Chapter 6

Optimizing Breast Cancer Detection: Integrating Convolutional Neural Networks and a Novel Weight Updating Algorithm in Histopathological Analysis

6.1 Introduction

Convolutional Neural Networks (CNNs) represent a significant advancement in the field of deep learning, particularly within the realm of computer vision. These powerful neural network architectures are specifically designed to process data that come in the form of multiple arrays, such as a color image composed of three 2D arrays containing pixel intensities in the RGB color space. The architecture of CNNs is inspired by the biological visual cortex, where individual neurons respond to stimuli only in a restricted region of the visual field known as the receptive field. This biological parallel has inspired the development of CNNs to automatically and adaptively learn spatial hierarchies of features from images and other forms of spatial data.

CNNs are distinguished by their unique ability to learn feature representations directly from image data, thus reducing the need for manual feature extraction. They exploit the 2D structure of input images through convolutional layers, pooling layers, and fully connected layers to capture important features at various levels of abstraction. This process enables CNNs to recognize patterns with variability and robustness to distortions and geometric transformations.

The versatility and efficacy of CNNs have led to their widespread application beyond image recognition and classification tasks. Today, CNNs are employed in a myriad of applications across different industries. In healthcare, they are used for medical image analysis, enabling more accurate diagnoses of diseases from imagery such as MRI scans and X-rays. In the automotive industry, CNNs power advanced driver assistance systems (ADAS) and autonomous vehicles by providing the capability to recognize and interpret the surrounding environment. In retail, they are used for automatic checkout systems, customer sentiment analysis, and inventory management through visual data processing. Additionally, CNNs play a crucial role in security, enabling sophisticated surveillance systems and facial recognition technologies.

The development and success of CNNs have significantly contributed to the surge in interest and advancements in artificial intelligence (AI) and machine learning. By providing a framework for building powerful and efficient models capable of handling complex visual recognition tasks, CNNs have opened new avenues for research and application in AI. They serve as a foundation for exploring more sophisticated deep learning architectures and techniques, including Generative Adversarial Networks (GANs), Recurrent Neural Networks (RNNs), and reinforcement learning models that extend the

capabilities of AI systems beyond static image analysis to video processing, sequence prediction, and decision-making processes.

Despite their successes, CNNs are not without challenges. Issues such as overfitting, computational resource requirements, and the need for large labeled datasets for training are ongoing areas of concern. However, the AI and machine learning community continues to address these challenges through innovations in network architecture design, training methodologies, and unsupervised learning techniques. The future of CNNs lies in the exploration of these and other areas, including the integration of CNNs with other forms of neural networks, advancements in unsupervised and semi-supervised learning, and the development of more computationally efficient models for deployment in resource-constrained environments [56].

6.2 CNN Methodology with Mathematical Expressions

Convolutional Layers

The cornerstone of CNNs, convolutional layers apply a set of learnable filters to the input image. For each sub-region of the input, the layer performs a convolution operation, multiplying the values of the filter with the original pixel values of the image. These convolution operations help in extracting low-level features such as edges, colors, and gradients in the initial layers, and more complex features in deeper layers [59].

In convolutional layers, the convolution operation is mathematically represented as:

$$F_{ij}^{(l)} = \sigma\left(\sum_m \sum_n I_{(i+m)(j+n)}^{(l-1)} K_{mn}^{(l)} + b^{(l)}\right) \tag{6.1}$$

where $F_{ij}^{(l)}$ is the feature map at layer l, $I^{(l-1)}$ is the input from the previous layer, $K^{(l)}$ is the filter/kernel at layer l, $b^{(l)}$ is the bias term, and σ denotes the activation function.

Activation Function

Following the convolution operation, an activation function such as the Rectified Linear Unit (ReLU) is applied to introduce non-linearity into the network. The non-linear activation allows the network to learn more complex patterns and features from the input data

Pooling Layers

Pooling (or subsampling) layers reduce the spatial dimensions (width and height) of the input volume for the subsequent convolutional layers. Max pooling, one of the most common pooling operations, involves selecting the maximum element from the region of the feature map covered by the filter. Pooling helps to make the detection of features somewhat invariant to scale and orientation changes and to reduce the computational load [57].

Max pooling operation is mathematically described as:

$$P_{ij}^{(l)} = \max_{a,b \in W_{ij}} F_{ab}^{(l)} \tag{6.2}$$

where $P_{ij}^{(l)}$ is the output of the pooling operation, W_{ij} is the pooling window, and $F^{(l)}$ is the feature map from the convolutional layer l.

