

NATIONAL UNIVERSITY OF COMPUTER & EMERGING SCIENCES
DEPARTMENT OF COMPUTER SCIENCE



**Exploring Deep Learning Models for
Cancer Detection and Classification**

Hamza Tatheer	18L-1029
Khizar Hayat Saani	18L-1133
Mohammad Ahmed Irfan	18L-0979
Muhammad Taifoor Saleem	18L-1186

**B.S. Computer Science
2022**

Supervisor: Ms. Noshaba Nasir

Co-Supervisor: Dr. Kashif Zafar

Anti-Plagiarism Declaration

This is to declare that the above publication produced under the:

Title: Exploring Deep Learning Models for Cancer Detection and Classification

is the sole contribution of the author(s) and no part hereof has been reproduced on **as it is** basis (cut and paste) which can be considered as Plagiarism. All referenced parts have been used to argue the idea and have been cited properly. I/We will be responsible and liable for any consequence if violation of this declaration is determined.

Date: 01/06/2022

Student 1

Name: Hamza Tatheer

Signature: 

Student 2

Name: Khizar Hayat Saani

Signature: 

Student 3

Name: Mohammad Ahmed Irfan

Signature: 

Student 4

Name: Muhammad Taifoor Saleem

Signature: 

Table of Contents

Table of Contents

Abstract	1
Chapter 1: Introduction	2
Scope of the Project	2
Goals and Objectives	2
Chapter 2: Literature Survey	3
2.1 Datasets	3
2.1.1 BreaKHis	4
2.1.2 Bioimaging Challenge 2015 dataset	5
2.1.3 BACH	6
2.1.4 Kumar Dataset	7
2.2 Methods and Models	7
2.2.1 Pre-processing	8
2.2.2 Convolution Neural Networks	9
2.2.2 BreastNet	10
2.2.3 PMNet	12
2.2.4 HoVerNet	13
2.3 Test Metrics	14
2.3.1 Receiver Operating Characteristic (ROC) Curve	14
2.3.2 Gleason Score	14
2.3.3 Dice-Similarity Coefficient	15
2.4 Literature Review Summary Table	15
Chapter 3: Requirements and Design	19
3.1 Functional Requirements	19
3.2 Non-Functional Requirements	19
3.3 Assumptions	19
3.4 System Architecture	20
3.4.1 BreastNet	20
3.4.2 PMNet	21
3.4.3 Comparative Analysis	22
3.4.4 Building the Expert System	23
3.5 System Requirements	24
3.5.1 Hardware Requirements	24
3.5.2 Software Requirements	24
3.6 Methodologies	24
3.6.1 BreastNet	24
3.6.2 PMNet	24
3.6.3 HoverNet	25

Chapter 4: Implementation and Test Cases	27
4.1 Implementation	27
4.1.1 Implementation of BreastNet	27
4.1.2 Crossover: BACH dataset on BreastNet model	28
4.1.3 Implementation of PMNet	30
4.1.4 Expert System	30
4.2 Test Case Design and Description	31
4.3 Test Metrics	37
Chapter 5: Experimental Results and Analysis	36
5.1 Results and Discussion	36
Chapter 6: Conclusion	37
References	39

List of Tables

Table 1: Dataset Breakdown	4
Table 2: Summarized Performance Statistics	12
Table 3: Literature Review Summary	16
Table 4: Result summary for BreastNet using BreakHis and BACH	31
Table 5: Result summary for BreastNet vs PMNet using BACH dataset	31
Table 6: Result summary for HoVerNet vs PMNet using CoNSeP dataset	31

List of Figures

List of Figures

Figure 1: Sample Images from Dataset	4
Figure 2: Sample Images from Dataset	5
Figure 3: Cases of microscopy picture patches from the utilized dataset	6
Figure 4: Examples of microscopy images from the BACH dataset	6
Figure 5: Sample of segmented histology images from the Kumar Dataset	7
Figure 6: Pre-processing steps exhibited	8
Figure 7: Actual images with corresponding augmented images	9
Figure 8: The schematic diagram of VGG and AlexNet models	10
Figure 9: Residual Learning: Building Block	11
Figure 10: The general design of BreastNet	11
Figure 11: The PMNet Architecture	13
Figure 12: HoVer-Net Architecture	14
Figure 13: The general design of BreastNet	20
Figure 14: Overview of Comparative Analysis	22
Figure 15: Overview of the localized Expert System	23
Figure 16: EfficientNet B0 Architecture	25
Figure 17: Screenshots from proposed Expert System	31

Abstract

Cancer is a chronic disease that occurs when cells within one part grow uncontrollably and spread to other parts of the body. Often, cancer can go undetected as it is not caused by an invasive pathogen but created by the body's own cells. Medical images produced using common tests in diagnosing cancer are examined by a professional to determine the existence of cancer which requires a lot of manual work and money. Using deep learning frameworks to train the system to accurately detect and classify cancer with these medical images, we can significantly reduce the delay in cancer diagnosis. The complete glory of cancer detection lies in a system that can efficiently detect cancer from medical imagery through segmentation of regions of interest and classification of the stage. Given a medical image, our expert system will accurately detect and classify a particular type of cancer. It may also be trained to detect the presence of cancer in its early stages, playing a major role in global cancer control. Our system will be able to segment and annotate the classes of nuclei and tissue environments within histopathological imagery at runtime to assist medical professionals in their ontological diagnosis and analysis. Our report will serve as the gateway towards research into the field of Cancer Detection and Image Classification.

Chapter 1: Introduction

Cancer is a chronic disease that occurs when cells within one part grow uncontrollably and spread to other parts of the body. Metastasis is the spreading out of malignant cells from a primary tumor to distant regions. It poses the biggest threat towards the cause of death of cancer patients [1]. Often, cancer can go undetected as it is not caused by an invasive pathogen but by the body's own cells. Common tests used to diagnose cancer include MRIs, ultrasounds, X-rays, biopsies, etc. These medical images are examined by a professional to determine the existence of cancer. Using deep learning frameworks to train the system to accurately detect and classify cancer with these medical images, we can significantly reduce the delay in cancer diagnosis. Further, using relevant datasets, the system may also be trained to detect the presence of cancer in its early stages, playing a major role in global cancer control. Given a medical image, the framework should accurately be able to detect and classify a particular type of cancer. It is worth noting that the complete glory of cancer detection lies in a system that can efficiently detect cancer from medical imagery through segmentation of regions of interest and the classification of the stage. We will hence be focusing on models that cater to both these requirements of segmentation and classification.

1.1 Scope of the Project

In the report, we discuss various models used for the detection and classification of breast cancer along with their associated, publicly available datasets. From among these models, we select two high performing models, namely, BreastNet and PMNet as we move towards the implementation phase. The requirements, constraints and implementation details of these models are discussed.

1.2 Goals and Objectives

Using these models, we will aim to form a comparative analysis as we variate the datasets upon which these datasets were originally created. Through this, we hope to locate a model that is more fit for generalization and shows promise for use in our Expert System.

A more elaborated explanation of the goals and objectives are as follows:

- Explore and understand the recently updated models of breast cancer detection.
- Implement model based on the following criteria
 - Room for Improvement
 - Dataset Availability and Size
 - Feasibility based on Implementation Requirements
- Recreate and implement a system that allows us to detect malignant and benign cancers in a given stream of imagery.
- Recreate comparative results using existing models based on strengths, weaknesses, accuracy, performance metrics etc.
- Build a prototype for an Expert System with a stable GUI for nuclear segmentation and classification for use in oncological diagnosis.

Chapter 2: Literature Survey

The literature that we have covered over the course of this project can broadly be categorized into three classifications of cancer covering multiple techniques and models and a variety of datasets. In the course of this report, we will be elaborating these details in the chronology of the said classification and will be providing a detailed analysis of everything covered so far. We will be limiting the literature survey to cover only the models and datasets associated with breast cancer and detection.

2.1 Datasets

The datasets discussed below show the most promise to our goal of cancer detection and classification when it comes to breast cancer. They have been extensively used over the years in association with deep learning models [2]. Over the course of the research, further datasets may also be added for more effective training and generalization.

When employing these datasets within the domain of deep learning, classification can occur in two ways: binary and multi-class. Within binary classification, the input image of the breast tissue is classified as either benign (non-cancerous) or malignant (cancerous). Within multi-class classification, the type of tumor is also recognized whether benign or malignant. Each class is further subdivided into four classes each as shown in Figure 1.

For benign lesions, these are [3]:

1. Adenosis (A)
2. Fibroadenoma (F)
3. Tubular adenoma (TA)
4. Phyllodes tumor (PT)

For malignant lesions, these are [3]:

1. Ductal carcinoma (DC)
2. Lobular carcinoma (LC)
3. Mucinous carcinoma (MC)
4. Papillary carcinoma (PC)

Requirements and Design

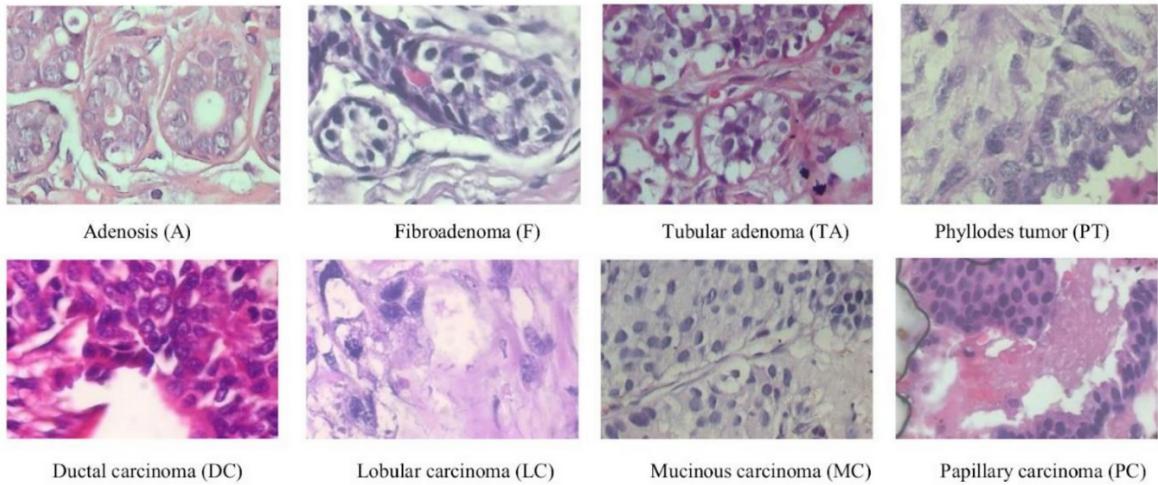


Figure 1: Sample Images from Dataset [3]

The top row shows benign samples while the bottom row refers to malignant samples

2.1.1 BreaKHis

Breast cancer detection and classification often involve the careful analysis of histopathological (histology) images under a microscope by pathologists. The standard benchmark data within this domain is the Breast Cancer Histopathological Database - BreakHis [4].

