Experimental Evaluation of CNN Optimizers in MRI-based Breast Cancer Detection

Ally Halid: Orange County School of The Arts
Mentors: Professor Mohammad Husain and Antoine Si

Abstract

Cancer is one of the leading causes of death globally [1], with female breast cancer being one of the most common types of cancer [2]. Early-stage detection is crucial for improving treatment efficiency and patient prognosis. Due to factors such as human error and limited resources such as time, cancer detection via manually examining MRI images can be resource intensive and inefficient. Optimizing the application of machine learning in the field of cancer detection can minimize the time needed to identify cancer, which can lead to earlier diagnoses in patients.

Approach

A popular approach for applying machine learning to medical imagery for diagnosis includes the use of neural networks such as the Convolutional Neural Network (CNN) which is a machine learning model that can be trained with image datasets to help analyze tumors in medical scans. CNNs are made up of layers, including convolutional and pooling layers. for feature extraction [3] ,which utilizes and learns patterns within the data (in this case the medical scans) to help the neural network recognize these patterns in other images. CNNs, once trained, can be used for classification of tumors such as benign/malignant classification. By optimizing CNN architecture and experimenting with optimizers, higher classification accuracy can be achieved, leading to more reliable methods of using machine learning to aid in cancer diagnosis. Overfitting, which is when the CNN model cannot generalize to new data inputs after training, leading to low validation accuracies, was addressed as well when adjusting the CNN model. In order to combat this, methods such using regularizers can be employed. Regularizers, help with generalization, and L1 and L2 regularizers (utilized in this experiment) help to regulate and reduce the weight that the model places on features. Optimizers aim to reduce the loss between the actual and expected output by changing the weights that the network places. In this experiment, a dataset containing MRIs of breast tumors was used to train a CNN model. The CNN model was then used as a base to compare the validation accuracies of different optimizers.

Methodology

Preprocessing for this experiment included normalizing the pixel values of the grayscale images by dividing the values by 255.0 so that the pixel values were between 0.0-1.0. The images of the dataset were also preprocessed, as the images' dimensiolns were already resized on Kaggle by Benjelloun [4] to be 224x224 pixels. CNN model architecture from Google Colab using Tensorflow that was modified was fed images from The Cancer Imaging Archive (TCIA) [5] accessed via Kaggle. Different optimizers, including Adam, AdamW, Adamax, and Adagrad, were then applied, and the accuracies of the model with each optimizer were compared.