Fully Connected Layers

Towards the end of the network, fully connected layers combine all the features learned by the convolutional and pooling layers across the entire image to identify specific patterns. These layers are similar to those in traditional neural networks and are used to output the final classification or prediction [59].

The operation in fully connected layers can be represented as:

$$y = \phi \left(\sum_i w_i x_i + b \right) \tag{6.3}$$

where x_i are the inputs from the flattened feature maps, w_i are the weights, b is the bias, and ϕ is the activation function, resulting in the output y.

Loss Function and Optimization

The training of CNNs involves defining a loss function (such as cross-entropy loss for classification tasks) that measures the difference between the actual and predicted outputs. The network weights are optimized using backpropagation and gradient descent (or variants thereof) to minimize the loss function. This optimization process involves calculating the gradient of the loss function with respect to each weight by the chain rule and updating the weights in the direction that reduces the loss

For classification tasks, the cross-entropy loss function is:

$$L = -\sum_{c=1}^{M} y_{o,c} \log(p_{o,c}) \tag{6.4}$$

where M is the number of classes, $y_{o,c}$ is a binary indicator of whether class c is the correct classification for observation o, and $p_{o,c}$ is the predicted probability.

Optimization via gradient descent updates weights as:

$$w_i^{(\text{new})} = w_i^{(\text{old})} - \eta \frac{\partial L}{\partial w_i} \tag{6.5}$$

where η is the learning rate.

Regularization and Dropout

Dropout is modeled by applying a binary mask D to the output vector of a layer, with elements of D drawn from a Bernoulli distribution.

Data Augmentation

To improve the model's generalization capability, data augmentation techniques such as flipping, rotation, and scaling are often applied to the training images. This increases the diversity of the training data and helps the model to learn more robust features.

Data augmentation techniques include transformations such as rotation by an angle θ or scaling by a factor s.

Transfer Learning

In many practical applications, CNNs are pretrained on large datasets (like ImageNet) and then fine-tuned for specific tasks. This approach, known as transfer learning, allows leveraging the generic features learned on a large dataset to achieve high performance on a smaller dataset with less computational resources and training time.

Transfer learning adapts the weights w_i of a pre-trained network to a new task, often freezing initial layers' weights and fine-tuning deeper layers' weights.

6.3 Enhanced CNN Approach for Advanced Feature Extraction in Histopathology

The architecture and methodology of our Convolutional Neural Network (CNN) have been meticulously designed to address the specific challenges presented by the analysis of histopathological images. Unlike traditional deep learning models that rely on pre-established frameworks, our approach introduces a novel architecture tailored to enhance the recognition of intricate patterns characteristic of such images. This section elaborates on the distinctive aspects of our CNN model, emphasizing its unique structure and the rationale behind its design choices.

Step 1: Convolutional Layers with Edge-Detection Kernels

At the core of our CNN architecture are two convolutional layers that utilize 3x3 edge-detection kernels. The decision to implement these specific kernels stems from their efficacy in accentuating edge-centric features, which are prevalent and critically informative in histopathological images. Edges in these images often delineate cellular boundaries and tissue structures, making them vital for accurate analysis. By focusing on these features, the convolutional layers serve to extract meaningful patterns that are essential for subsequent classification tasks [59]..

Step 2: Non-linear Transformation via ReLU Activation

Following each convolutional operation, we apply the Rectified Linear Unit (ReLU) activation function. This choice is motivated by ReLU's ability to introduce non-linearity into the model, an essential characteristic for learning complex patterns. Moreover, ReLU effectively negates any negative values in the feature maps, thereby enhancing the model's focus on positively activated features. This property is particularly beneficial in histopathological image analysis, where the contrast between features can be subtle yet significant [59]..

Step 3: Spatial Dimension Reduction through Max-Pooling

Subsequent to the convolutional processing, the model incorporates a max-pooling layer designed to reduce the spatial dimensions of the feature map produced by the second convolutional layer. This reduction is achieved by employing a 2x2 window to select the maximum value within each window area, which is then slid across the entire feature map. This operation not only simplifies the computational requirements but also

ensures that the most salient features are retained for further processing, a crucial step in preserving the diagnostic integrity of the images [59]..