The BreakHis database consists of 7,909 microscopic images (2,480 benign and 5,429 malignant) of breast tumor tissue collected using different magnifying factors (40x, 100x, 200x, and 400x) as shown in Figure 2. This public database has been built in collaboration with the P&D Laboratory (Pathological Anatomy and Cytopathology) in Brazil. Coming to specifications, all the samples in the dataset have 700x460 pixels, 3-channel RGB, 8-bit depth in each channel and PNG format [4].

The BreakHis 1.0 dataset is broken down in Table 1 as follows:

Table 1: Dataset Breakdown
BreakHis 1.0 dataset is distributed in the below mentioned categories.

Magnification	Benign	Malignant	Total
40x	652	1370	1995
100x	644	1437	2081
200x	623	1390	2013
400x	588	1232	1820
Total of Images	2480	5429	7909

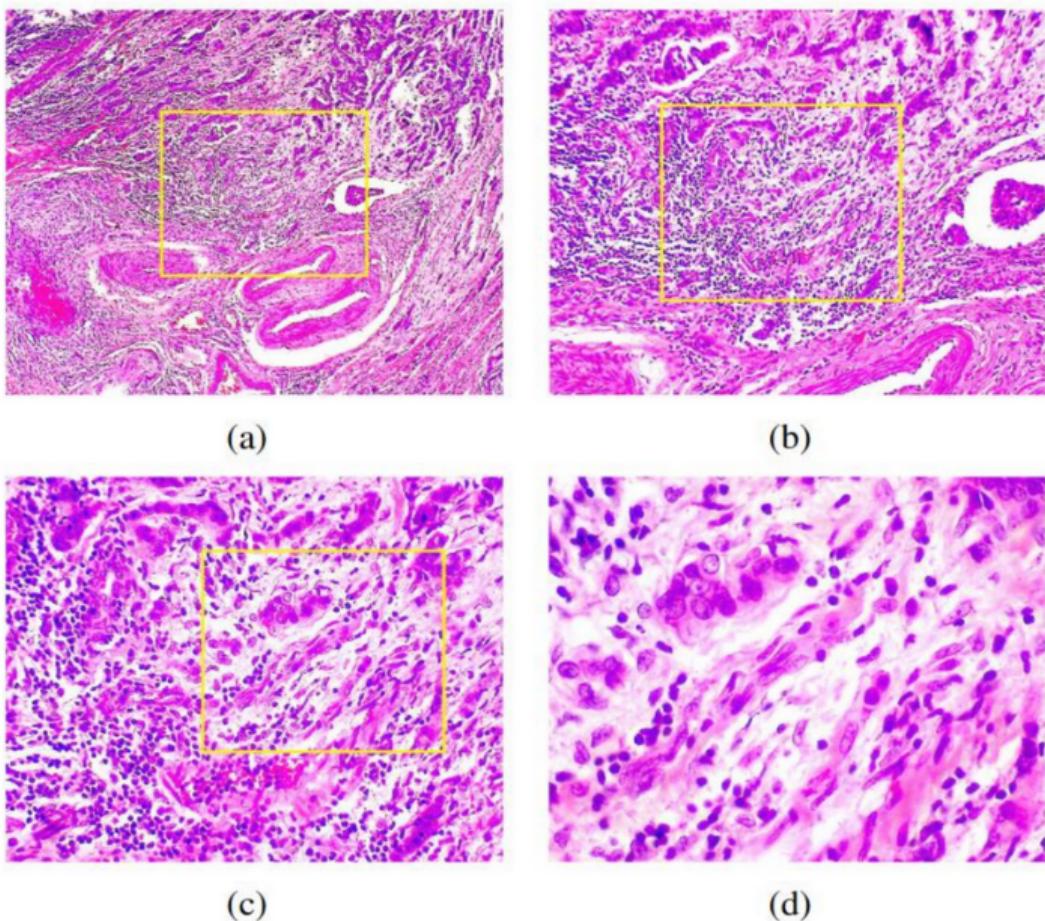


Figure 2: Sample Images from Dataset [4]

*A slide of breast malignant tumor (stained with H&E) seen in different magnification factors:
 (a) x40, (b) x100, (c) x200, and (d) x400 [4].*

2.1.2 Bioimaging Challenge 2015 dataset

To complement the BreakHis database, we have also found other histology images databases relevant to breast cancer detection and classification. The Bioimaging Challenge 2015 dataset consists of annotated H&E-stained images (2040×1536 pixels each) [5].

The dataset contains an extended training set of 249 images and a separate test set of 20 images. All the images are digitized with the same conditions; a magnification of $200\times$ and pixel size of $0.42\mu\text{m} \times 0.42\mu\text{m}$. Each image is labeled with one of four classes as shown in Figure 3:

- I. normal tissue,
- II. benign lesion,
- III. in situ carcinoma
- IV. invasive carcinoma [5]

Requirements and Design

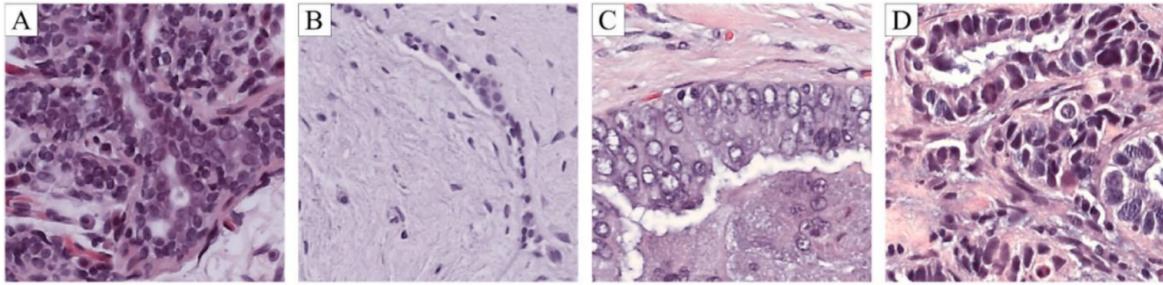


Figure 3: Cases of microscopy picture patches from the utilized dataset [5]

(A) Normal Tissue; (B) Benign Abnormality; (C) Malignant Carcinoma in Situ;
(D) Malignant Invasive Carcinoma

2.1.3 BACH

Another dataset that will be used in our research is the Grand Challenge on BreAst Cancer Histology images (BACH) which was assembled together for the 15th International Conference on Image Analysis and Recognition (ICCIAR 2018) [6]. The dataset is composed of two parts, the microscopy images dataset and the whole-slide images dataset. Each image within the complete dataset can be classified into one of four groups, as shown in Figure 4, which are:

- 1) Normal
- 2) Benign
- 3) In situ carcinoma
- 4) Invasive carcinoma.

The microscopy images consist of 400 training and 100 test images in RGB, TIFF format, a pixel scale of $0.42 \mu\text{m} \times 0.42 \mu\text{m}$ and a size of 2048×1536 pixels. The WSI dataset is composed of 30 images to train and 10 images for validation testing. For the WSIs, it provides 10 pixel-wise annotated regions for identifying regions of interest.

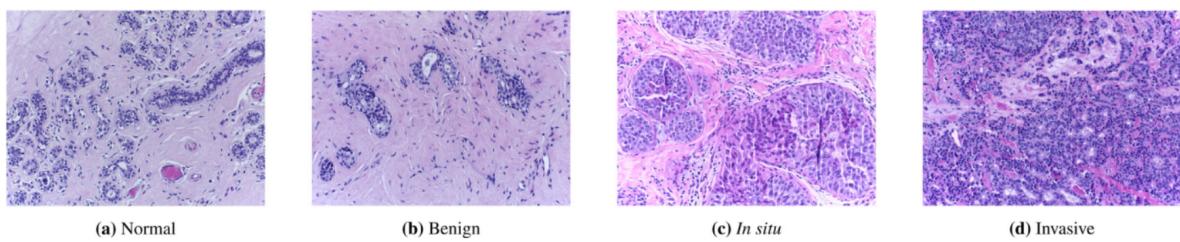


Figure 4: Examples of microscopy images from the BACH dataset [6].

(A) Normal Tissue; (B) Benign Abnormality; (C) Malignant Carcinoma in Situ;
(D) Malignant Invasive Carcinoma

2.1.4 Kumar Dataset

The Kumar dataset contains 30 1000 x 1000 picture tiles taken at 40 magnification from The Cancer Genome Atlas (TCGA) database for seven organs (6 breast, 6 liver, 6 kidney, 6 prostate, 2 bladder, 2 colon, and 2 stomach), a sample of which are shown in Figure 5. Each nucleus' boundary is completely marked within each image.

