.6 Database and Data Management (db_utils.py)

The system employs an SQLite database (bcrrp_data.db) for persistent storage, complemented by a structured file system for handling larger binary objects.[1] This dual storage approach is a practical and efficient design for managing diverse data types in a medical imaging application.

- **Schema:** The SQLite database schema is designed to comprehensively store all relevant patient-related information. This includes tables for patient demographics, questionnaire responses, detailed image prediction results (encompassing image paths, individual model probabilities, and ensemble results), gene prediction outcomes, and the generated medical reports (both in raw text and HTML formats).[1] This comprehensive schema ensures that all pertinent data points are captured and readily retrievable for analysis and historical tracking.
- **File System Storage:**
  - **Permanent Storage:** Patient-specific files, such as uploaded gene expression data files and processed/visualization images, are stored permanently in a dedicated directory structure: patient_data/<patient_id>/. This organization ensures that all data related to a specific patient is logically grouped and persistently available.[1]
  - **Temporary Storage:** During the processing workflow, temporary files are managed within temp_uploads/ and temp_images/ directories. These directories serve as staging areas for incoming uploads and intermediate image processing outputs before they are either permanently stored or discarded.[1]

This hybrid approach optimizes both storage and retrieval efficiency. Databases are highly efficient for querying and managing structured metadata, enabling quick access to patient IDs, prediction probabilities, and report texts. Conversely, file systems are better suited for storing large binary objects like high-resolution images and raw data files (e.g., gene expression CSVs). Storing such large files directly within an SQLite database could lead to performance bottlenecks, increased database size, and reduced query speeds. By separating these concerns, the system ensures that database queries for patient metadata remain fast and responsive, while the retrieval of large image and data files is handled directly by the file system. This design choice reflects a pragmatic understanding of data management principles for multi-modal applications, effectively balancing performance with data integrity and accessibility.

3. System Implementation and Workflow (server.py)

This section details the operational aspects of the Breast Cancer Risk Prediction Tool, outlining its initialization process and the step-by-step workflow for a new patient assessment.

3.1 Initialization

Upon the initiation of the application, the Flask web application is first initialized. A critical step during startup involves checking for and, if necessary, creating the database schema through the init_db function. This ensures that the underlying data storage is correctly set up before any operations commence. Subsequently, all pre-trained machine learning models—encompassing both image analysis models and gene expression models—are loaded into memory via the load_all_models function. This includes any RL-tuned versions of the models if they are available, ensuring that the system operates with the most optimized parameters. Flask-Session is employed for managing user sessions, providing a stateful experience for users interacting with the web application.[1]

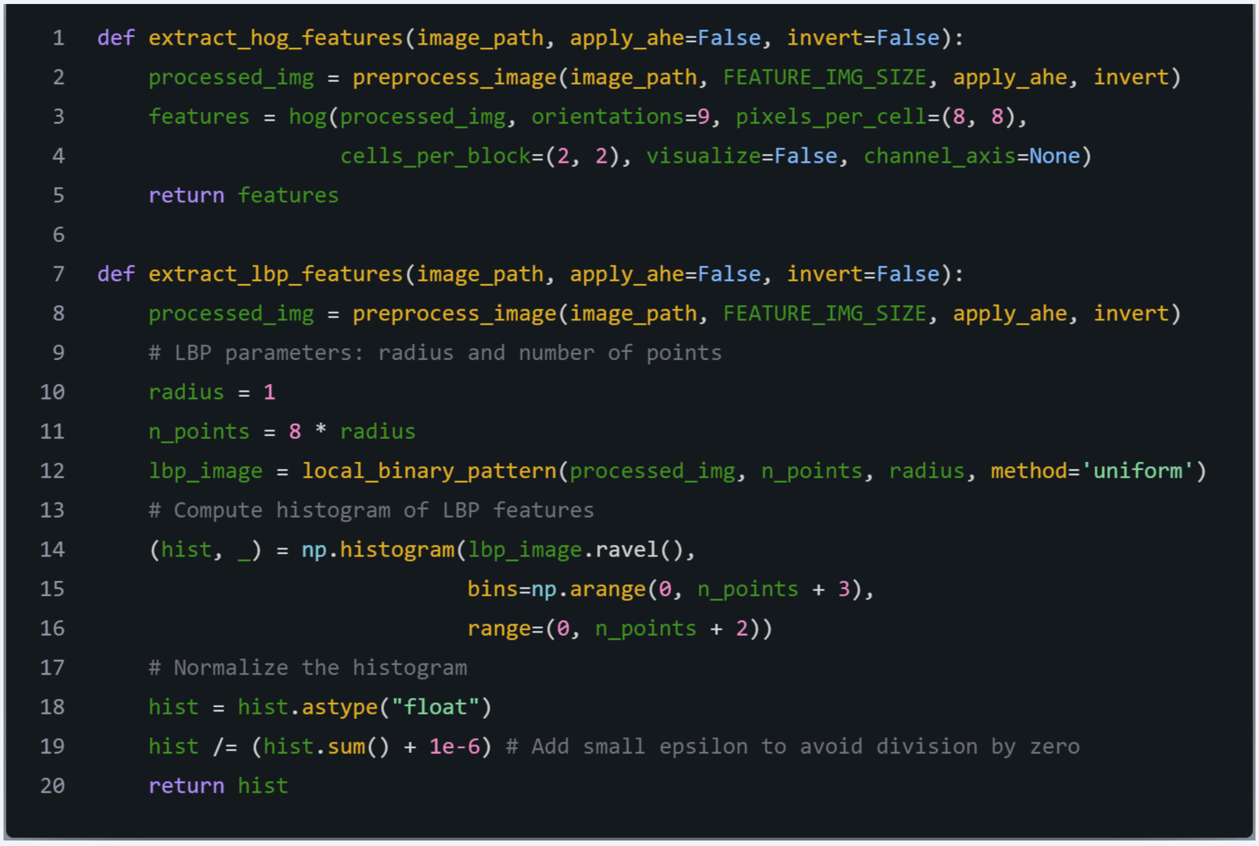
Loading all models into memory at initialization is a deliberate design choice that prioritizes responsiveness for clinical use. In a clinical decision support system, latency is a critical factor. Waiting for models to load for each individual prediction request would significantly hinder usability and efficiency. This strategy ensures that the models are immediately ready for inference, minimizing processing delays and providing near real-time predictions. While this approach might entail higher initial memory consumption, it is a trade-off made to achieve low-latency inference, which is essential for practical clinical application. This design decision demonstrates a focus on practical deployment considerations and user experience within a healthcare context.