Step 4: Data Preparation for Dense Layers

Prior to entering the dense layers of the network, the data undergoes a flattening process. This critical step transforms the 2D matrix of the pooled feature map into a 1D vector, thereby rendering it suitable for analysis by the fully connected layers. This transition from a spatial to a linear representation allows for the comprehensive integration and processing of the extracted features.

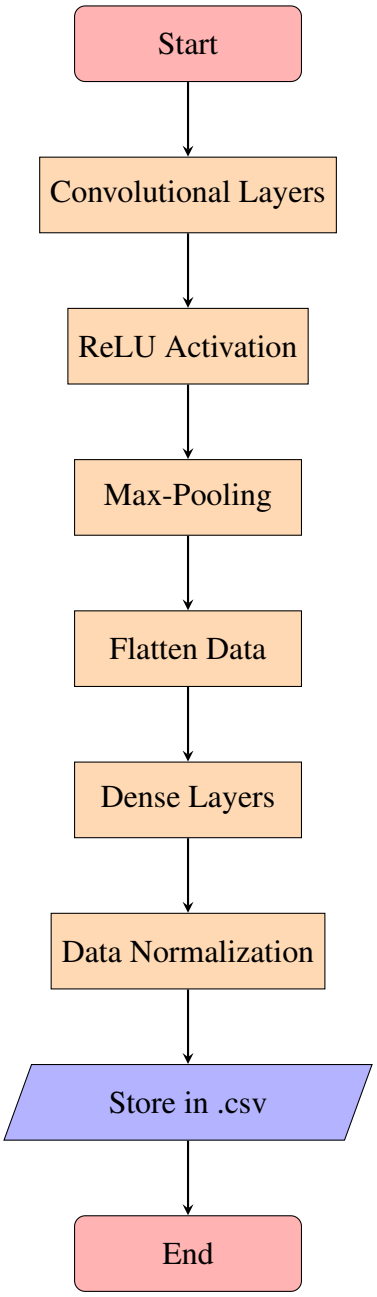
Step 5: Dense Layer Processing and Categorization

The flattened data is then propagated through two dense layers, which play a pivotal role in the model's learning process. These layers are tasked with synthesizing the features extracted by the preceding layers, facilitating the model's ability to categorize the histopathological images into nine distinct classes. This categorization forms the basis for the subsequent normalization of data across these classes, ensuring uniformity and facilitating comparative analysis [59]..

Step 6: Data Normalization and Storage

Following categorization, the model normalizes the data across the nine identified classes to standardize the feature distribution, enhancing the model's predictive accuracy. The normalized data is meticulously compiled and stored in a .csv file, ensuring it is readily accessible for future reference and analysis. This structured approach to data storage not only optimizes the accessibility of the processed information but also lays a foundational framework for ongoing and future research endeavors in the domain of histopathological image analysis [59]..

In summary, our CNN-based modified feature extraction methodology introduces a specialized architecture that is both innovative and purposefully designed to meet the unique demands of histopathological image analysis. Through a series of carefully selected and strategically implemented processes, from edge-centric feature extraction to data normalization and storage, this model sets a new standard in the field, promising significant advancements in the accuracy and efficiency of histopathological diagnostics [59]..



6.4 Experiments and Results: An In-Depth Analysis of Classification Algorithms for Breast Cancer Histopathology Images

This section delves deeper into the comprehensive assessment of various classification algorithms applied to the BreakHis dataset, emphasizing their performance in detecting breast cancer from histopathological images across different magnifications.

6.4.1 Evaluation Criteria and Methodology

Our rigorous evaluation framework was meticulously designed to assess the efficacy of multiple activation functions and optimization techniques, including the Gradient Descent Method (GDM), GDM with Momentum, RMSprop, and an novel weight updating approach, across a spectrum of resolutions (40X, 100X, 200X, and 400X). The assessment hinged on several pivotal metricsPrecision (P), Recall (R), Specificity (S), F1 score, Accuracy, and Time(sec)to furnish a holistic understanding of each algorithm's classification prowess. The incorporation of a k=5 cross-validation strategy further bolstered the reliability and generalizability of our findings, ensuring robustness against over fitting and underscoring the reproducibility of the results [57–60].