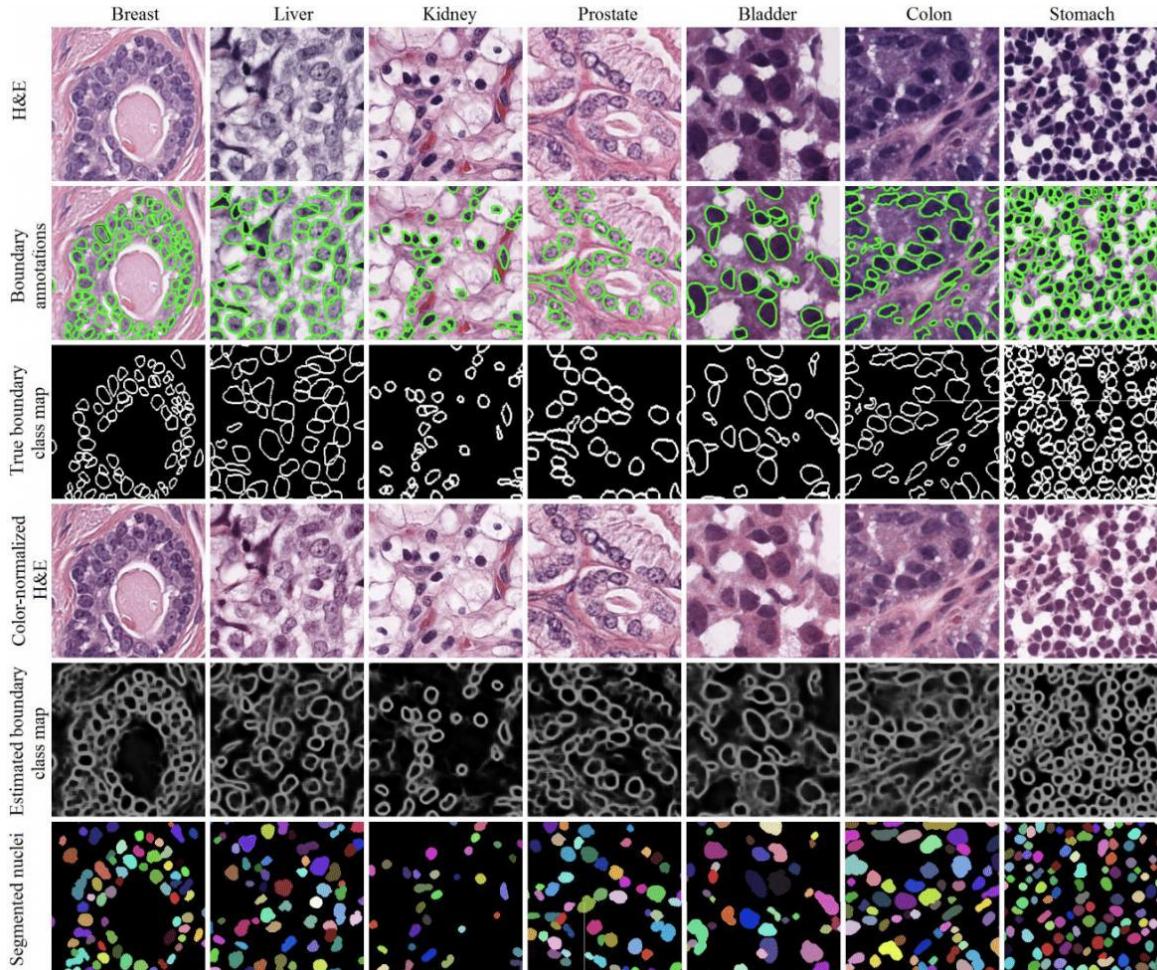


Figure 5: Sample of segmented histology images from the Kumar Dataset [10]

Columns left to right: breast, liver, kidney, prostate, bladder, colon, stomach.

Rows top to bottom: H&E, Boundary annotations, True boundary class map, color-normalized H&E, Estimated boundary class map, Segmented nuclei.

2.2 Methods and Models

In this section, we will be discussing the model and architectures that we have implemented and used to generate a comparative analysis. These are the BreastNet and PMNet models. These models were selected due to their relative high accuracy and the promise they show should the models achieve generalization. The models work on the principles of detection and classification (BreastNet) as well as segmentation (PMNet, HoVerNet). Together, these

Requirements and Design

models could form the foundation for breast cancer detection, segmentation and classification using microscopy images and WSIs.

2.2.1 Pre-processing

Before we enter the domain of the neural network, it is essential to effectively pre-process the input images to remove noise, resize them, or perform augmentation to increase the dataset size. Most histology images are H&E stained. Hematoxylin selectively stains nucleic acids in a blue-purple hue and eosin stains proteins with a bright pink color [7]. The most common pre-processing techniques include color normalization, data augmentation, and segmentation, as presented in Figure 6.

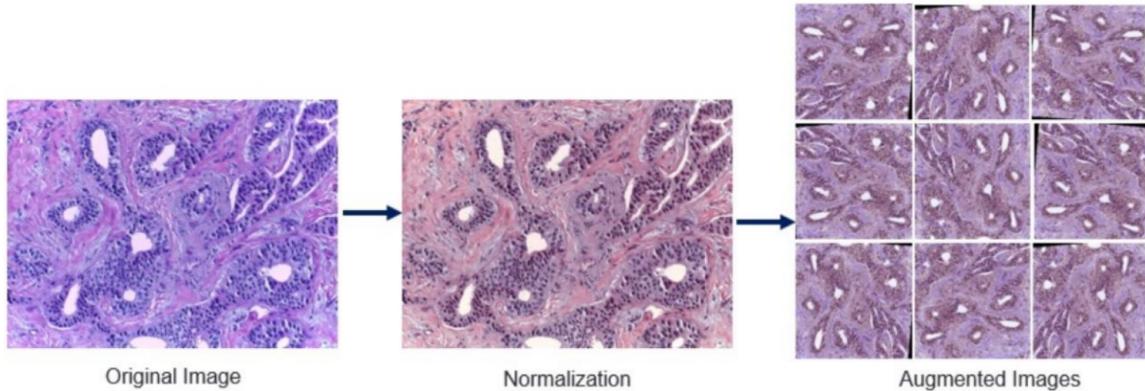


Figure 6: Pre-processing steps exhibited [8]

Normalization and image augmentation are performed on the input image, respectively.

Models proposed by Ud din et al [9] and Vo et al [8] use pre-processing techniques to correct stain vector and intensity variation using the technique proposed in [7]. By this method, the H&E-stained images are normalized leading to much improved visual consistency which makes the learning process much more efficient.

One of the most important steps in the pre-processing phase involves data augmentation. Here, the dataset is increased by using geometric transformations to the image datasets via simple image processing techniques [9]. In the model proposed by Alom et al. [3], 21 samples were generated from each input image using translations, rotations, shears and flipping. The Figure 7 demonstrates samples from the Bioimaging Challenge dataset after data augmentation techniques have been applied.

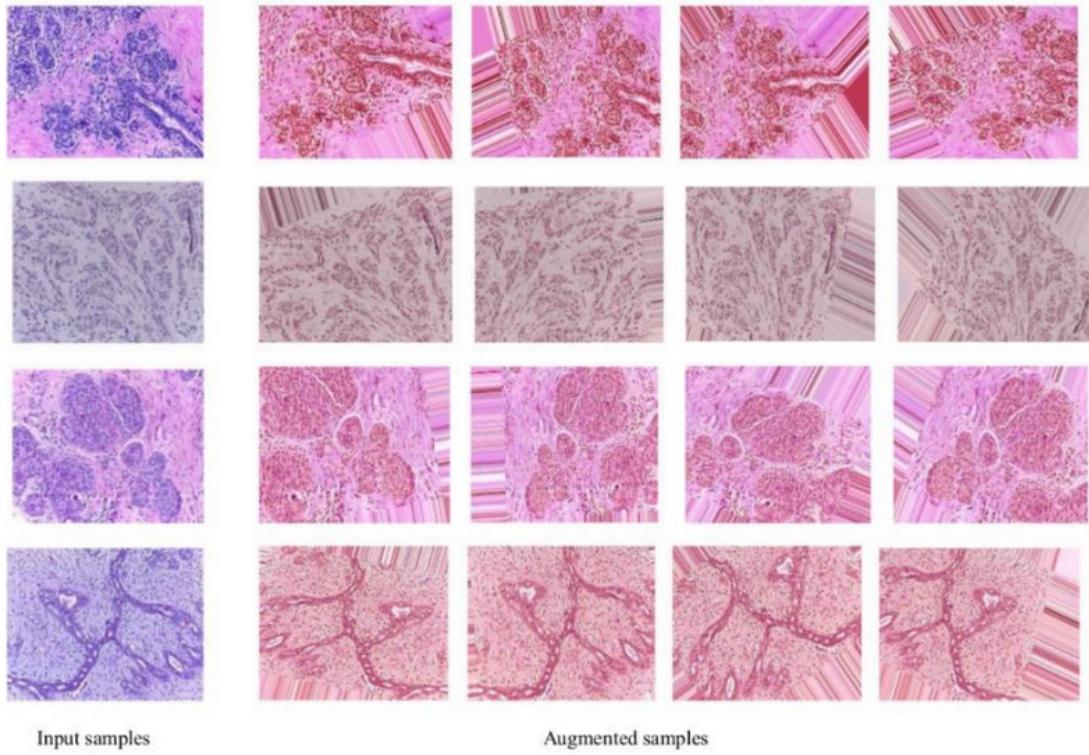


Figure 7: Actual images with corresponding augmented images.

The actual images are shown on the left. The 20 augmented images of these four samples are shown on the right [6]

After color normalization and data augmentation, segmentation has also been observed to be an effective approach towards pre-processing histology images. The segmentation process locates the edges and boundaries of regions in a histopathology image to locate and then extract the cells in the images. This is key towards identifying the regions of interest (ROI) and highlighting them in the images [2]. Kumar et al. [10], used k-means clustering as means for segmentation claiming that this method performs better in comparison with other commonly used segmentation methods.