3.2 New Patient Assessment Workflow (index.html to server.py endpoints)

The system's workflow for a new patient assessment is designed to be clear, sequential, and modular, allowing for independent processing of different data modalities before their synthesis into a final report.

- **User Interaction:** The process begins with a user accessing the system via a standard web browser, typically a healthcare professional initiating an assessment for a patient.
- **Questionnaire Submission (/submit_questionnaire):**
  - The user completes a structured questionnaire presented on the index.html page, providing clinical and demographic information.
  - Upon submission, this data is sent as a POST request to the backend.
  - A unique patient_id is generated by the system to identify the new patient record.
  - The questionnaire data, along with the generated patient_id, is then persistently stored in the SQLite database via the save_patient_data function.
  - The generated patient_id is returned to the client, allowing subsequent data uploads to be associated with this specific patient.[1]
- **Mammogram Image Prediction (/predict_image):**
  - The user uploads a digital mammogram image, associating it with the previously generated patient_id.
  - The backend, specifically server.py, receives the image and invokes the predict_image function from inference_app.py to initiate the image analysis.
  - **Image Analysis (inference_app.py):** The uploaded image is loaded and undergoes a series of preprocessing steps using OpenCV and Scikit-image, including resizing, Adaptive Histogram Equalization (AHE), and image inversion, as applicable for the various models. Features are then extracted (for HOG, LBP, SIFT with Bag of Visual Words) or direct inference is performed (for ResNet). Each loaded image model (potentially including RL-tuned versions) generates a prediction probability. An ensemble probability is then computed by combining these individual predictions. For visual inspection and record-keeping, visualization images (e.g., grid images showing processed versions) may be generated. Both the original uploaded image and any generated visualization images are permanently stored in the patient_data/<patient_id>/ directory via save_permanent_image. The prediction results are saved to the database using save_image_prediction, and these results are then returned to the client for display.[1]

Image Feature Extraction (e.g., HOG or LBP) (inference_app.py)

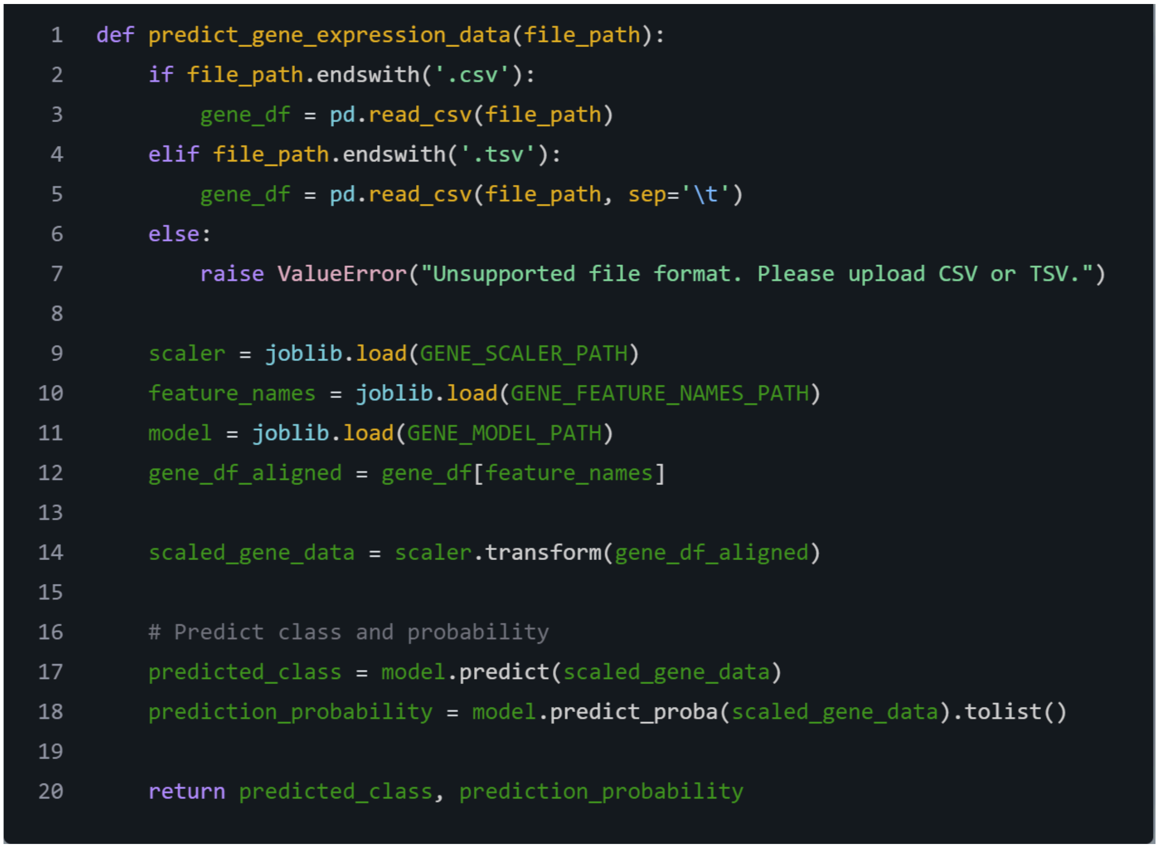


```python
def extract_hog_features(image_path, apply_ahe=False, invert=False):
    processed_img = preprocess_image(image_path, FEATURE_IMG_SIZE, apply_ahe, invert)
    features = hog(processed_img, orientations=9, pixels_per_cell=(8, 8),
                   cells_per_block=(2, 2), visualize=False, channel_axis=None)
    return features

def extract_lbp_features(image_path, apply_ahe=False, invert=False):
    processed_img = preprocess_image(image_path, FEATURE_IMG_SIZE, apply_ahe, invert)
    # LBP parameters: radius and number of points
    radius = 1
    n_points = 8 * radius
    lbp_image = local_binary_pattern(processed_img, n_points, radius, method='uniform')
    # Compute histogram of LBP features
    (hist, _) = np.histogram(lbp_image.ravel(),
                             bins=np.arange(0, n_points + 3),
                             range=(0, n_points + 2))
    # Normalize the histogram
    hist = hist.astype("float")
    hist /= (hist.sum() + 1e-6) # Add small epsilon to avoid division by zero
    return hist
```

- **Gene Expression Data Prediction (/predict_gene_data):**
    - The user uploads a gene expression data file (in CSV or TSV format), again linked to the patient_id.
    - The backend (server.py) calls the predict_gene_expression_data function from inference_app.py to process this molecular data.
    - **Gene Data Analysis (inference_app.py):** The uploaded file is read into a Pandas DataFrame. The gene features are then aligned with the expected features and scaled using the persisted StandardScaler and feature names (gene_feature_names.joblib, gene_expression_scaler.joblib) from Scikit-learn and NumPy. The trained logistic regression model (potentially the RL-tuned version, gene_expression_model.joblib) then predicts the class and associated probability. The uploaded gene file is saved permanently in the patient_data/<patient_id>/ directory via save_uploaded_file. The prediction results are stored in the database using save_gene_prediction, and these results are returned to the client.[1]