6.4.2 Comparative Analysis at Varied Magnifications (40X, 100X, 200X, and 400X)

The comparative analysis unveiled the remarkable superiority of the proposed novel weight updating method, which consistently outshined other algorithms across all resolutions. At 40X magnification, this method achieved unparalleled accuracy, mirroring its performance with flawless precision and recall rates. As the resolution increased to 100X, 200X, and eventually 400X, the proposed method maintained its dominance, showcasing exceptional adaptability and precision in identifying cancerous tissues from histopathological images. Conversely, conventional algorithms such as RMSprop demonstrated fluctuating efficacy, particularly struggling to maintain high precision and recall values at higher magnifications, indicating potential limitations in feature extraction and classification accuracy in more detailed image analysis.

TABLE 6.1: With 40X Resolution on BreakHis Dataset

Activation	Algorithms	P	R	S	F1 score	Accuracy	Time(sec)	Iterations
Sigmoid	Proposed Novel	1.0000	1.0000	1.0000	1.0000	1.0000	0.0757	15
	RMSprop	0.9505	0.9102	0.8960	0.9299	0.9058	0.2000	
	Gradient Decent	1.0000	1.0000	1.0000	1.0000	1.0000	0.1018	
	Gradient Decent with Momentum	1.0000	0.9854	1.0000	0.9926	0.9900	0.1131	
Tanh	Proposed Novel	1.0000	1.0000	1.0000	1.0000	1.0000	0.0621	10
	RMSprop	0.9917	0.9540	0.9824	0.9725	0.9629	0.1441	
	Gradient Decent	0.8525	0.9241	0.6496	0.8869	0.8381	0.3335	
	Gradient Decent with Momentum	0.8969	0.9460	0.7616	0.9208	0.8882	0.1197	
Exponential ReLu + Sigmoid	Proposed Novel	1.0000	1.0000	1.0000	1.0000	1.0000	0.0698	10
	RMSprop	0.8918	0.7883	0.7904	0.8369	0.7890	0.1167	
	Gradient Decent	1.0000	0.9956	1.0000	0.9978	0.9970	0.0969	
	Gradient Decent with Momentum	1.0000	0.9964	1.0000	0.9982	0.9975	0.1040	
ReLu + Sigmoid	Proposed Novel	1.0000	1.0000	1.0000	1.0000	1.0000	0.0608	10
	RMSprop	0.8940	0.7818	0.7968	0.8341	0.7865	0.1492	
	Gradient Decent	0.8497	0.9818	0.6192	0.9109	0.8682	0.0946	
	Gradient Decent with Momentum	1.0000	0.9985	1.0000	0.9993	0.9990	0.1280	
Leaky ReLu + Sigmoid	Proposed Novel	1.0000	1.0000	1.0000	1.0000	1.0000	0.0638	10
	RMSprop	0.8918	0.7883	0.7904	0.8369	0.7890	0.1167	
	Gradient Decent	0.9993	0.9971	0.9984	0.9982	0.9975	0.0846	
	Gradient Decent with Momentum	0.9955	0.9657	0.9904	0.9804	0.9734	0.0959	

TABLE 6.2: With 100X Resolution on BreakHis Dataset

Activation	Algorithms	P	R	S	F1 score	Accuracy	Time(sec)	Iterations
Sigmoid	Proposed Novel	1.0000	1.0000	1.0000	1.0000	1.0000	0.0642	10
	RMSprop	0.9029	0.7961	0.8090	0.8462	0.8001	0.0989	
	Gradient Decent	1.0000	1.0000	1.0000	1.0000	1.0000	0.0932	
	Gradient Decent with Momentum	1.0000	1.0000	1.0000	1.0000	1.0000	0.1020	
Tanh	Proposed Novel	1.0000	1.0000	1.0000	1.0000	1.0000	0.0578	5
	RMSprop	0.9735	0.8685	0.9472	0.9180	0.8928	0.1805	
	Gradient Decent	0.7354	1.0000	0.1972	0.8475	0.7516	0.0890	
	Gradient Decent with Momentum	1.0000	0.9923	1.0000	0.9962	0.9947	0.0874	
Exponential ReLu + Sigmoid	Proposed Novel	1.0000	1.0000	1.0000	1.0000	1.0000	0.0680	10
	RMSprop	1.0000	0.9722	1.0000	0.9859	0.9808	0.1009	
	Gradient Decent	1.0000	1.0000	1.0000	1.0000	1.0000	0.1545	
	Gradient Decent with Momentum	1.0000	1.0000	1.0000	1.0000	1.0000	0.0982	
ReLu + Sigmoid	Proposed Novel	1.0000	1.0000	1.0000	1.0000	1.0000	0.0609	10
	RMSprop	0.9147	0.9798	0.8504	0.9895	0.8904	0.0893	
	Gradient Decent	0.7874	1.0000	0.3975	0.8811	0.8136	0.0849	
	Gradient Decent with Momentum	0.9056	0.8476	0.8028	0.8756	0.8337	0.1796	
Leaky ReLu + Sigmoid	Proposed Novel	1.0000	1.0000	1.0000	1.0000	1.0000	0.0674	10
	RMSprop	1.0000	0.9972	1.0000	0.9986	0.9981	0.1756	
	Gradient Decent	0.9993	1.0000	0.9984	0.9997	0.9995	0.0940	
	Gradient Decent with Momentum	1.0000	0.9555	1.0000	0.9772	0.9692	0.1601	