While these pre-processing techniques are important, the entire step has also been observed to be skipped completely at times. In BreastNet, classification is carried out without using any preprocessing [11].

2.2.2 Convolution Neural Networks

Convolution neural networks (CNNs) are a widely popular architecture for image classification tasks. A typical CNN uses filters to extract features from within the pixels of an image [14] and consist of three bases [11]:

1. Convolution layers: These layers include convolution filters that are applied to the image. Several operations are performed to obtain values where the filter, used as a

Requirements and Design

sliding window, passes. Mostly, these layers are followed by the ReLU activation to prevent the vanishing gradient problem.

2. Pooling layers: These layers summarize the information brought in by the previous layers by reducing the size of the previous layer and keeping only important information.
3. Dense layers: These are fully connected layers that perform classification. This is the final layer that contains a single neuron and uses an activation function, such as the Softmax function, to define a threshold for classification [12].

These techniques equipped with varying filter sizes, layers, and activation functions form the basis of state-of-the-art CNNs such as VGGNet or AlexNet. The basic structures of these models are shown in Figure 8 [11].

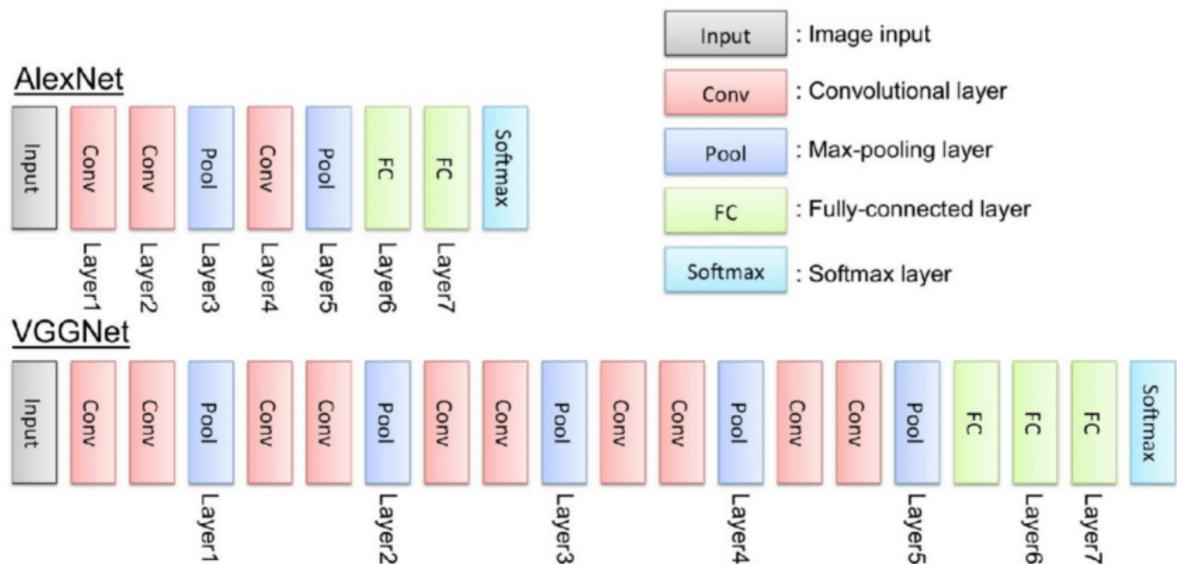


Figure 8: The schematic diagram of VGG and AlexNet models [13]

Top Left: Basic structure of AlexNet. Top Right: Standard CNN layers.

Bottom: Basic structure of VGGNet

2.2.2 BreastNet

The BreastNet architecture is an end-to-end model that is trained and tested on the BreaKHis dataset [4]. This architecture consists of the following modules, in general [11]:

- Convolutional Block Attention Module (CBAM)
- Dense block
- Residual block
- Hypercolumn technique

The CBAM is an attention module that identifies the key places within the WSIs. It highlights the important areas of an image and allows the model to aim there to get the most out of crucial segments within the input image. The use of CBAM in this model has contributed significantly towards accuracy and optimization [11].

The residual block aims to smoothen the gradient and hence decrease the overall loss of the input as it passes through the various layers of the architecture. In a network with residual blocks, skip connections or “shortcut” connections are used to tackle the problem of

Exploring Deep Learning Models for Cancer Detection and Classification

vanishing gradients as our network becomes deeper [14]. The use of the residual blocks, as shown in Figure 9, is also the basis of the state-of-the-art ResNet architecture.

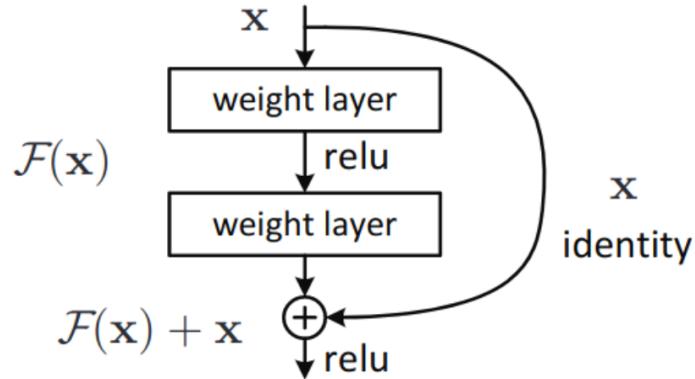


Figure 9: Residual Learning: Building Block [14]

The Rectified Linear, or ReLu, serves as the activation function for ResNet.

The Hypercolumn examines the images on various scales and provides concrete results in the classification phase and improves performance. The entire flow of the BreastNet architecture is shown in Figure 10.

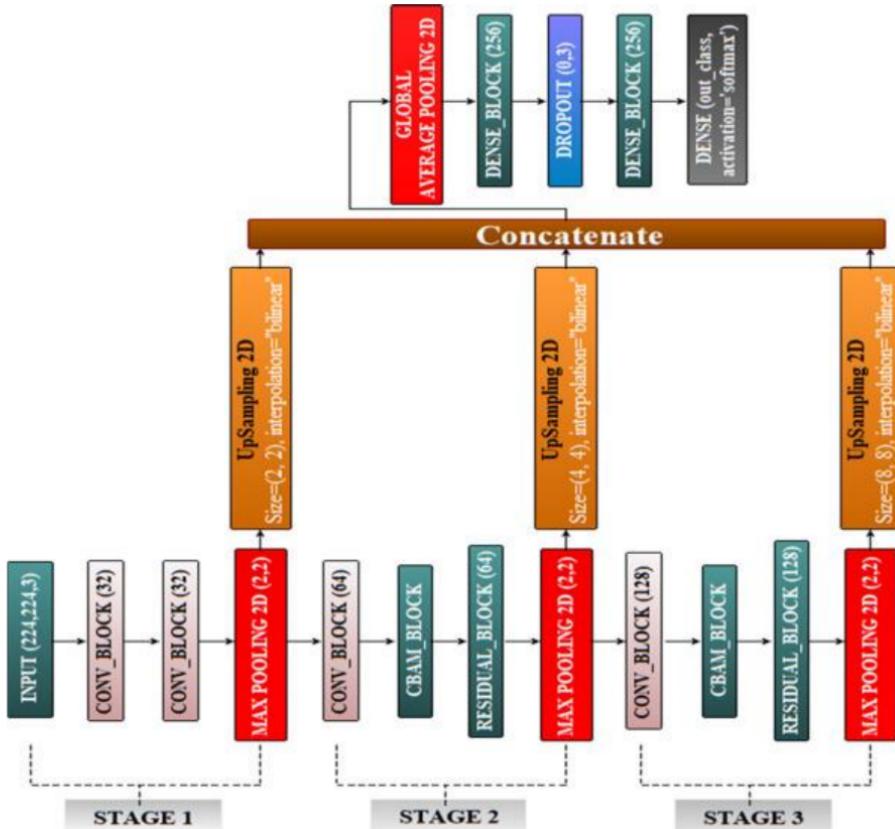


Figure 10: The general design of BreastNet as shown in [11]

The images H&E images are processed at three different zoom rates and scales. The information gathered individually in each stage is combined/concatenated to obtain global average results.

Requirements and Design

For the purpose of this study [11], the BreaKHis dataset was split by 80-20 by a train/test split. The classification process was carried out without any pre-processing with hypercolumn examining the images at three different stages rather than creating a separate model for each zoom rate leading to a further optimized architecture.

The best classification result was 98.51%. This result was increased to 98.80% by performing repeated binary classification by combining data from all the magnification factors [11]. Table 2 shows the overall performance of the BreastNet architecture.

Table 2: Summarized Performance Statistics

The results of the original research carried out using BreastNet over BreakHis dataset are presented here.

Accuracy	Sensitivity	Precision	F1-Score
98.80	98.35	98.84	98.59

2.2.3 PMNet

The PMNet architecture introduces a probability map based pipeline to detect regions of cancer in WSIs. The PMNet is trained using the BACH dataset which consists of WSIs of 10 patients and the validation of the approach made is done using 173 annotated WSIs of the Dryad dataset. Figure 11 demonstrates the architectural layout for PMNet.

The primary stage of this state-of-the-art model is patch level training on WSIs. Class imbalance and overfitting are tackled initially. EfficientNet (B0) is used for this due to its ability to perform a neural architecture search whilst optimizing the number of parameters and the overall accuracy. In this stage, the probabilities for each patch of WSI is predicted by the model. A probability map is formed containing the combined probabilities for each WSI patch, respectively in the same order as that of the WSIs.

In the second stage, two inputs are fed to the PMNet architecture: resized WSIs and their probability maps. These probability maps contain the probability that each patch in stage 1 can contribute towards the detection of the cancer. The two inputs are fed through individual series of convolution and max pool operations with the results later to be concatenated together using the Rectified Adam (RAdam) optimizer. A series of convolutions and upsampling operations are applied on the combined layer to learn from both features. Finally, the sigmoid activation function is used for both outputs.