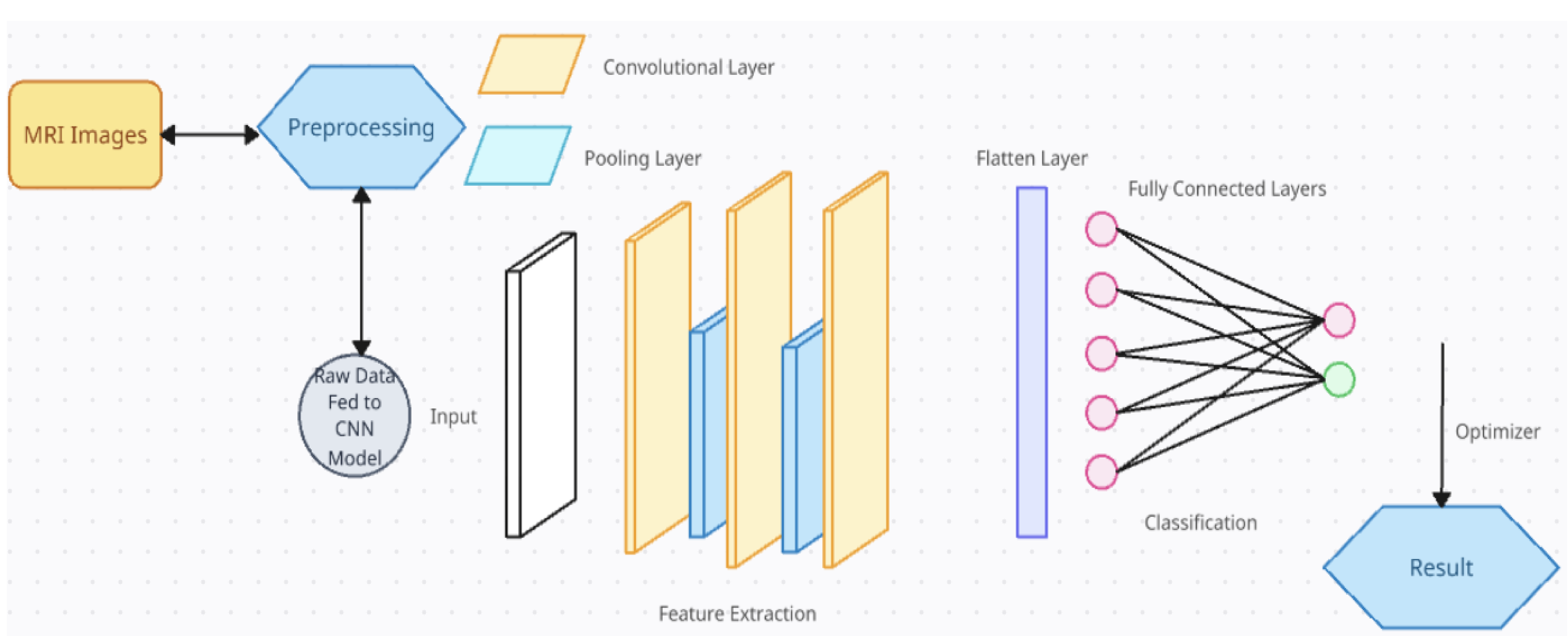

Figure 3: Model Process Visualization

Dataset

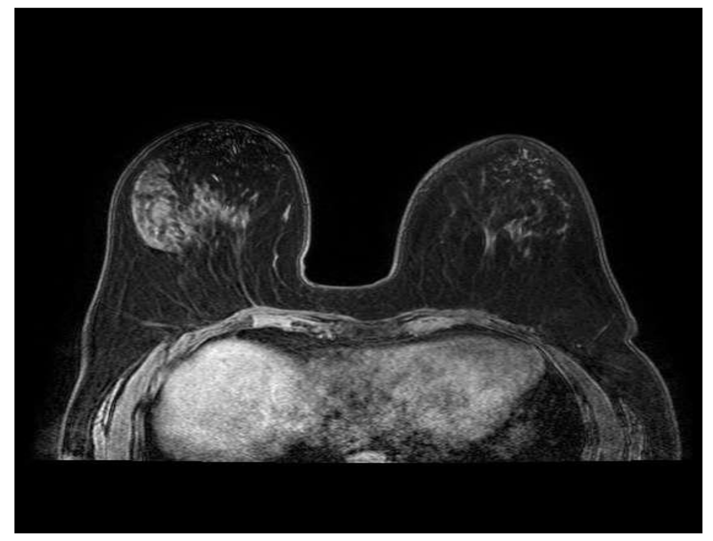

Figure 1: MRI of Benign Tumor [2]

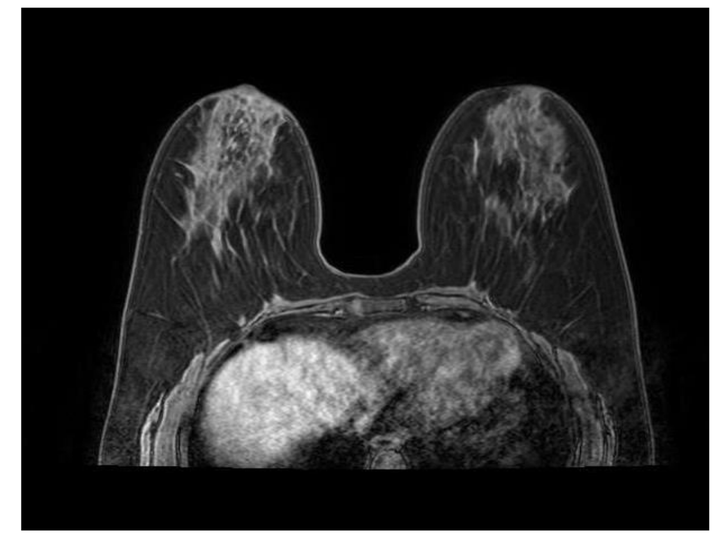

Figure 2: MRI of Malignant Tumor [2]