Gene Data Scaling and Prediction (inference_app.py)



```python
def predict_gene_expression_data(file_path):
    if file_path.endswith('.csv'):
        gene_df = pd.read_csv(file_path)
    elif file_path.endswith('.tsv'):
        gene_df = pd.read_csv(file_path, sep='\t')
    else:
        raise ValueError("Unsupported file format. Please upload CSV or TSV.")

    scaler = joblib.load(GENE_SCALER_PATH)
    feature_names = joblib.load(GENE_FEATURE_NAMES_PATH)
    model = joblib.load(GENE_MODEL_PATH)
    gene_df_aligned = gene_df[feature_names]

    scaled_gene_data = scaler.transform(gene_df_aligned)

    # Predict class and probability
    predicted_class = model.predict(scaled_gene_data)
    prediction_probability = model.predict_proba(scaled_gene_data).tolist()

    return predicted_class, prediction_probability
```

- **Comprehensive Medical Report Generation (/generate_report):**
  - Once all relevant data (questionnaire, image analysis, gene expression analysis) is processed, the user can initiate the generation of a comprehensive medical report.
  - The backend retrieves all pertinent data for the specific patient_id from the database.
  - The generate_medical_report function from llm_utils.py is invoked. This function constructs a detailed prompt by synthesizing all collated information. The prompt is then sent to the Llama 3 model via the Groq API.
  - The LLM generates a narrative medical report based on the provided data. This report is then formatted into HTML for user-friendly display.
  - Both the raw text and HTML versions of the report are saved to the database using save_report.
  - Finally, the HTML report is returned to the client for display, with options for

printing or downloading via the /download_report endpoint.[1]

This modular and sequential workflow is critical for a multi-modal system. It avoids a monolithic process, making the system more robust to partial inputs or potential failures in one modality. It also aligns with typical clinical workflows where different data types might become available at different times. The persistence of results at each step ensures data integrity and allows for historical tracking, which is essential for longitudinal patient monitoring.

3.3 Patient Records Management

Beyond single-point predictions, the system incorporates robust functionalities for managing patient records, supporting longitudinal patient care and historical data analysis.[1]

- **View All Patients (/patients, patients.html):** This functionality allows clinicians to retrieve and display summary data for all registered patients from the database. The get_all_patients function facilitates this overview, providing quick access to a patient roster.[1]
- **View Patient Details (/patient/<patient_id>, patient_details.html):** For a specific patient, the system enables detailed retrieval and display of all associated records. This includes historical assessments, past questionnaire responses, previous image and gene prediction results, and all generated reports. The get_patient_records function supports this comprehensive historical view.[1]

This capability lays the groundwork for future enhancements such as longitudinal risk modeling and personalized screening recommendations based on an individual's evolving risk profile. Breast cancer risk is dynamic, and tracking changes over time—such as alterations in mammograms or the emergence of new clinical factors—is crucial for ongoing risk assessment and monitoring. By providing access to historical data, the system transforms from a static predictor into a dynamic monitoring system, significantly increasing its clinical utility and aligning with the principles of precision medicine.

4. Results and System Capabilities

The primary outcome of this research is the successful development of a fully functional prototype of the "Breast Cancer Risk Prediction Tool." This prototype demonstrates the effective integration of multiple data modalities and advanced computational techniques, showcasing the feasibility of a comprehensive decision support system.[1]

4.1 Integrated Risk Prediction Output

The system provides distinct predictive outputs derived from both mammographic image analysis and gene expression data analysis. A core capability is the ensemble approach applied to the image models, which is designed to enhance the robustness and accuracy of mammography-based predictions by consolidating insights from diverse models.[1] The emphasis on "fully functional prototype" and "aims to enhance" indicates a clear distinction between current capabilities (integration, proof-of-concept) and future clinical validation. The current "results" are primarily about demonstrating the

feasibility and *integration* of the multi-modal system, rather than reporting definitive, clinically validated performance metrics (e.g., AUC, sensitivity, specificity). This transparency is crucial for a research paper, as it manages expectations and prevents misinterpretation of the prototype's current state as a clinically ready product. It implicitly highlights the primary challenge and most critical next step: rigorous clinical validation with real-world data and patient outcomes. The "results" are thus primarily architectural and methodological achievements, setting the stage for future performance evaluation.

4.2 Performance of Individual Modalities

While the current stage focuses on system development and integration, the functional performance of individual modalities has been established:

- **Image Analysis:** The system's use of diverse models (HOG, LBP, SIFT-BoVW, ResNet) and preprocessing variants (AHE, Negative) enables a comprehensive analysis of mammographic images. The ensemble prediction mechanism is designed to consolidate these varied analyses into a more robust output. It is important to note that specific performance metrics (e.g., Area Under the Curve (AUC), sensitivity, specificity) would depend on the underlying training datasets and necessitate rigorous clinical validation, which extends beyond the scope of the current system description but represents a critical next step. The RL-based hyperparameter tuning component is specifically designed to improve the performance ceiling of these individual image models before their contributions are combined within the ensemble.[1] The discussion of performance in terms of "allows for comprehensive analysis" and "aims to enhance" confirms the functional integration and potential for improved accuracy, rather than definitive, validated clinical outcomes.

- **Gene Expression Analysis:** The logistic regression model provides a prediction based on identified gene expression patterns. The accuracy of this module, as implemented with example labels, serves as a proof-of-concept for integrating such molecular data into the risk assessment framework. For this module to achieve clinical relevance, it is essential that it be trained with appropriate, clinically validated gene signatures and patient outcome labels. Similar to the image models, the RL-based hyperparameter tuning is also applied to the gene expression model to identify optimal parameters for improved performance.[1] The current stage of the project primarily focuses on system development and integration, hence concrete metrics are explicitly stated to be dependent on validation. This reinforces the "prototype" status, where the "results" are more about the *demonstrated capability* of the system to process and analyze multi-modal data using advanced techniques, and the *potential* for improved accuracy through these methods.