Exploring Deep Learning Models for Cancer Detection and Classification

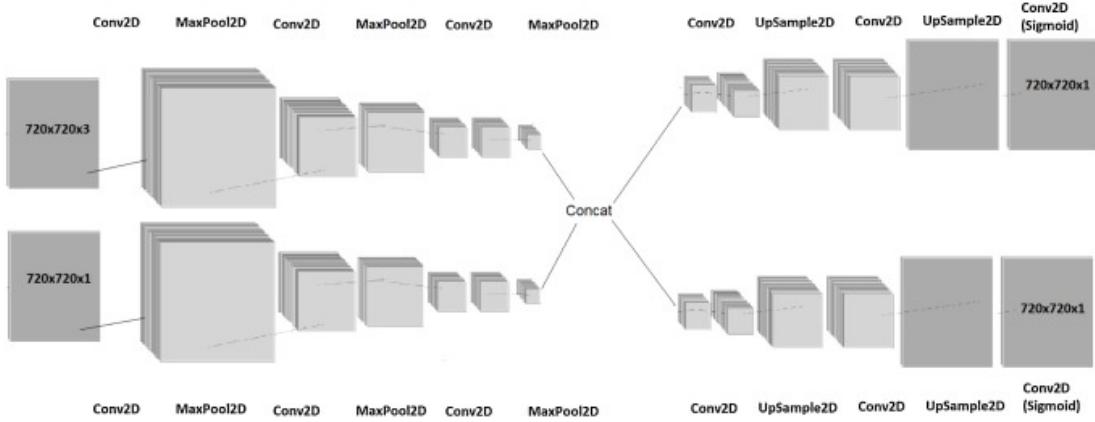


Figure 11: The PMNet Architecture as shown in [15]

The model is fed two inputs. The first half generates probability masks (both on a local and global scale) followed by Conv2D and max-pooling operations. The second half combines these results and calculates the dice coefficients for cancerous and non-cancerous regions followed by Conv2D and upsampling operation.

Salman et al [15] trained the model on the BACH dataset and used the Dryad dataset for validation. The approach yielded an f1-score of $88.9(\pm 1.7)\%$ that outflanks the benchmark f1-score of $81.2 (\pm 1.3)\%$, accomplished a normal dice coefficient of 69.8% on 10 whole slide pictures compared to the benchmark normal dice coefficient of 61.5% on BACH dataset, and on the Dryad test dataset that comprises 173 WSIs, a normal dice coefficient of 82.7% as compared 76% of the past model without fine-tuning on this dataset.

2.2.4 HoVerNet

In our strive to prototype an industry standard Expert System, we turned towards deep learning models that could effectively aid in segmentation and annotation of Whole Slide Images. This will free doctors and pathologists of the tedious task of manually annotating these slides while focusing on quick diagnosis that could potentially save the lives of millions. Furthermore, such a system will be quick to earn the trust of medical professionals rather than a “black box” system that merely classifies input images into different cancerous/non-cancerous classes.

Annotation and segmentation of nucleic tissue environments is an essential step in the diagnosis of Cancer from histology images. During our research for segmentation models we came across the HoVerNet model.

The HoverNet model is a very recent contribution to the deep learning paradigm for computational pathology. The model is based on public pathology datasets and provides nucleus segmentation and classification capabilities for diagnostic investigation.

The model uses pre-configured residual blocks, which are a combination of CNN layers and pooling algorithms, to perform typical computer vision tasks like nucleus segmentation. It also concentrates on the encoder-decoder design, which has become well-known in modern semantic segmentation and classification problems.

The pipeline's initial effort focuses on mutual downsampling of the input patch. This is

Requirements and Design

accomplished with the help of residual blocks, which use the CNN workflow to create feature maps of the patch's regions of interest. This is done in order to prepare the input for the segmentation and classification pipelines, which use weakly supervised learning to train the model for tasks like these.

To perform the segmentation and classification tasks, the model architecture comprises of the following three layers, as shown in Figure 12:

- Nuclear pixel (NP) branch
- HoVer branch and
- Nuclear classification (NC) branch.

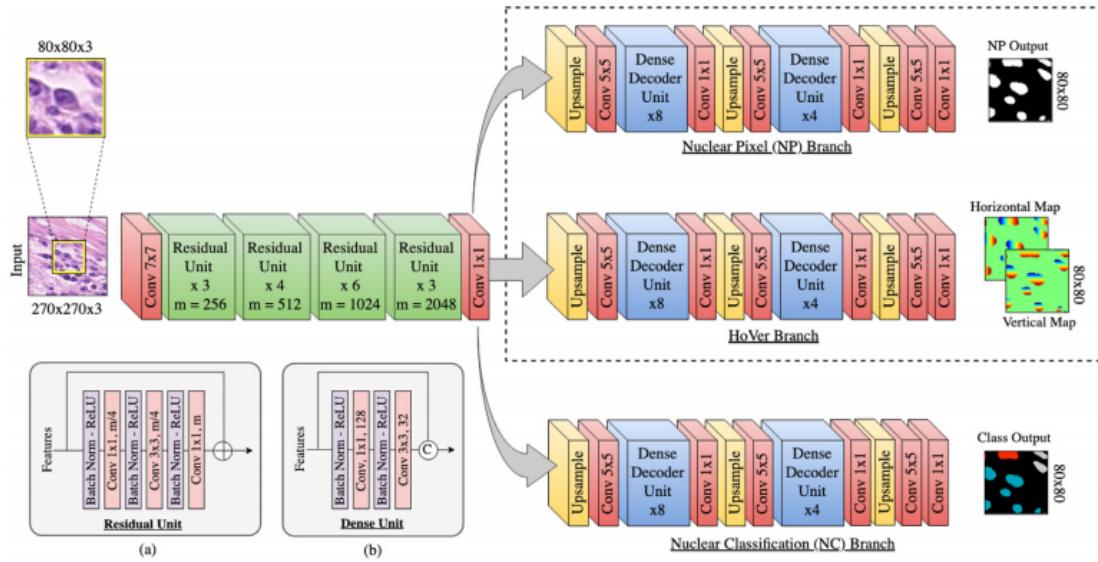


Figure 12: HoVer-Net Architecture [20]

Overview of the HoVer-Net Architecture. (a) Residual Unit, (b) dense unit.

2.3 Test Metrics

2.3.1 Receiver Operating Characteristic (ROC) Curve

The ROC Curve provides a visual representation of the overall performance of a classification model to overcome discrimination between data belonging to two different classes. The performance of the model is measured by constructing all the possible decision boundaries for classification. A graph of true positives rate vs false positives rate is plotted for all those decision boundaries. However, an ROC curve is found to be most profitable when there are multiple models involved for classification and their accuracy for detecting true positives needs to be measured. The Area Under ROC (AUROC) provides this detail. Higher the area indicates better ability of the model to detect true positives and overcome discrimination between the data points.

$$\text{True positives rate} = \text{Sensitivity} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

$$\text{False positives rate} = (1 - \text{Specificity}) = \frac{\text{False Positives}}{\text{False Positives} + \text{True Negatives}}$$

2.3.2 Gleason Score

According to [16], the threat that a cancer poses to the patient is seen in two ways: the stage of the cancer and the grade of the cancer. The former looks at where the cancer is present in the body while the latter determines what the actual cancer cell looks like under the microscope. All cancer patients need to be tested in terms of the cancer stage that has developed. Skin cancers require an assessment of the cancer stage as well the cancer grade. Gleason grades are awarded to the two most predominant samples in the biopsy. The Gleason Score is the sum of both grades which tells how critical the cancer is. GS \leq 6 suggests cancer cells that are similar to normal cells whereas GS \geq 8 signify a high risk of advancing cancer. According to R. Cao [17], the GS prediction is usually evaluated by receiver operating characteristic (ROC) analysis. The scores are tested by forming groups of the critical scores e.g., GS \geq 7 vs GS $<$ 7 and GS \geq 4+3 vs GS \leq 3+4 and ROC values are tested for each group in joint models.

C. de Vente et. al. [17] continued to make use of prepositioned data such as prostate zonal segmentation and supervised learning techniques such as ensembling. They published their result as being significantly accurate with a voxel-wise weighted kappa of 0.446 ± 0.082 and a Dice-Coefficient for segmenting csPCa of 0.370 ± 0.046 . Particularly on the Prostate-X Challenge Dataset [18], the lesion-wise weighted kappa was 0.13 ± 0.27 . They claimed to outperform “*standard multiclass classification and multi-label ordinal regression*”. They concluded that soft-label ordinal regression improved over the detection accuracy for bp-MRIs more than any other model tested and provided a comparative of certain models as well that allowed us to analyze at the surface level what it meant to use and deploy these said models.

2.3.3 Dice-Similarity Coefficient

Dice-Similarity Coefficient:

$$\frac{2TP}{2TP + FP + FN} = \frac{2|X \cap Y|}{|X| + |Y|}$$

From [19], values are between 0 and 1. As the DC comes closer to 1, the model becomes increasingly good at predicting results.

- Simply put, the Dice-Coefficient, also known as the Dice Loss, equals twice the number of elements common to both sets divided by the sum of the number of elements in each set.
- It is a widely used metric in computer vision as a means to calculate the similarity between two images.
- Segmentation results can be evaluated quantitatively by the dice coefficient.
- It has inspired the use of a variation of loss functions for the HoVerNet architecture to measure the detection quality, segmentation quality, and the overall panoptic quality [20].

Requirements and Design

$$\text{Detection Quality} = \frac{|TP|}{|TP| + 0.5|FP| + 0.5|FN|}$$

$$\text{Segmentation Quality} = \frac{\sum IoU(x,y)}{|TP|}$$

$$\text{Panoptic Quality} = \text{Detection Quality} \times \text{Segmentation Quality}$$

2.4 Literature Review Summary Table

Table 3 mentions the amalgamation of the research papers and articles that would be the most significant to our research and were used as “base papers”.