Model Architecture

Layer (type)	Output Shape	Param #
conv2d_3 (Conv2D)	(None, 222, 222, 32)	320
batch_normalization_3 (BatchNormalization)	(None, 222, 222, 32)	128
max_pooling2d_2 (MaxPooling2D)	(None, 111, 111, 32)	0
conv2d_4 (Conv2D)	(None, 109, 109, 64)	18,496
batch_normalization_4 (BatchNormalization)	(None, 109, 109, 64)	256
max_pooling2d_3 (MaxPooling2D)	(None, 54, 54, 64)	0
conv2d_5 (Conv2D)	(None, 52, 52, 64)	36,928
batch_normalization_5 (BatchNormalization)	(None, 52, 52, 64)	256
flatten_1 (Flatten)	(None, 173056)	0
dense_2 (Dense)	(None, 64)	11,075,648
dropout_1 (Dropout)	(None, 64)	0
dense_3 (Dense)	(None, 1)	65

Figure 4: CNN Model Architecture

Results

For this specific image dataset of breast cancer MRIs, the Adagrad optimizer applied to the CNN model led to marginally higher validation accuracies, with an average 99.432% final validation accuracy in comparison to other optimizers such as Adadelta. However, the Adamax, AdamW and Adam optimizer resulted in similar average final validation accuracy of 99.396%, 99.21% and 98.96% respectively.

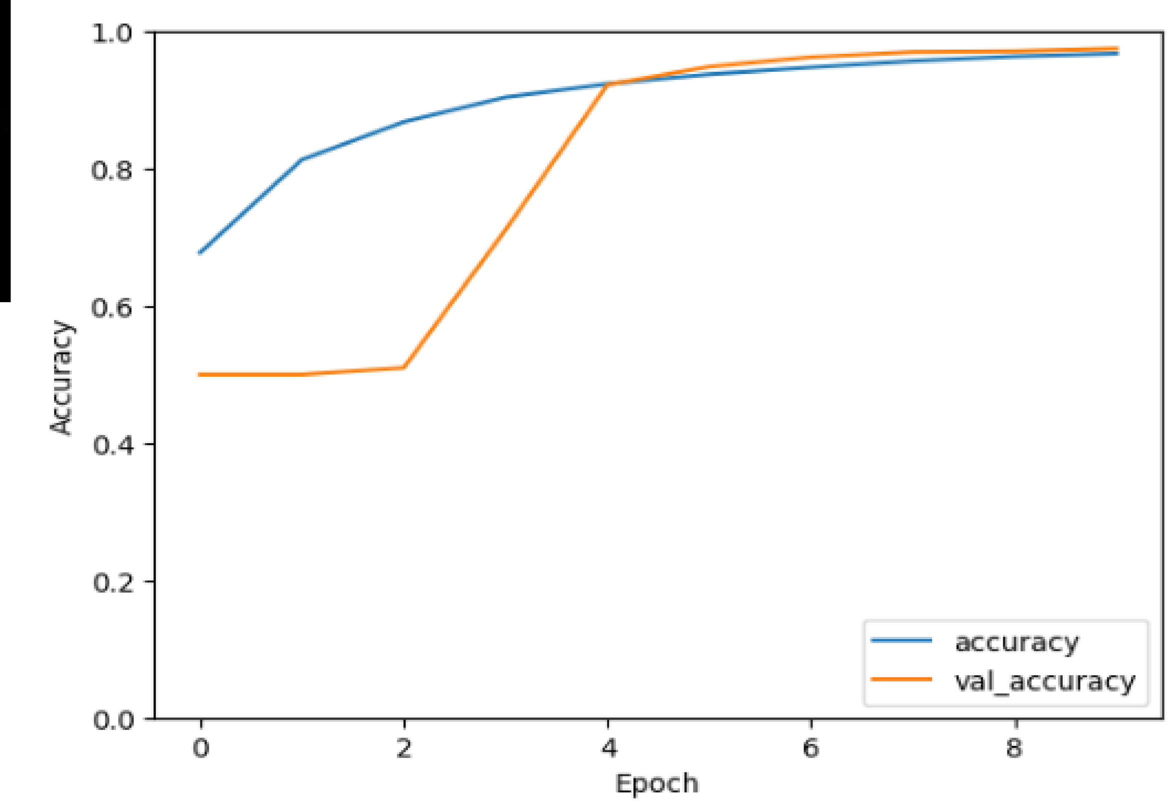

Figure 5: Training and Validation Accuracy with Adadelta Optimizer
Validation accuracy steadily increases but in comparison to other optimizers its validation accuracy isn't as close to 100%.