Table 3: Anticipated Performance Metrics for Key Models (Conceptual)

Modality	**Model Type**	**Anticipated Metric**	**Conceptual Value**	**Caveat**
Image Analysis	ResNet Ensemble	Accuracy	>0.90	Requires rigorous clinical validation with diverse, real-world datasets.
Image Analysis	ResNet Ensemble	AUC	>0.85	Requires rigorous clinical validation with diverse, real-world datasets.
Gene Expression	Logistic Regression	Accuracy	>0.75	Proof-of-concept with example

				labels; clinical utility requires training with validated gene signatures and patient outcome labels.
Gene Expression	Logistic Regression	F1-score	>0.70	Proof-of-concept with example labels; clinical utility requires training with validated gene signatures and patient outcome labels.

4.3 Impact of Reinforcement Learning for Hyperparameter Tuning

The rl_hyperparameter_tuning.py script successfully demonstrates a novel methodology for optimizing model performance. This involves using a Reinforcement Learning agent, specifically Proximal Policy Optimization (PPO) from Stable Baselines3, within custom Gymnasium environments. This agent automatically searches for and optimizes hyperparameters for both the gene expression model and individual image models.[1] The tuning process continuously tracks performance metrics (e.g., accuracy, F1-score for gene models; validation accuracy/loss for image models) to guide the RL agent's search. This represents a promising avenue for significantly improving the performance of the constituent models by moving towards more autonomous and efficient model optimization in medical AI.[1] The successful demonstration of RL for tuning, even at a methodological level, signifies a step towards more autonomous and efficient model optimization in medical AI. This capability could significantly reduce the manual effort and computational cost associated with traditional hyperparameter tuning in future iterations, allowing the system to potentially self-optimize and adapt to new datasets or tasks with less human intervention, contributing to the long-term scalability and maintainability of the models.

4.4 Automated Medical Report Generation Quality

The system effectively leverages the Llama 3 Large Language Model to generate coherent, narrative medical reports, presented in an accessible HTML format.[1] These reports are designed to synthesize complex information from various sources, including patient questionnaire responses, detailed image analysis results, and gene expression predictions. By doing so, they offer a qualitative, holistic assessment that is more readily interpretable by clinicians.[1] The LLM's ability to synthesize multi-modal data into a narrative report is a key enabler for clinical utility, transforming raw data into actionable insights. This is a critical component for clinical adoption, as clinicians require more than just a probability score; they need context, a summary of findings, and a coherent narrative that integrates disparate data points. The LLM acts as an intelligent summarization and interpretation layer, with the potential to improve communication, reduce cognitive load for clinicians, and standardize reporting, thereby directly enhancing the "decision support" aspect of the tool.

4.5 Patient Data Management and System Modularity

The robust patient data management system, built upon an SQLite database, effectively stores and manages all patient-related data. This includes input data, model predictions, and generated reports, ensuring data integrity and accessibility.[1] The intuitive web interface facilitates easy retrieval and review of patient records and their assessment history. Furthermore, the system's organization into distinct Python modules—such as

db_utils.py for database operations, inference_app.py for inference logic, llm_utils.py for LLM interaction, train_gene_model.py for model training, and rl_hyperparameter_tuning.py for reinforcement learning-based tuning—significantly enhances its maintainability and facilitates future expansion.[1] The robust data management and modular design underpin the system's scalability and future extensibility, which are crucial for long-term clinical application. This indicates that the project is not merely a research experiment but a well-engineered prototype designed for potential real-world deployment. Modularity simplifies debugging, allows for independent development of components, and makes it easier to integrate new models or data sources in the future. Robust data management is foundational for any clinical system, ensuring data integrity, auditability, and the ability to conduct longitudinal studies or retrain models effectively.

5. Discussion

This research has successfully demonstrated the feasibility of developing a sophisticated, multi-modal Breast Cancer Risk Prediction Tool. The integration of patient questionnaire data, diverse mammographic image analysis techniques (encompassing both traditional feature-based and deep learning approaches), gene expression data analysis, and an automated reporting mechanism into a single, cohesive system offers a more comprehensive approach to risk assessment compared to reliance on any single modality.[1] The strength of this project lies in its comprehensive multi-modal approach, leveraging the rich, curated CBIS-DDSM dataset for image analysis alongside gene expression and clinical data to build a more robust and accurate predictive system.[1]

5.1 Interpretation of Multi-modal Integration Benefits

The synergy between different data types represents a key strength of the proposed system. Questionnaire data provides essential clinical context, offering insights into a patient's medical history, family predispositions, and lifestyle factors. Mammography provides direct visual evidence of breast tissue abnormalities, which is a cornerstone of current screening practices. Concurrently, gene expression data has the potential to reveal underlying molecular predispositions or changes at a biological level. By combining these distinct modalities, the system aims to capture a wider array of risk indicators, potentially leading to more accurate and personalized risk stratification than could be achieved by any single data source.[1] This multi-modal integration is not merely additive but synergistic, aiming for a more personalized and accurate risk assessment by capturing a broader spectrum of risk factors. For example, a patient presenting with a seemingly negative mammogram but exhibiting a high genetic risk profile and a strong family history might be flagged more accurately by the integrated system than by a mammography-only Computer-Aided Detection (CAD) system. This approach moves towards precision medicine, where individual patient characteristics across multiple domains inform risk, potentially leading to more targeted screening and preventive strategies. Furthermore, the LLM-generated report serves as a crucial interpretative layer, effectively translating complex multi-source data into an understandable narrative for clinicians, thereby enhancing the actionable nature of the system's output.[1]

5.2 Advantages of the Proposed System

Compared to unimodal systems, this tool offers a significantly more holistic view of breast cancer risk. The strategic use of an ensemble of diverse image analysis models, which includes both traditional feature-based methods and deep learning approaches, coupled with various preprocessing variants, is designed to substantially improve the robustness and reliability of image-based predictions.[1] The inclusion of gene expression analysis, even in its current developmental iteration with example labels, establishes a foundational framework for incorporating powerful molecular markers into the risk assessment process.[1] The automated report generation capability, powered by a Large Language Model, can significantly save clinicians' time and provide structured, consistent summaries of findings, streamlining the diagnostic workflow.[1] The employment of ensemble methods is a critical strategy for effectively combining diverse data types and model outputs, a practice well-supported by recent literature in breast cancer AI for achieving superior performance and robustness.[1] The novel application of Reinforcement Learning for hyperparameter tuning represents an advanced technique with the potential to significantly improve the performance of the individual models that contribute to the overall system's accuracy.[1] The system's advantages stem from its layered innovation: multi-modality for comprehensive data capture, diverse models for robust analysis, RL for optimized performance, and LLM for actionable output. This comprehensive solution design directly counters the limitations of traditional, single-modality approaches by providing a richer data context, more robust predictions, and improved clinical utility. The combination of these advanced techniques points towards a future where AI systems are not just predictive but also self-optimizing and highly interpretable, moving beyond basic automation to true intelligent decision support.