Table 3: Literature Review Summary

The summary of various research techniques and datasets discovered over the course of our study are presented here.

No.	Name, reference	Description	Year	Model/Technique	Dataset	Performance
1.	BreastNet: A Novel Convolutional Neural Network Model through Histopathological Images for the Diagnosis of Breast Cancer [11]	Use of CNN, attention modules dense and residual blocks	2020	BreastNet	BreaKHis	Accuracy: 98.80% Sensitivity: 98.35% Precision: 98.84% F1-Score: 98.59%
2.	Breast Cancer Classification from Histopathological Images with Inception Recurrent Residual Convolutional Neural Network [3].	Use of several convolution layers, IRRUs, transition blocks, and a softmax at the output layer.	2019	IRRCNN	BreaKHis and Bioimaging 2015 breast histology classification challenge	99.05% and 98.59% testing accuracy for binary and multi class breast cancer recognition
3	Ud Din, Ikram & Islam,	Use of pre trained	2019	Transfer Learning	BreaKHis	Overall accuracy

Exploring Deep Learning Models for Cancer Detection and Classification

	Naveed & Rodrigues, Joel. (2019). A Novel Deep Learning based Framework for the Detection and Classification of Breast Cancer Using Transfer Learning. Pattern Recognition Letters [9]	models and Transfer Learning for detection and classification		GoogLeNet VGGNet ResNet	private dataset	of 97.52%
4	Vo, D.M.; Nguyen, N.Q.; Lee, S.W. Classification of Breast Cancer Histology Images Using Incremental Boosting Convolution Networks [8]	Use of multiple Inception Resnet-v2 and gradient boosting trees	2019	Ensembling Inception-ResNet v2	Bioimaging 2015 breast histology classification challenge and the BreaKHis	99.5% for binary classification and 96.4% for four classes
5	Salman Ahmed, Maria Tariq, Hammad Naveed, PMNet: A probability map based scaled network for breast cancer diagnosis, Computerized Medical Imaging and Graphics,	Using a probability map to classify WSI and segment regions that cause prognosis.	2021	PMNet	BACH and Dryad	An F1-score of 88.9(± 1.7)% and an average dice coefficient of 69.8% on 10 whole slide images

Requirements and Design

	Volume 89, 2021 [15].					on BACH dataset. On the Dryad test dataset, an average dice coefficient of 82.7% was achieved.
6	Liew, X.Y.; Hameed, N.; Clos, J. A Review of Computer-Aided Expert Systems for Breast Cancer Diagnosis. Cancers 2021, 13, 2764 [2]	Various	2021	Various	Various	Various
8	HoVer-Net: Simultaneous Segmentation and Classification of Nuclei in Multi-Tissue Histology Images [20]	CNN working on feature extraction and nuclear segmentation/classification	2019	HoVerNet	CoNSeP	Dice: 0.801 F1-Score : 0.516

Chapter 3: Requirements and Design

In this chapter, we will be discussing the requirements and design of our implementations for the BreastNet and PMNet models.

3.1 Functional Requirements

Since this project is mainly research-based, it consists of an extensive back-end system, which has very little interaction with the user. So it has only one functional requirement, which is essentially its main purpose. The user must be able to:

1. receive accurate detection, classification and/or segmentation of cancer from the histopathological images that were input into the system.

3.2 Non-Functional Requirements

Our system has the following non-functional requirements:

1. Adaptability: the system shall accept and work on histology images from different sources, having different zoom rates
2. Extensibility: the system shall be able to integrate further sub-modules developed in the future, which perhaps detect further information from the input images, in order to generate even more accurate detection and classification
3. Performance: the system shall generate accurate results, using less time and resources.
4. Reliability: the system shall always work in its expected manner
5. Robustness: the system shall be able to cope with errors during execution
6. Response time: the system shall generate results in as little time as possible
7. Reusability: the system shall be able to use existing, pre-trained models in order to decrease its training time (Transfer Learning)

3.3 Assumptions

For the scope of this FYP, we have made the following assumptions and designed our system accordingly:

1. The preparation of the databases for our secondary data is done under the supervision of certified pathologists and through a series of well-known steps safely recognized by pathologists around the world.
2. Any analysis, annotation and augmentation of histopathological (histology) images is done under a microscope by certified pathologists and the resulting images are perfect and without any external noise.
3. The databases for our secondary data sets have been built in collaboration with certified Prognostics and Diagnostics (P&D) Laboratories.
4. The annotation and augmentation of any data done is informed in the documentation of their corresponding literature reviews.
5. The research will improve the performance of breast cancer detection and classification in future.

3.4 System Architecture

3.4.1 BreastNet

The BreastNet architecture is an end-to-end model that is trained and tested on the BreaKHis dataset [4]. The BreastNet architecture, as shown in Figure 13, consists of the following modules [11]:

- Convolutional Block Attention Module (CBAM)
- Dense block
- Residual block
- Hypercolumn technique

The CBAM helps in the identification of important places within histopathological images. It selects important areas of the image and allows the model to focus there to get the most out of crucial segments within the input image. The use of CBAM in this model has contributed significantly towards accuracy and optimization.

The residual block aims to smoothen the gradient and hence decrease the overall loss of the input as it passes through the various layers of the architecture. In a network with residual blocks, skip connections or “shortcut” connections are used to tackle the problem of vanishing gradients as our network becomes deeper [14]. The use of the residual blocks is also the basis of the state-of-the-art ResNet architecture.

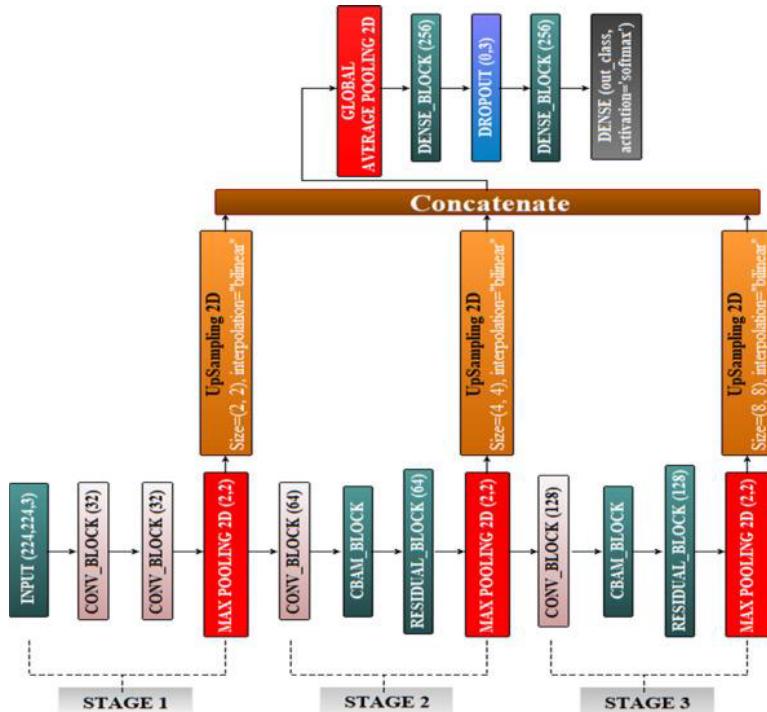


Figure 13: The general design of BreastNet as shown in [11]

The images H&E images are processed at three different zoom rates and scales. The information gathered individually in each stage is combined/concatenated to obtain global average results.

3.4.2 PMNet

The PMNet architecture introduces a probability map based pipeline to detect regions of cancer in WSIs. The PMNet is trained using the BACH dataset which consists of WSIs of 10 patients and the validation of the approach made is done using 173 annotated WSIs of the Dryad dataset.

The primary stage of this state-of-the-art model is patch level training on WSIs. Class imbalance and overfitting are tackled initially. EfficientNet (B0) is used for this due to its ability to perform a neural architecture search whilst optimizing the accuracy and the number of parameters.

In the second stage, two inputs are fed to the PMNet architecture: resized WSIs and their probability maps. These probability maps contain the probability that each patch in stage 1 can contribute towards the detection of the cancer. The two inputs are fed through individual series of convolution and max pool operations with the results later to be concatenated together using the Rectified Adam (RAdam) optimizer. A series of convolutions and upsampling operations are applied on the combined layer to learn from both features. Finally, the sigmoid activation function is used for both outputs.

3.4.3 Comparative Analysis

Figure 14 presents an overview of the comparative analysis performed in our study.