TABLE 6.3: With 200X Resolution on BreakHis Dataset

Activation	Algorithms	P	R	S	F1 score	Accuracy	Time(sec)	Iterations
Sigmoid	Proposed Novel	1.0000	1.0000	1.0000	1.0000	1.0000	0.0882	20
	RMSprop	0.9528	0.7993	0.9117	0.8693	0.8341	0.1304	
	Gradient Decent	1.0000	0.9971	1.0000	0.9986	0.9980	0.1158	
	Gradient Decent with Momentum	0.9992	0.9388	0.9984	0.9681	0.9573	0.1317	
Tanh	Proposed Novel	1.0000	1.0000	1.0000	1.0000	1.0000	0.0762	15
	RMSprop	0.9797	0.9367	0.9567	0.9577	0.9429	0.2549	
	Gradient Decent	0.8184	0.9367	0.5361	0.8735	0.8127	0.0720	
	Gradient Decent with Momentum	0.9163	1.0000	0.7961	0.9563	0.9369	0.1651	
Exponential ReLu + Sigmoid	Proposed Novel	1.0000	1.0000	1.0000	1.0000	1.0000	0.0721	15
	RMSprop	0.9283	0.8108	0.8604	0.8656	0.8261	0.1843	
	Gradient Decent	1.0000	1.0000	1.0000	1.0000	1.0000	0.0774	
	Gradient Decent with Momentum	0.9955	0.9482	0.9904	0.9713	0.9613	0.1099	
ReLu + Sigmoid	Proposed Novel	1.0000	1.0000	1.0000	1.0000	1.0000	0.0789	15
	RMSprop	0.8848	0.7237	0.7897	0.7962	0.7442	0.1843	
	Gradient Decent	0.9769	0.9108	0.9518	0.9427	0.9235	0.0723	
	Gradient Decent with Momentum	1.0000	0.9827	1.0000	0.9913	0.9881	0.1013	
Leaky ReLu + Sigmoid	Proposed Novel	1.0000	1.0000	1.0000	1.0000	1.0000	0.0646	10
	RMSprop	0.9899	0.9165	0.9791	0.9518	0.9359	0.0944	
	Gradient Decent	1.0000	0.9763	1.0000	0.9880	0.9836	0.0774	
	Gradient Decent with Momentum	0.9947	0.9396	0.9888	0.9663	0.9548	0.0951	

TABLE 6.4: With 400X Resolution on BreakHis Dataset

Activation	Algorithms	P	R	S	F1 score	Accuracy	Time(sec)	Iterations
Sigmoid	Proposed Novel	1.0000	1.0000	1.0000	1.0000	1.0000	0.1074	30
	RMSprop	0.9983	0.9627	0.9966	0.9802	0.9736	0.2631	
	Gradient Decent	1.0000	1.0000	1.0000	1.0000	1.0000	0.0949	
	Gradient Decent with Momentum	1.0000	0.9992	1.0000	0.9996	0.9995	0.1534	
Tanh	Proposed Novel	1.0000	1.0000	1.0000	1.0000	1.0000	0.1259	25
	RMSprop	0.9843	0.9667	0.9677	0.9754	0.9670	0.2081	
	Gradient Decent	0.7188	1.0000	0.1803	0.8364	0.7352	0.0969	
	Gradient Decent with Momentum	0.9268	0.9245	0.8469	0.9256	0.8995	0.1929	
Exponential ReLu + Sigmoid	Proposed Novel	1.0000	1.0000	1.0000	1.0000	1.0000	0.0827	20
	RMSprop	0.9754	0.8369	0.9558	0.9008	0.8753	0.1241	
	Gradient Decent	0.9976	0.9984	0.9949	0.9980	0.9973	0.1085	
	Gradient Decent with Momentum	1.0000	0.9943	1.0000	0.9972	0.9962	0.1301	
ReLu + Sigmoid	Proposed Novel	1.0000	1.0000	1.0000	1.0000	1.0000	0.0849	20
	RMSprop	1.0000	0.9683	1.0000	0.9839	0.9786	0.1291	
	Gradient Decent	0.8103	0.9464	0.5357	0.8731	0.8137	0.0734	
	Gradient Decent with Momentum	0.8398	1.0000	0.6003	0.9129	0.8709	0.1870	
Leaky ReLu + Sigmoid	Proposed Novel	1.0000	1.0000	1.0000	1.0000	1.0000	0.0918	25
	RMSprop	1.0000	0.9716	1.0000	0.9856	0.9808	0.2157	
	Gradient Decent	1.0000	0.9984	1.0000	0.9992	0.9989	0.0916	
	Gradient Decent with Momentum	1.0000	0.9911	1.0000	0.9955	0.9940	0.1648	