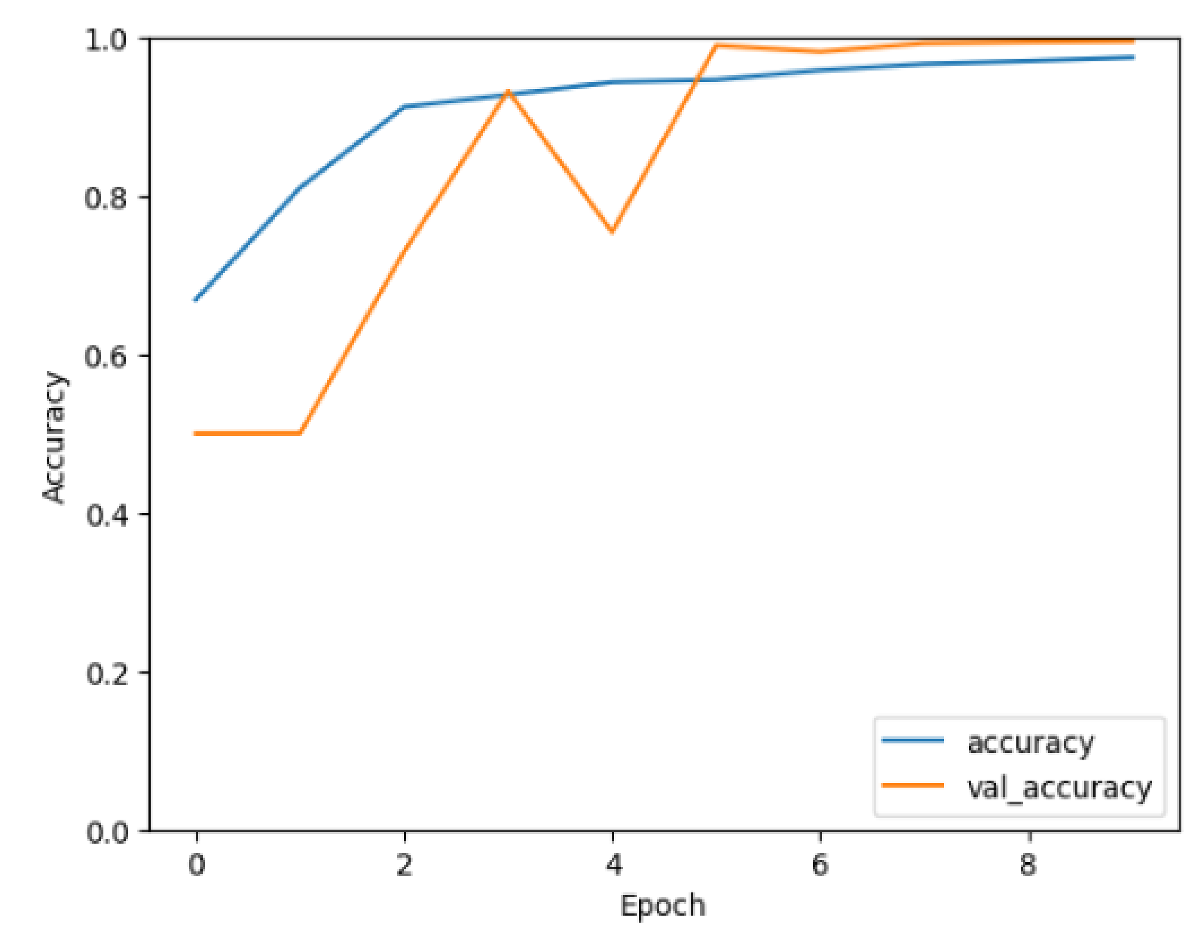

Figure 6: Training and Validation Accuracy with 'Adam' Optimizer
Validation accuracy presents fluctuation before stabilizing around epoch 5 and demonstrating accuracy close to 100%