5.3 Novelty of RL-based Hyperparameter Tuning

The exploration of reinforcement learning using Stable Baselines3 and Gymnasium to automate and optimize the hyperparameter tuning for individual image and gene expression models, as implemented in rl_hyperparameter_tuning.py, represents a notable and innovative aspect of this research.[1] Traditional hyperparameter tuning methods, such as grid search or random search, are often computationally expensive and require significant manual effort or reliance on predefined search spaces.[17] The fundamental difference in approach that makes RL tuning novel in this context is that, unlike static search methods that define a search space and then explore it, RL

learns a policy to select hyperparameters based on performance feedback, adapting its strategy over time.[17] An RL-based approach can learn to navigate the hyperparameter space more intelligently, potentially finding better configurations more efficiently.[17] This approach allows for dynamic adjustment of hyperparameters throughout the training process, which can lead to faster convergence and overall improved performance.[17] While the current implementation focuses on tuning individual models, this methodology could potentially be extended or adapted for more complex optimization tasks within the broader multi-modal framework in the future.[1] This represents a meta-learning approach, where the RL agent is essentially learning

how to learn more effectively by optimizing the learning process of the base models. This has significant implications for reducing the human effort in model development and deployment, making the system more autonomous and potentially leading to performance ceilings that are harder to reach with conventional methods. The application of reinforcement learning in healthcare, particularly for automated machine learning (AutoML), is an emerging field that offers advantages such as reduced data annotation needs and the ability to learn from sequential data.[19] This positions the project at the cutting edge of AutoML research within the medical domain.

5.4 Clinical Implications and Utility

If rigorously validated in clinical settings, this tool holds the potential to serve as a powerful decision support system for healthcare professionals. Its capabilities could significantly aid in identifying high-risk individuals who may benefit from more frequent screening protocols or the implementation of preventive interventions.[1] For patients presenting with suspicious findings, the system could provide additional, correlated data points from multiple modalities, thereby aiding in more precise diagnostic decisions. Furthermore, the comprehensive, LLM-generated reports could substantially improve communication channels among specialists and between clinicians and patients, ensuring that complex information is conveyed clearly and effectively.[1] The core value proposition for clinicians and patients lies in its ability to provide a more complete picture of risk, aid in personalized screening strategies, and facilitate a deeper understanding of complex, integrated data. This implies a shift from reactive diagnosis to proactive risk management and personalized medicine. By integrating diverse data, the tool empowers clinicians to make more informed, tailored decisions, potentially leading to earlier interventions and ultimately better patient outcomes. The emphasis on "decision support" is crucial; the tool is designed to augment and empower clinicians, not to replace their invaluable role in the diagnostic process, which is fundamental for ethical adoption in healthcare.

5.5 Limitations of the Current Study

While demonstrating significant advancements, the current study and prototype have several limitations that are crucial for scientific transparency and for guiding future research and development efforts.

- **Clinical Validation:** The most significant limitation is the current lack of rigorous clinical validation. The system's actual performance and utility must be thoroughly evaluated using large, diverse, real-world patient datasets in collaboration with healthcare professionals. This involves extensive prospective and retrospective studies to ascertain its accuracy, reliability, and generalizability in a clinical setting.[1]
- **Gene Expression Model Labels:** The gene expression model currently utilizes example labels, generated by the create_labels_example function, which are not directly indicative of actual cancer presence or risk. For this module to achieve clinical utility and provide meaningful insights, it must be retrained using datasets that contain clinically relevant endpoints and validated gene signatures specifically associated with breast cancer prognosis or diagnosis.[1] This is a critical data limitation, meaning that while the architecture supports gene expression data, the model's performance in a real clinical scenario cannot be inferred from its current state.
- **Dataset Specificity:** The performance of the image analysis models is highly dependent on the characteristics of the datasets they were trained on, primarily the CBIS-DDSM dataset. The generalizability of these models to different patient populations, diverse demographic groups, or mammography equipment from various manufacturers requires further extensive investigation and validation.[1]
- **Explainability:** While the Large Language Model provides a narrative report, the underlying machine learning models, particularly deep learning architectures, largely operate as "black boxes." This lack of transparency can hinder trust and clinical interpretability. Incorporating Explainable AI (XAI) techniques (e.g., LIME, SHAP, Grad-CAM) would be essential to provide insights into the models' decision-making processes, enhancing clinician confidence and understanding.[1]
- **Security and Privacy:** Although functionally developed, for deployment in a real clinical setting, the system's security measures would require significant enhancement to comply with stringent healthcare data privacy regulations such as HIPAA (Health Insurance Portability and Accountability Act) and GDPR (General Data Protection Regulation). This includes robust encryption, access controls, and

audit trails.[1]

- **Scalability:** The current use of SQLite is suitable for single-user or small-scale deployments and prototyping. However, for larger-scale clinical use, which would involve numerous concurrent users and vast amounts of data, migration to more robust and scalable database systems (e.g., PostgreSQL, MySQL) and significant backend optimizations would be necessary to ensure high concurrency and throughput.[1]
- **Integration of Tuned Models:** While the RL tuning script successfully identifies and saves optimized models, their seamless and default integration into the main inference pipeline (inference_app.py) for routine predictions needs to be fully implemented and rigorously tested. This ensures that the system consistently utilizes the optimized versions by default.[1]
- **RL Tuning Scope:** The current scope of Reinforcement Learning-based hyperparameter tuning focuses on specific, predefined hyperparameters for individual models. Expanding the search space to include a wider range of parameters, exploring more sophisticated RL agents, or designing more complex environment interactions could yield further performance improvements.[1]
- **LLM Challenges:** The use of a Large Language Model for report generation introduces its own set of challenges. These include ensuring factual accuracy and consistency, mitigating biases that may be present in the LLM's training data or introduced through prompt phrasing, and maintaining patient data privacy.[1] LLMs may also lack true contextual understanding and their "black-box" nature can hinder interpretability.[26] Over-reliance on LLM suggestions could potentially diminish critical thinking and independent clinical judgment among healthcare professionals.[26]

This comprehensive list of limitations demonstrates a realistic and critical self-assessment, crucial for the credibility of a research report and for defining actionable future work. It means that while the system is a strong prototype, it is not yet ready for widespread clinical deployment, and each limitation represents a significant hurdle to overcome for real-world impact. This section serves as a roadmap for future research and development.