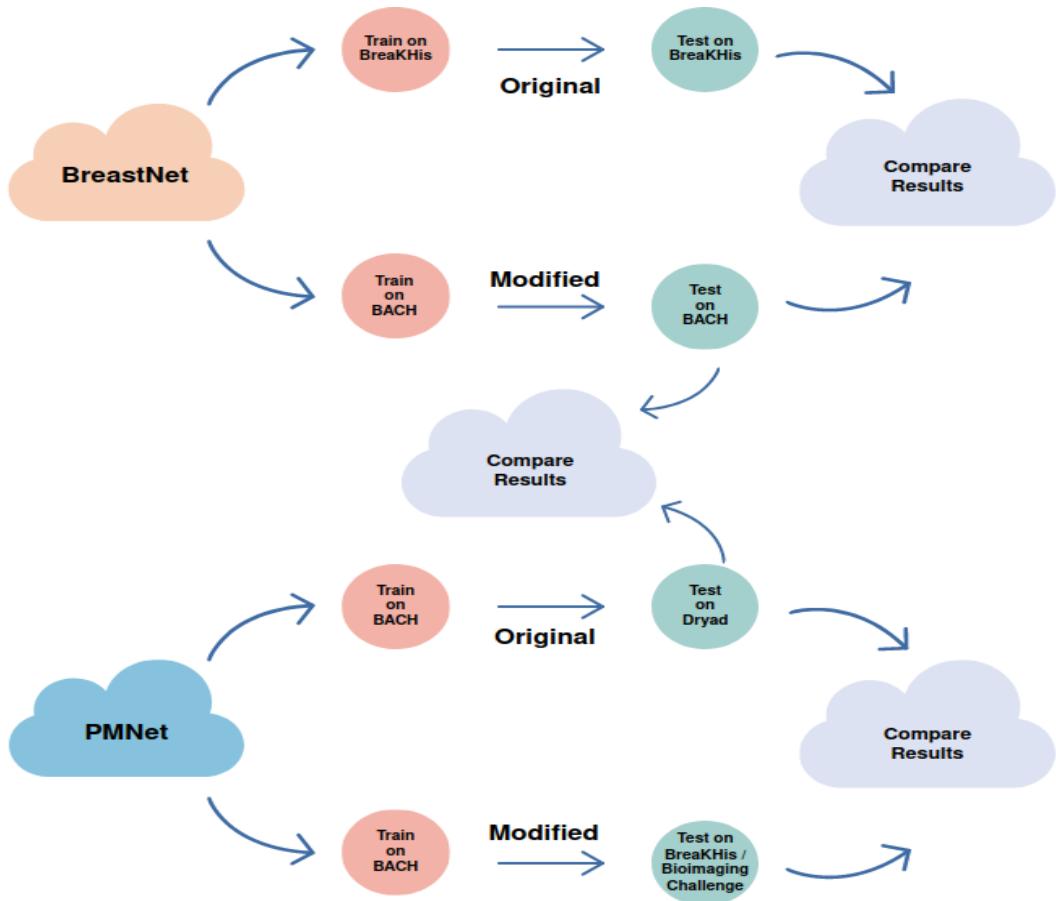


Figure 14: Overview of Comparative Analysis

The comparative analysis consists of testing BreastNet using BreakHis dataset and PMNet using BACH dataset and the crossovers of Breastnet using BACH dataset and PMNet using BreakHis dataset.

3.4.4 Building the Expert System

The expert system primarily consists of the GPU and the integrated RabbitMQ which runs over the operating system of the GPU. The purpose of RabbitMQ is to put longer tasks in a queue based task management so they may be executed when the GPU makes itself available. In the meantime, the user may continue with other routine tasks rather than having to put other operations on hold until the GPU can begin these task(s). This benefits the system in avoiding any load on the API. Normally, we would not want the GPU to execute, for example, 10 to 20 incoming requests to be executed all at the same time. We want these requests to make use of the GPU inference smartly and one at a time. Cellery has been used along with RabbitMQ. Cellery is a library which makes it easy to use RabbitMQ for asynchronous tasks. The GPU may be using Cellery and RabbitMQ, however, the APIs have already been defined for it in a “flask”. This flask gets a call, receives an image, ensures that the task has been added to the task management queue, and returns an estimated time it will require to provide a result. The GPU contacts the back-end to confirm whether the result has been produced and is returned a unique ID for the task.

The back-end is made up of 2 microservices. Firstly, the SVS Viewer does caching on the occasion that two different users provide the exact same whole-slide images and request for inference. This frees the GPU from performing any inference for both images. Moreover, the GPU, as a service, does not need to have knowledge about the patches being stored as it only needs to provide the results. The SVS Viewer would then define some specific APIs which allow for movement over that image using a cursor at the user’s end. The second microservice is the Ambassador. This gives the expert system an architectural benefit that a third-party involved does not have to worry about the patient’s information. They would only require the SVS Viewer service in order to proceed with inference of any images.

Figure 15 demonstrates an overview of the lifecycle of data and how it transitions through our expert system.

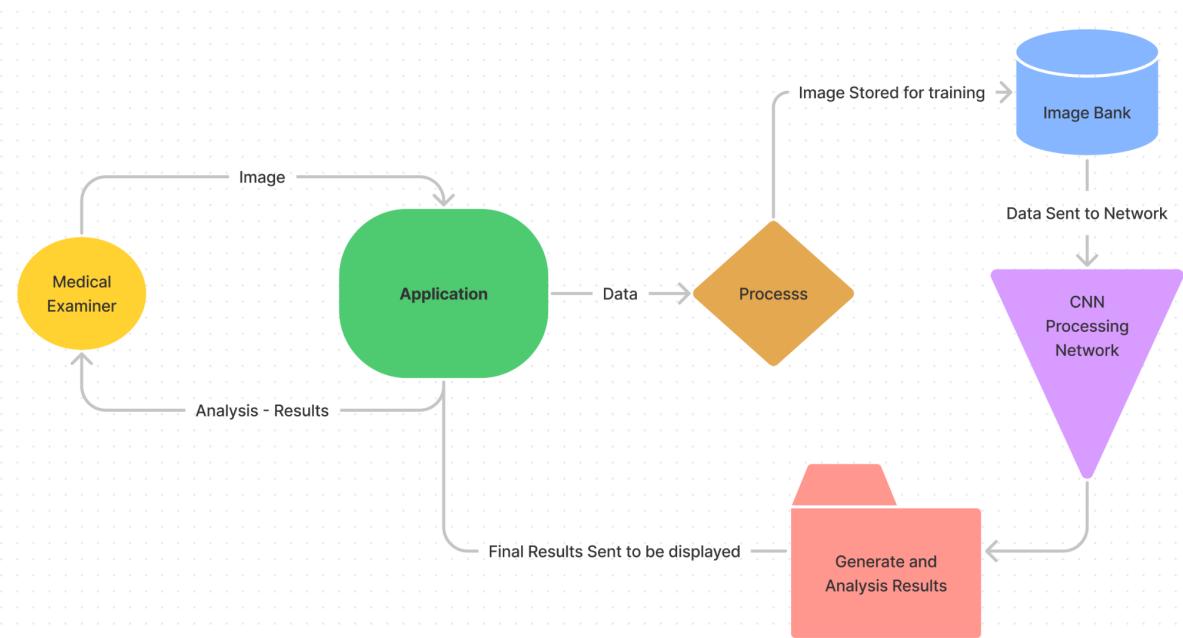


Figure 15: Overview of the localized Expert System

Requirements and Design

The expert system makes use of processed/trained data from the image bank and uses CNN in real-time to analyze and present the user with results through the front-end of the application.

3.5 System Requirements

Our system would require the following hardware and software modules in order to successfully do its job.

3.5.1 Hardware Requirements

1. CPU: 2x 4210 Silver 10 core / 20 Thread Processor
2. RAM: 128GB RAM
3. GPU: 1x 24GB Tesla P40 GPU Card
4. SSD: 1x 480GB SSD DELL Enterprises
5. Internal: 2x 1G, 10G Network Adapter

3.5.2 Software Requirements

1. Python
2. Open slide
3. OpenCV
4. Tensorflow
5. Albumentations
6. Matplotlib
7. Numpy
8. Pillow
9. Scikit
10. tqdm
11. Panda

3.6 Methodologies

3.6.1 BreastNet

The first module in the system is configuring the Convolutional and Dense blocks. It will be trained using Convolutional Neural Networks (CNN) to find identifiable features in the histology images e.g. cancer cells. The features extracted from here will then be used in the next stages of the architecture.

The next two stages of the architecture involve setting up the CBAM. This module contributes to the classification process by extracting more efficient features.

As we move towards completion, the final module to be implemented involves the concatenation of outputs from each stage and feeding them to the hypercolumn. The CBAM will contain residual blocks to tackle the problem of vanishing gradients along with Max pooling as we switch from one stage to the other. Finally, the output of each layer is upsampled, concatenated and fed to the hypercolumn.

3.6.2 PMNet

The PMNet architecture has two functioning stages. The first stage overcomes overfitting and increases robustness by flipping the images horizontally and vertically and class imbalance is overcome using the cross-entropy loss function. EfficientNet (B0) architecture, as shown in Figure 16, is then used to do patch level training on the BACH dataset in order to generate probability maps for the WSIs. B0 makes use of the swish activation function which is a variation of the sigmoid activation function.

Since EfficientNet collects information at patch level, it is unable to capture information at a global scale. This is where PMNet acts as an enhancer for the outputs produced in stage 1. The RAdam optimizer is used to converge the results of stage 1 to feed into PMNet. It basically allows for dynamic adjustments in the learning rate by studying the variance and momentum after each epoch.

PMNet applies 3 pairs of max pool with convolutions on the resized WSI images and probability maps before convergence and upsampling with convolutions after convergence. Two output branches are produced after convergence for the purpose of back propagation to accommodate the dice loss obtained. The probability masks generated after convergence are used to determine the output probabilities for cancer detection from the patches and the global information.

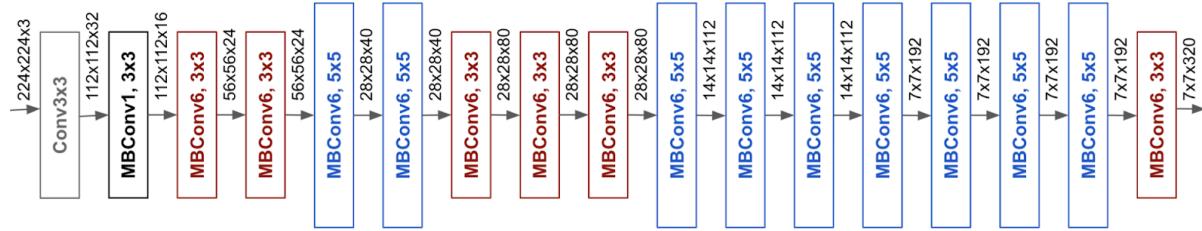


Figure 16: EfficientNet B0 Architecture as shown in [15]

Efficient B0 helps PMNet optimize both the number of parameters and its accuracy.