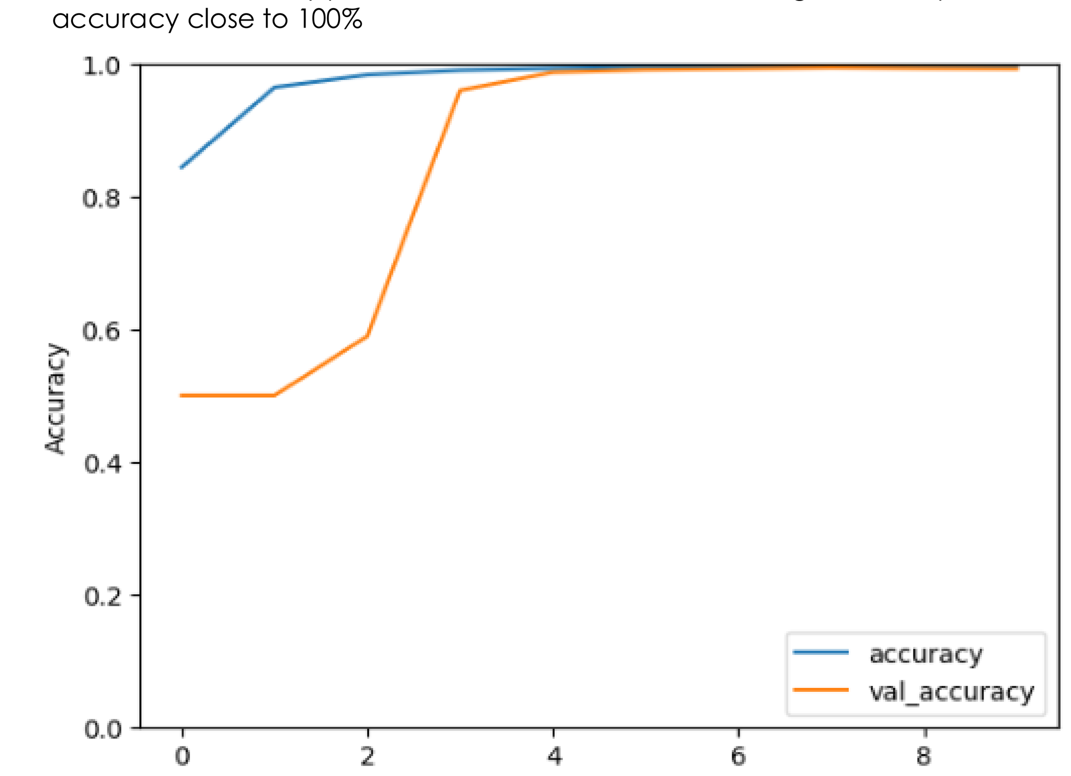

Figure 7: Testing and Validation Accuracy with Adagrad Optimizer
Validation accuracy demonstrates a near 100% accuracy after the 4th epoch without fluctuating

Results Summary

Optimizer	Average Validation Accuracy
Adadelta	96.988
Adam	98.96
AdamW	99.21
Adamax	99.396
Adagrad	99.432

Figure 8: Average Validation Accuracies of Optimizers
The Adagrad optimizer demonstrates the highest average validation accuracy

Challenges

During this experiment, the model's validation would stagnate around 50% without improving, due to overfitting. In order to prevent this, methods such as employing L1 and L2 regularizers were utilized. Adjusting the parameters (to decrease the learning rate) for the kernel L1 and L2 regularizers to 1e^-3 and $1e^{-4}$ respectively from $1e^{-2}$, and 1e-3, and the bias and activity L2 regularizers from $1e^{-2}$ to $1e^{-3}$ aided in reducing stagnation significantly.

The model also exhibit a case of dropping off in validation accuracy around the last epoch, so to combat this, early stopping was utilized. Drop out layers and batch normalization layers were added as well as part of the CNN model architecture to combat overfitting. Occasionally, the validation accuracy would fluctuate for the model but for the most part consistently increased after applying these modifications. Thus, after addressing overfitting, the optimizers' effectiveness could then be evaluated properly.

References

[1] Saloni Dattani, Fiona Spooner, Hannah Ritchie, and Max Roser (2023) - "Causes of Death" Published online at OurWorldinData.org. Retrieved from: 'https://ourworldindata.org/causes-of-death' [Online Resource]

[2] World Cancer Research Fund. Global Cancer Observatory. "Worldwide Cancer Data" https://www.wcrf.org/preventing-cancer/cancer-statistics/worldwide-cancer-data/

[3] Oliveira, Spanhol, Petitjean, and Heutte, (2016). Breast Cancer Histopathological Image Classification using Convolutional Neural Networks. Retrieved from https://www.inf.ufpr.br/lesoliveira/download/IJCNN2016-BC.pdf

[4] Benjelloun. Kaggle. "Breast MRI Tumor Classification Dataset" Accessed 2025. https://www.kaggle.com/datasets/abenjelloun/breast-mri-tumor-classification-dataset/data

[5] Bloch, B. Nicolas, Jain, Ashali, & Jaffe, C. Carl. (2015). BREAST-DIAGNOSIS [Data set]. The Cancer Imaging Archive. http://doi.org/10.7937/K9/TCIA.2015.SDNRQXXR