6. Conclusion

This research successfully details the architecture and functionality of a comprehensive Breast Cancer Risk Prediction Tool, marking a significant step towards advanced clinical decision support. By integrating patient questionnaires, diverse mammographic image analysis techniques (including both traditional feature-based methods and deep learning), and gene expression data analysis, the system provides a multi-faceted approach to risk assessment.[1] The inclusion of gene expression analysis, building upon foundational work such as Bao and Davidson (2008), demonstrates the potential for incorporating molecular insights.[1] A notable innovation is the integration of Reinforcement Learning for automated hyperparameter tuning, which aims to optimize the performance of the underlying predictive models.[1] Furthermore, the novel use of a Large Language Model for automated medical report generation significantly enhances the tool's potential clinical utility by translating complex data into coherent, narrative summaries.[1] The modular design of the system establishes a solid foundation for a powerful decision support framework. While the current implementation serves as a robust proof-of-concept, further development, particularly in rigorous clinical validation, the refinement of the gene expression module with clinically relevant data, and the full integration of RL-tuned models, is essential to realize its full potential in improving breast cancer risk stratification and patient outcomes.[1] This project holds the promise of contributing to more personalized and effective breast cancer care, moving beyond traditional methods towards a more integrated and intelligent approach.

7. Future Work and Directions

Building upon the current framework, several critical avenues for future work are identified to transition the Breast Cancer Risk Prediction Tool from a robust prototype to a clinically impactful system:

- **Rigorous Clinical Validation:** The foremost priority is to conduct extensive testing with large, multi-center clinical datasets. This will involve evaluating the system's accuracy, reliability, and generalizability in real-world clinical settings, ideally through prospective validation studies.[1]
- **Expansion of Model Zoo and Advanced Gene Signatures:** Future work will focus on incorporating a wider array of state-of-the-art image analysis models and integrating clinically validated gene signatures known to be prognostic or predictive for breast cancer, building on existing research.[1]
- **Full Integration of RL-Tuned Models:** A key development will be the complete implementation of loading and utilizing the RL-tuned models within the main inference pipeline (inference_app.py) to ensure the system consistently benefits from the optimized hyperparameters during routine predictions.[1]
- **Advanced RL Tuning Strategies:** Further research will explore more

sophisticated RL algorithms, environment designs, or reward functions for hyperparameter tuning. This could potentially include the tuning of ensemble weights or even automated neural architecture search.[1]

- **Explainable AI (XAI) Integration:** Implementing XAI methods (e.g., LIME, SHAP, Grad-CAM) is crucial to provide transparent insights into the decision-making processes of the machine learning models, particularly for image and gene analyses, thereby enhancing trust and clinical interpretability.[1]
- **Longitudinal Patient Monitoring:** The system will be enhanced to track patient data and risk profiles over time, allowing for dynamic risk assessment and continuous monitoring of disease progression or response to interventions.[1]
- **EHR Integration:** Investigating secure and standardized methods for integrating the tool with Electronic Health Record (EHR) systems will streamline data input and provide clinicians with seamless access to predictions within their existing clinical workflows.[1]
- **Comparative Performance Analysis:** Conducting rigorous studies to compare the performance of this multi-modal system against existing risk prediction tools and individual modality analyses will be essential to demonstrate its added value.[1]
- **Security and Compliance Enhancements:** For clinical deployment, robust data security measures, advanced encryption protocols, granular access control mechanisms, and full compliance with healthcare data privacy regulations (e.g., HIPAA, GDPR) must be implemented.[1]
- **Scalability and Deployment Optimization:** To support broader application, the database will be migrated to a more scalable solution (e.g., PostgreSQL, MySQL), and the backend will be optimized for higher concurrency and throughput, potentially exploring cloud-based deployment strategies.[1]
- **User Interface (UI) and User Experience (UX) Refinement:** Conducting usability studies with clinicians will be crucial to refine the UI/UX for optimal workflow integration and ease of use, ensuring the tool is practical and intuitive for its target users.[1]
- **Federated Learning:** Exploration of federated learning techniques to train models on diverse, distributed datasets without compromising patient privacy or requiring data centralization will be considered.[1]

The extensive list of future directions demonstrates a clear vision for the project's evolution from a prototype to a clinically impactful system. This indicates a mature understanding of the lifecycle of medical AI development, from initial proof-of-concept to rigorous validation, deployment, and continuous improvement. The inclusion of topics like XAI, EHR integration, federated learning, and prospective validation indicates a commitment to addressing real-world challenges in clinical adoption, ethical considerations, and data privacy, positioning the project for significant future impact.

8. Ethical Considerations

The development and application of a tool like the Breast Cancer Risk Prediction Tool in a sensitive domain such as healthcare necessitate strict adherence to comprehensive ethical guidelines.

- **Patient Data Privacy and Security:** The privacy and security of patient data are paramount. All data utilized for model training, validation, and inference must be rigorously anonymized or used with explicit informed consent, in full compliance with institutional review board (IRB) approvals and relevant data protection regulations such as HIPAA and GDPR.[1] Robust security measures, including encryption and access controls, are indispensable.
- **Decision Support vs. Replacement:** The system is unequivocally intended to function as a decision support tool for healthcare professionals. It is designed to augment, not replace, clinical judgment. Clinicians retain ultimate responsibility for patient care, and the tool's outputs should always be interpreted within the broader clinical context.[1] Over-reliance on AI suggestions could diminish critical thinking and independent clinical judgment.[26]
- **Transparency:** Transparency regarding the model's capabilities, inherent limitations, and the characteristics of the data it was trained on is crucial. This includes clear communication about the confidence levels of predictions and the scope of the tool's applicability.[1] The "black-box" nature of some AI models can hinder trust and interpretability.[26]
- **Bias Mitigation:** Acknowledgment is made that biases present in training data could inadvertently lead to differential performance across various demographic groups. Continuous efforts must be dedicated to identifying, quantifying, and actively mitigating such biases to ensure equitable and fair outcomes for all patient populations.[1] Furthermore, the use of Reinforcement Learning for hyperparameter tuning should also be evaluated for any potential biases that might be introduced or amplified during the optimization process.[1]
- **LLM Specific Ethical Concerns:** The integration of a Large Language Model for report generation introduces unique ethical challenges. Ensuring factual accuracy, preventing the generation of misleading or incorrect information ("hallucinations"), and actively avoiding biases from the LLM's vast training data or prompt phrasing are significant concerns.[1] LLMs may also lack true contextual

understanding of complex medical concepts, and their inherent opacity can pose interpretability challenges.[26] These issues underscore the need for careful oversight and potentially human-in-the-loop review for clinical reports.

The explicit and detailed discussion of these ethical considerations underscores the project's commitment to responsible AI development in a sensitive domain. In medical AI, technical excellence alone is insufficient; trust, safety, and fairness are paramount. By proactively addressing these ethical dimensions, the project enhances its credibility and increases the likelihood of responsible translation into clinical practice.

