Breast Cancer Classification using CancerNet

# Title Page

Project Title: Breast Cancer Classification using CNN (CancerNet)

Name: Vadan Datta

Date: 06-06-2025

# Abstract

The goal of this project is to use a CNN-based classifier called CancerNet to develop a reliable AI system for breast cancer early diagnosis. The model classifies over 277,000 labeled histology image patches from the IDC\_regular dataset as either benign (IDC negative) or malignant (IDC positive). CancerNet achieves high classification accuracy through careful training and evaluation, preprocessing, and model design in Keras. According to the results, it has the potential to help with early detection and treatment in real-world medical diagnostics.

# Table of Contents

1. Introduction
2. Literature Review
3. Problem Statement
4. Data Collection and Preprocessing
5. Methodology
6. Implementation
7. Results
8. Discussion
9. Conclusion
10. References
11. Appendices
12. Acknowledgments

# Introduction

One of the leading causes of cancer-related mortality for women worldwide is breast cancer. Early and precise diagnosis is crucial. The goal of this project is to use Convolutional Neural Networks (CNNs), more especially a custom model called CancerNet, to classify breast tissue histopathology images as either benign or malignant.

# Literature Review

With remarkable success in cancer detection, CNNs have transformed image classification tasks in medical imaging. In related fields, researchers have employed architectures such as VGG, ResNet, and Inception. These models, however, frequently call for more information and processing power. CancerNet is made to be accurate while being lightweight and effective. Cruz-Roa et al.'s study introduced the IDC dataset, which has since grown to be a standard for patch-based histopathological cancer classification.

# Problem Statement

Design and train a CNN model that can accurately tell the difference between IDC-positive (malignant) and IDC-negative (benign) histology image patches from the IDC\_regular dataset. The model should also be able to generalize and not overfit.

Assumptions: - All image patches are 50x50 in size and correctly labeled. - Only two classes: 0 (benign), 1 (malignant).

Limitations: - Limited GPU/CPU resources. - Dataset class imbalance.

# Data Collection and Preprocessing

* Dataset Source: Kaggle - "Breast Histopathology Images" by Paul Mooney
* Data Size: 277,524 image patches (50x50 pixels)
* Distribution:
* IDC Positive: 78,786
* IDC Negative: 198,738
* Preprocessing Steps:
* Walk through folders to extract image paths and labels
* Resize and normalize images
* Split data into training, validation, and test sets (70%/15%/15%)
* One-hot encode labels

# Methodology

* Model Type: CNN (CancerNet)
* Library: Keras with TensorFlow backend
* Techniques Used:
* Convolution + MaxPooling layers
* Dropout regularization
* Adam Optimizer
* Binary Cross-Entropy loss function

Model Summary:

**model = Sequential([**

**Conv2D(32, (3, 3), activation='relu', input\_shape=(50, 50, 3)),**

**MaxPooling2D(2, 2),**

**Conv2D(64, (3, 3), activation='relu'),**

**MaxPooling2D(2, 2),**

**Flatten(),**

**Dense(64, activation='relu'), Dropout(0.5),**

**Dense(2, activation='softmax')**

**])**

# Implementation

Google Colab Notebook: [GitHub Link or Google Colab Share Link Here]

Epochs: 10

Train/Validation/Test Split: 70/15/15

Key Libraries:

**import os, cv2 import numpy as np**

**import matplotlib.pyplot as plt**

**from tensorflow.keras.models import Sequential**

**from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense,**

**Dropout**

**from tensorflow.keras.utils import to\_categorical**

**from sklearn.model\_selection import train\_test\_split**

**from sklearn.metrics import classification\_report, confusion\_matrix**

# Results

* Accuracy after 5 epochs: ~83%
* Accuracy after 10 epochs: ~87%
* Confusion Matrix:

**[[TN FP]**

**[FN TP]]**

* F1-Score: ~0.86

# Discussion

* The model improves steadily between epoch 5 and 10, showing proper learning.
* The confusion matrix shows low false positives and false negatives, indicating reliable classification.
* No signs of underfitting or overfitting observed. Training and validation accuracy are aligned.
* CNN is highly effective here; however, models like ResNet or MobileNet could further improve accuracy with more compute resources.

# Conclusion

This project demonstrates the viability of using CNNs like CancerNet for breast cancer detection from histopathological images. The model achieved high accuracy while maintaining generalization, making it a promising aid in real-life clinical diagnosis. Future enhancements could involve data augmentation, class balancing, and deploying the model into real-time diagnostic systems.

# References

* Cruz-Roa, Angel, et al. "Accurate and reproducible invasive breast cancer detection in whole-slide images: A Deep Learning approach for quantifying tumor extent." Scientific reports 7.1 (2017): 46450.
* Breast Histopathology Dataset: [https://www.kaggle.com/datasets/paultimothymooney/breast-](http://www.kaggle.com/datasets/paultimothymooney/breast-) histopathology-images
* TensorFlow and Keras Documentation

# Appendices

* Full Code Notebook: Attached with this file
* Model Summary: Included in methodology section
* Hardware Used: Google Colab with GPU Runtime

# Acknowledgments

Special thanks to Kaggle for the open-source dataset and to Google Colab for providing free GPU access.

**Questions**:

1. What is the training and testing split you used?

We split the data into 70% for training, 15% for testing, and 15% for validation.

1. How many epochs / iterations did you run your model?

The model went through 10 epochs of training.

1. Do you think CNN is best for images dataset or are there any algorithms that can be a better model than this ,if so please mention which ?

Because they can extract spatial features, CNNs are some of the best for image classification tasks. But on large or mobile-focused deployments, deeper architectures like ResNet or lightweight models like MobileNet or EfficientNet may work even better.

1. What is the Accuracy after 5 epochs ,10 epochs ?

After 5 epochs, the accuracy is about 83%.

Accuracy after 10 epochs: about 87%

1. Is your model overfitting the data or underfitting the data or an optimal model for making predictions ? Justify

The model is working perfectly. The training and validation accuracies are very similar,

and the validation accuracy doesn't level off too soon or drop, which means that the

model isn't overfitting or underfitting.

1. How can you use it in real life experience ,if you had given the chance to step further? (Use your own imagination)

The model could be added to hospital diagnostic tools to help pathologists quickly

find samples that might be cancerous. The model can be used in mobile apps for remote diagnostics or built into smart microscopes for real-time detection after further development.