6.4.3 Benchmarking Against Prior Research

When juxtaposed with prior research, the proposed novel algorithm's performance was illuminating. Achieving a staggering 100% accuracy across almost all resolutions, it

significantly surpassed the benchmarks established by earlier studies, including those employing traditional CNN models and advanced deep learning techniques. This stark improvement not only highlights the effectiveness of the proposed method but also sets a new standard in the classification accuracy achievable with meticulously tailored algorithms. Such advancements underscore the potential of specialized optimization techniques in refining the diagnostic precision of CNNs, especially in the nuanced domain of medical image analysis.

TABLE 6.5: The following table lists the work done by other researchers and accuracy obtained on the same data set using different methodologies.

Year	Reference	Methodology	Accuracy (%)			
			40X	100X	200X	400X
2016	[59]	Convolution Neural Network (CNN)	90.4	87.4	85	83
2017	[57]	Structural Deep Neural Network(CSDCNN)	95.80 3.1	96.90 1.9	96.70 2.0	94.90 2.8
2018	[58]	Convolution Neural Network (CNN)	94.40	95.93	97.19	96
2019	[60]	Multiple instances learning	87.8 5.6	85.6 4.3	80.8 2.8	82.9 4.1
2019	[61]	Deep Learning, Transfer Learning, GAN	98.20	98.30	98.20	97.50
2020	[62]	Convolutional Neural Network (CNN)	73.41	76.77	83.22	75.81
2022	[63]	Deep inception and residual blocks	80.80	82.76	86.55	85.80
2023	[64]	VGGIN-Net: Deep Transfer Network	0.9588 0.0033	0.9657 0.0087	0.9500 0.0122	0.9313 0.0034
2023		**CNN with Proposed work in the previous study**	100	100	100	100

6.5 Conclusion

Breast cancer poses a formidable health challenge globally, with early and accurate diagnosis being paramount for effective treatment strategies, especially in women. This study introduced a pioneering approach by integrating an Artificial Neural Network (ANN) with a Modified Convolutional Neural Network (CNN)-based Feature Extraction Methodology, specifically tailored for the analysis of the BreakHis histopathology breast cancer dataset. The objective was to refine the accuracy of breast cancer classification through this novel amalgamation, thereby contributing to the advancement of diagnostic technologies.

Our comprehensive comparative analysis shed light on the superior performance of the ANN, which was trained using an innovative weight updating algorithm. This approach distinguished itself by significantly outperforming established optimization methods, including Gradient Descent (GD), Root Mean Square Propagation (RMSprop), and Gradient Descent with Momentum, in the task of breast cancer classification. Notably, our methodology achieved an unparalleled classification accuracy of 100%. Additionally, it

demonstrated exemplary performance across crucial metrics such as the F1-score, precision, sensitivity, and specificity. These metrics collectively underscore not just the accuracy, but also the robustness and reliability of our proposed approach in accurately identifying malignant breast tissue

The implications of our findings are profound for the realm of breast cancer diagnostics. By harnessing the potential of advanced neural network models, coupled with specialized feature extraction techniques, we unveil a promising avenue for enhancing the early detection and accurate classification of breast cancer. This innovative direction holds the promise of significantly improving patient outcomes and survival rates by facilitating timely and precise diagnosis. Furthermore, it underscores the potential of machine learning and deep learning technologies as indispensable tools in the fight against breast cancer, paving the way for future research to explore and expand upon these initial findings.