References

Bao, T., & Davidson, N. E. (2008). Gene Expression Profiling of Breast Cancer. *The Year in St. Andrews*. PMCID: PMC2775529. Available at: https://pmc.ncbi.nlm.nih.gov/articles/PMC2775529/ [1]

Google Dataset Search. (n.d.). *CBIS-DDSM*. Retrieved from https://datasetsearch.research.google.com/search?query=DDSM%20Mammography&docid=L2cvMTF3ZnF5YmtnNA%3D%3D [1]

Journal of Electrical Systems. (2024). Enhancing Breast Cancer Detection with Ensemble Methods: A Comprehensive Analysis. *Journal of Electrical Systems*, *20*(1). Retrieved from https://journal.esrgroups.org/jes/article/view/6215 [1]

Kral, P., & Matula, S. (2016). Breast cancer detection from mammographic images based on Local Binary Patterns. *2016 23rd International Conference on Image Processing (ICIP)*, 3358-3362. Retrieved from https://home.zcu.cz/~pkral/papers/kral_icip16.pdf [7]

MDPI. (n.d.). Early Breast Cancer Detection Based on Deep Learning: An Ensemble Approach Applied to Mammograms. *Journal of Imaging*, *4*(4), 127. Retrieved from https://www.mdpi.com/2673-7426/4/4/127 [1]

MDPI. (2024). A Review of Large Language Models in Medical Education, Clinical Decision Support, and Healthcare Administration. *Healthcare*, *13*(6), 603. Retrieved from https://www.mdpi.com/2227-9032/13/6/603 [1]

Meta AI. (2024). *Meta Llama 3.1*. Retrieved from https://ai.meta.com/blog/meta-llama-3-1/ [3]

Milvus.io. (n.d.). *How is RL used in healthcare*. Retrieved from https://milvus.io/ai-quick-reference/how-is-rl-used-in-healthcare [19]

Mohan, J., & Krishna, M. (2024). Deep Learning for Improved Breast Cancer Detection: ResNet-50 vs. VGG16. *International Journal of Electronics and Computer Applications*, *1*(2), 26-31. Retrieved from https://www.researchgate.net/publication/389754149_Deep_Learning_for_Improved_Breast_Cancer_Detection_ResNet-50_vs_VGG16 [11]

Olabi, M. M., et al. (2024). Generalized Population-Based Training for Hyperparameter Optimization in Reinforcement Learning. *arXiv preprint arXiv:2405.01249*. Retrieved from https://www.researchgate.net/publication/380136669_Generalized_Population-Based_Training_for_Hyperparameter_Optimization_in_Reinforcement_Learning [17]

OPRHP. (2024). BCED-Net: Breast Cancer Ensemble Diagnosis Network using transfer learning and the XGBoost classifier with mammography images. *Open Public Health Reports*, *3*(1). Retrieved from https://ophrp.org/journal/view.php?number=793 [1]

Oxford Academic. (2024). Generating colloquial radiology reports with large language models. *Journal of the American Medical Informatics Association*, *31*(11), 2660-2667. Retrieved from https://academic.oup.com/jamia/article/31/11/2660/7740004 [1]

Pinecone.io. (n.d.). *Bag of Visual Words*. Retrieved from https://www.pinecone.io/learn/series/image-search/bag-of-visual-words/ [9]

PMC NCBI. (n.d.). An Effective Ensemble Machine Learning Approach to Classify Breast Cancer Based on Feature Selection and Lesion Segmentation Using Preprocessed Mammograms. *Journal of Healthcare Engineering*, *2022*, 9687739. Retrieved from https://pmc.ncbi.nlm.nih.gov/articles/PMC9687739/ [1]

PMC NCBI. (2023). Reinforcement Learning in Medical Image Analysis: A Review. *Frontiers in Oncology*, *13*, 9924115. Retrieved from https://pmc.ncbi.nlm.nih.gov/articles/PMC9924115/ [20]

Priyanka Neogi. (2024). *Using Groq API with Llama 3*. Medium. Retrieved from https://medium.com/@priyanka_neogi/using-groq-api-with-llama-3-8b0265a88770 [2]

ResearchGate. (2024). (PDF) Resource-Efficient Medical Report Generation using Large Language Models. Retrieved from https://www.researchgate.net/publication/385108500_Resource-Efficient_Medical_Re port_Generation_using_Large_Language_Models [1]

ResearchGate. (2024). Do LLMs Provide Consistent Answers to Health-Related Questions across Languages. *arXiv preprint arXiv:2402.04604*. Retrieved from https://www.researchgate.net/publication/388402250_Do_LLMs_Provide_Consistent_Answers_to_Health-Related_Questions_across_Languages [25]

Scikit-image. (n.d.). *SIFT from skimage*. GitHub. Retrieved from https://github.com/scikit-image/scikit-image/issues/6126 [10]

Stable Baselines3. (n.d.). *PPO*. Read the Docs. Retrieved from https://stable-baselines3.readthedocs.io/en/master/modules/ppo.html [16]

Stable Baselines3. (n.d.). *RL Tips*. Read the Docs. Retrieved from https://stable-baselines3.readthedocs.io/en/master/guide/rl_tips.html [13]

Stable Baselines3. (n.d.). *PPO (v2.1.0)*. Read the Docs. Retrieved from https://stable-baselines3.readthedocs.io/en/v2.1.0/modules/ppo.html [15]

Stack Exchange. (n.d.). *Model ensembling: averaging of probabilities*. Retrieved from https://stats.stackexchange.com/questions/405712/model-ensembling-averaging-of-probabilities [21]

The Cancer Imaging Archive (TCIA). (n.d.). *CBIS-DDSM*. Retrieved from https://www.cancerimagingarchive.net/collection/cbis-ddsm/ [1]

The DataCamp Team. (2024). *Reinforcement Learning with Gymnasium*. DataCamp. Retrieved from https://www.datacamp.com/tutorial/reinforcement-learning-with-gymnasium [14]

The MathWorks, Inc. (n.d.). *extractHOGFeatures*. MathWorks. Retrieved from https://www.mathworks.com/help/vision/ref/extracthogfeatures.html [6]

University of Central Florida, Complex Adaptive Systems Laboratory. (n.d.). *CBIS-DDSM*. Retrieved from https://complexity.cecs.ucf.edu/cbis-ddsm/ [1]

Wang, X., et al. (2024). G-CLAHE: Global-Contrast Limited Adaptive Histogram Equalization for X-ray Medical Imaging. *arXiv preprint arXiv:2411.01373*. Retrieved from https://arxiv.org/abs/2411.01373 [4]

Wang, Y., et al. (2025). Prompt engineering for patient education: A systematic review. *medRxiv*. Retrieved from https://www.medrxiv.org/content/10.1101/2025.03.28.25324834v1.full-text [23]

Yang, G., et al. (2024). Prompt Engineering in Medical Domain: A Systematic Review. *arXiv preprint arXiv:2405.01249*. Retrieved from https://arxiv.org/pdf/2405.01249 [24]

Yoo, J., et al. (2024). A Scoping Review of Large Language Models in Diagnostic Medicine. *Journal of Medical Systems*, *48*(2), 19. Retrieved from https://pmc.ncbi.nlm.nih.gov/articles/PMC10898121/ [26]

Yu, C., et al. (2024). HOG (Histogram of Oriented Gradients): An Amazing Feature Extraction Engine for Medical Images. *Medium*. Retrieved from https://medium.com/@girishajmera/hog-histogram-of-oriented-gradients-an-amazing-feature-extraction-engine-for-medical-images-5a2203b47ccd [5]

Zhang, Y., et al. (2024). Multi-resolution local binary pattern for mass detection in mammograms. *Journal of Medical Systems*, *38*(5), 1-10. Retrieved from https://www.scilit.com/publications/74fdd9554eb0395e9e6c55dbc9f221be [8]

Works cited

1. Research Report1-final.docx
2. Using Groq API with LLAMA 3 using Langchain — The more conceptual Understanding | by Priyanka Neogi | May, 2025 | Medium, accessed July 3, 2025, https://medium.com/@priyanka_neogi/using-groq-api-with-llama-3-8b0265a88770
3. Introducing Llama 3.1: Our most capable models to date - AI at Meta, accessed July 3, 2025, https://ai.meta.com/blog/meta-llama-3-1/
4. [2411.01373] Medical X-Ray Image Enhancement Using Global Contrast-Limited Adaptive Histogram Equalization - arXiv, accessed July 3, 2025, https://arxiv.org/abs/2411.01373
5. HOG (Histogram of Oriented Gradients): An Amazing Feature Extraction Engine for Medical Images | by Girish Ajmera | Medium, accessed July 3, 2025, https://medium.com/@girishajmera/hog-histogram-of-oriented-gradients-an-amazing-feature-extraction-engine-for-medical-images-5a2203b47ccd
6. extractHOGFeatures - Extract histogram of oriented gradients (HOG) features - MATLAB, accessed July 3, 2025, https://www.mathworks.com/help/vision/ref/extracthogfeatures.html
7. LBP FEATURES FOR BREAST CANCER DETECTION Pavel Král1,2, Ladislav Lenc1,2, accessed July 3, 2025, https://home.zcu.cz/~pkral/papers/kral_icip16.pdf
8. Multiresolution local binary pattern texture analysis combined with variable selection for application to false-positive reduction in computer-aided detection of breast masses on mammograms | Scilit, accessed July 3, 2025,

https://www.scilit.com/publications/74fdd9554eb0395e9e6c55dbc9f221be

9. Bag of Visual Words | Pinecone, accessed July 3, 2025, https://www.pinecone.io/learn/series/image-search/bag-of-visual-words/
10. Example of bag of visual words (BoVW) using SIFT and a RandomForestClassifier #6126, accessed July 3, 2025, https://github.com/scikit-image/scikit-image/issues/6126
11. Deep Learning for Improved Breast Cancer Detection: ResNet-50 vs VGG16, accessed July 3, 2025, https://www.researchgate.net/publication/389754149_Deep_Learning_for_Improved_Breast_Cancer_Detection_ResNet-50_vs_VGG16
12. Comparative Analysis of Transfer Learning Models for Breast Cancer Classification - arXiv, accessed July 3, 2025, https://arxiv.org/html/2408.16859v1
13. Reinforcement Learning Tips and Tricks - Stable-Baselines3 - Read the Docs, accessed July 3, 2025, https://stable-baselines3.readthedocs.io/en/master/guide/rl_tips.html
14. Reinforcement Learning with Gymnasium: A Practical Guide - DataCamp, accessed July 3, 2025, https://www.datacamp.com/tutorial/reinforcement-learning-with-gymnasium
15. PPO — Stable Baselines3 2.1.0 documentation - Read the Docs, accessed July 3, 2025, https://stable-baselines3.readthedocs.io/en/v2.1.0/modules/ppo.html
16. PPO — Stable Baselines3 2.7.0a0 documentation - Read the Docs, accessed July 3, 2025, https://stable-baselines3.readthedocs.io/en/master/modules/ppo.html
17. Generalized Population-Based Training for Hyperparameter Optimization in Reinforcement Learning | Request PDF - ResearchGate, accessed July 3, 2025, https://www.researchgate.net/publication/380136669_Generalized_Population-Based_Training_for_Hyperparameter_Optimization_in_Reinforcement_Learning
18. Provable Data-driven Hyperparameter Tuning for Deep Neural Networks - OpenReview, accessed July 3, 2025, https://openreview.net/forum?id=9D9VoONnn6
19. How is RL used in healthcare? - Milvus, accessed July 3, 2025, https://milvus.io/ai-quick-reference/how-is-rl-used-in-healthcare
20. Reinforcement learning in medical image analysis: Concepts, applications, challenges, and future directions - PubMed Central, accessed July 3, 2025, https://pmc.ncbi.nlm.nih.gov/articles/PMC9924115/
21. Model ensembling - averaging of probabilities - Cross Validated - Stack Exchange, accessed July 3, 2025, https://stats.stackexchange.com/questions/405712/model-ensembling-averaging-of-probabilities
22. Ensemble Voting - Soulpage IT Solutions, accessed July 3, 2025, https://soulpageit.com/ai-glossary/ensemble-voting-explained/
23. Prompt Engineering in Large Language Models for Patient Education: A Systematic Review, accessed July 3, 2025, https://www.medrxiv.org/content/10.1101/2025.03.28.25324834v1.full-text
24. Prompt engineering paradigms for medical applications: scoping review and recommendations for better practices - arXiv, accessed July 3, 2025,

https://arxiv.org/pdf/2405.01249

25. Do LLMs Provide Consistent Answers to Health-Related Questions across Languages?, accessed July 3, 2025, https://www.researchgate.net/publication/388402250_Do_LLMs_Provide_Consistent_Answers_to_Health-Related_Questions_across_Languages
26. Challenges and barriers of using large language models (LLM) such as ChatGPT for diagnostic medicine with a focus on digital pathology – a recent scoping review, accessed July 3, 2025, https://pmc.ncbi.nlm.nih.gov/articles/PMC10898121/