3.6.3 HoverNet

The feature extraction capability for the model is provided by the down-sampling branch, which is mutual. The model includes three unique branches to carry out the segmentation and classification tasks once the feature maps have been gathered.

The up-sampling module is used by the nuclear pixel branch to convert down-sampled features into greater dimensions. This is particularly critical because the model must operate on border detection at a higher level. The nuclear pixel branch's major goal is to help with the initial segmentation task of distinguishing background pixels from significant nucleus pixels. This is accomplished by using segmentation masks that correlate to the ground facts provided to the model. It would aid in providing the relevant nuclei boundary from the backdrop and carving out the Ho-Ver Branch's zone of interest.

The Ho-Ver branch is concerned with distinguishing the boundaries of overlapped nucleus pixels. Because histopathological slides contain several instances of the nuclei that may be

Requirements and Design

stacked on one another, a methodology that separates the distinct nuclei instances from one another is essential.

The HoVer branch demonstrates this notion by applying a simple color scheme to overlapping nuclei instances and then determining their horizontal and vertical distances to identify them from surrounding overlapping nuclei instances.

The segmented nuclei are covered by the classification branch, which works on them to classify them into the relevant classifications for our research. When the entropy loss is employed for classification, several loss functions are used. The regression loss function is computed by the HoVer branch for its output. This is accomplished essentially by computing the mean squared error between the predicted nuclei distances computed in both the horizontal and vertical directions and comparing them to the Ground truth labels available for our weakly supervised methods.

The loss functions of the nuclear pixel and the classification branch are very similar. For the NP and NC branches, we compute the cross-entropy loss and the dice loss. The total loss of the three branches is then added together to provide a cumulative loss.

Chapter 4: Implementation and Test Cases

4.1 Implementation

For implementation, we chose the BreastNet and PMNet architectures. BreastNet was chosen on the basis of its claimed high accuracy. PMNet was selected due to its dynamic nature that only classified cancer but also provides effective segmentation of the regions of interest.

For this deliverable, our goal was to perform a 2x2 comparative analysis as follows:

1. Original: BACH dataset on PMNet model
2. Original: BreakHis dataset on BreastNet model
3. Cross-over: BreakHis dataset on PMNet model
4. Cross-over: BACH dataset on BreastNet model

4.1.1 Implementation of BreastNet

The BreastNet model was successfully run on the BreakHis dataset using the publicly available code and datasets. Few changes were however made which are stated as follows.

4.1.1.1 Code Changes

These hard-coded changes were required to make the code working on the systems available to us. The code made publicly available was run and provided on Jupyter Notebook which required these few hard-coded changes to be made to make the code compatible with Google Colaboratory.

- 1) TensorFlow's built-in optimizer function "Adam()" was replaced with the built-in optimizer function "tf.keras.optimizers.Adam".
- 2) TensorFlow's built-in function "_keras_shape()" was replaced with the built-in function "shape()" throughout the code.
- 3) The value of the global variable "BASE_DIR" was set accordingly to allow the program to access the BreakHis dataset from our personal Google Drive folder. This was necessary to do because the code on GitHub is able to directly access the BreakHis dataset from Kaggle through an availed license.

4.1.1.2 Other Changes

- 1) The entire BreakHis dataset of 5.28GB was downloaded from kaggle and uploaded on our personal Google Drive. This was necessary to do as the original code on GitHub directly accessed the BreakHis dataset on Kaggle through an availed license. This dataset was however to be downloaded and maintained on Google Drive in its original structure as part of the BreastNet code created path directories by concatenation of every different source file available to the base directory in the BASE_DIR variable.

Conclusion

- 2) The above point also explained the reason to download the BreastNet code and run it on our personal Google Colaboratory files as the directories needed to be changed for each folder.

4.1.1.3 Results

The BreastNet code was successfully run for almost all epochs on the BreakHis dataset giving away very similar results of the values generated as the model learned from the images.

4.1.2 Crossover: BACH dataset on BreastNet model

This goal was achieved successfully. The BACH dataset was run on the BreastNet model thereby achieving a comparative analysis between the results of scenario i. in the research paper and our produced results in iv. In order to do so, the following issues were needed to be overcome.

4.1.2.1 Issues in Nature of Data

1. BreastNet was made to work on the BreakHis dataset having the following image properties: 2 classes: 2480 Benign and 5429 Malignant images, each class having further 4 sub-classes made according to cell diameter and shape, .png format, 700x460 pixels, un-annotated images, 4 different magnification rates: 40X, 100X, 200X and 400X.
2. PMNet was made to work on the BACH dataset having the following image properties: 4 classes: 100 Benign, 100 Normal, 100 In Situ and 100 Invasive, no further sub-classes, .tiff format, 2048x1536 pixels, annotated images, magnification rate in 200X only
3. It was crucial to overcome the above mentioned difference in properties of the datasets in order to make the BACH dataset compatible to work on the BreastNet program code.

4.1.2.2 Issues in Structure of Data

1. The BreakHis dataset is structured across 2 main classes: Benign and Malignant. The images in both these classes are further classified across 4 sub-classes each according to the cell diameter and shape. Each of those sub-classes contains multiple folders which represent the slide number for the contained images. Each of those folders contains 4 folders where images of each sample in each slide with the magnification rates of 40X, 100X, 200X and 400X are spread across these folders. Each image is named following a specific naming convention identifying the class, sub-class, slide number, magnification rate and image number of that image.
2. The BACH dataset is structured across 4 main classes: Benign, Normal, In Situ and Invasive. Each of these classes contains 100 WSIs and 100 microscopy images of the same samples. The remaining properties of all images across these 4 folders are the

Exploring Deep Learning Models for Cancer Detection and Classification

same. Each WSI image along with it contains an .xml file containing the annotations for that image with features like coordinates, label area, perimeter of region and pixel scale.

4.1.2.3 Solution Resolution

1. The Normal and Invasive classes were selected from the BACH dataset to narrow down the data to feed down to 2 classes (200 images) and resemble the main classes of Benign and Malignant of the BreakHis dataset, respectively.
2. The 9GB BACH dataset was downloaded from the BACH 2018 Challenge website and uploaded on our personal Google Drive.
3. Online 'tiff' to 'png' converters were used to convert the entire dataset. Multiple online converters were used and checked for quality of image type conversion performed to retain maximum pixels of the WSIs.
4. The BreastNet architecture does not require feeding of pre-annotated images so this was not a concern.
5. All 200 images were renamed individually with the same naming conventions as the BreakHis dataset. The images were also kept in similar structured folders in order to mimic the structure of the BreakHis dataset. This was done in order to minimize the hard-coded changes required to be made in the BreastNet code as the code created path directories by concatenations.
6. The BACH dataset contained no sub-classes. Since the images in the BACH dataset were un-annotated, it was not possible to determine which sub-class the image would belong to. Hence, only one subclass was maintained for each of the main classes (Benign and Malignant), regardless of the cell diameter and shape. This did not prove to be an issue for us as the BreastNet code accommodates equal amounts of learning from each of these sub-classes.
7. In order to accommodate for the changes made in point 6 above, the tailored-class of BreakHis_DATASET containing the `get_paths_n_labels()` method was changed. This method creates labels for each file directory in the dataset for each specific image following the rigorous naming convention chosen. The number of subclasses in the class members '`'self.BENIGN_SUB_CLASSES'`' and '`'self.MALIGNANT_SUB_CLASSES'`' were reduced down from 4 each to 1 each.
8. In order to accommodate for the changes made in point 7 above, a part of the entire path directory was to be maintained in a '`.stat.txt`' file in the Google Drive.
9. The hard-coded values of the parameters '`'EPOCHS'`', '`'N_SPLITS'`', '`'SEED'`' and '`'TRAIN_TEST_RATIO'`' were changed in accordance with the performance on each combination of discrete values. The value of the class member '`'DATASET_MODE'`' was set to '`'200X'`' due to the nature of the BACH dataset.

Conclusion

4.1.2.4 Results

The BreastNet architecture was successfully run on the BACH dataset giving the best results on the original parameter values of EPOCHS = 100, N_SPLITS = 5, SEEDS = 1881, TRAIN_TEST_RATIO = 0.2. The best result obtained was an f1-score of 0.9667.

4.1.3 Implementation of PMNet

The PMNet model was successfully implemented as in [15]. The model was trained on the BACH dataset and validated on the Dryad dataset. This was managed using the publicly available code and datasets. The results obtained were in accordance with those mentioned in [15].

This BreaKHis dataset has not yet been incorporated. The reasons for not being able to do so were quite understandable due to the specific nature of the PMNet code and the BreakHis dataset.

The PMNet model performs training on the BACH dataset and validation on the Dryad dataset. The involvement of the Dryad dataset in this comparative analysis was considered. However, it was noteworthy that the BreastNet architecture does a train/test split rather than a train/validate/test split. So the Dryad dataset was not involved in this comparative analysis.

The PMNet architecture is made to work successfully and derive its intended accuracy by feeding on annotated WSIs and patch level images using information of these annotations available in .xml files for each of the 10 patients in the BACH dataset. On the other hand, the BreakHis dataset contains microscopy images with no annotations available. These annotations can only be done by the providers of this primary data who are supposed to be a team of specialists. With no annotations available, the PMNet model is unable to learn from any of these images. Performing any augmentation on the BreakHis dataset would be unhelpful without any access to these annotations.

4.1.4 Expert System

The implementation of this expert system would be based on a React based front-end Web-Application that acts as a portal to allow the medical professionals access to this Network that can process vital information to assist in medical analysis and diagnosis. Ideally, for this purpose, an API would be required that can be hit by multiple platforms to get results. Our goal is to provide the Proof of Concept for this system (end-to-end) however, in the ideal situation our expert based system would be shadowed by an API that provides a tangible route to access the processing of imagery and result set that this model can generate.

Figure 17 shows screenshots from our prototype GUI:

Exploring Deep Learning Models for Cancer Detection and Classification

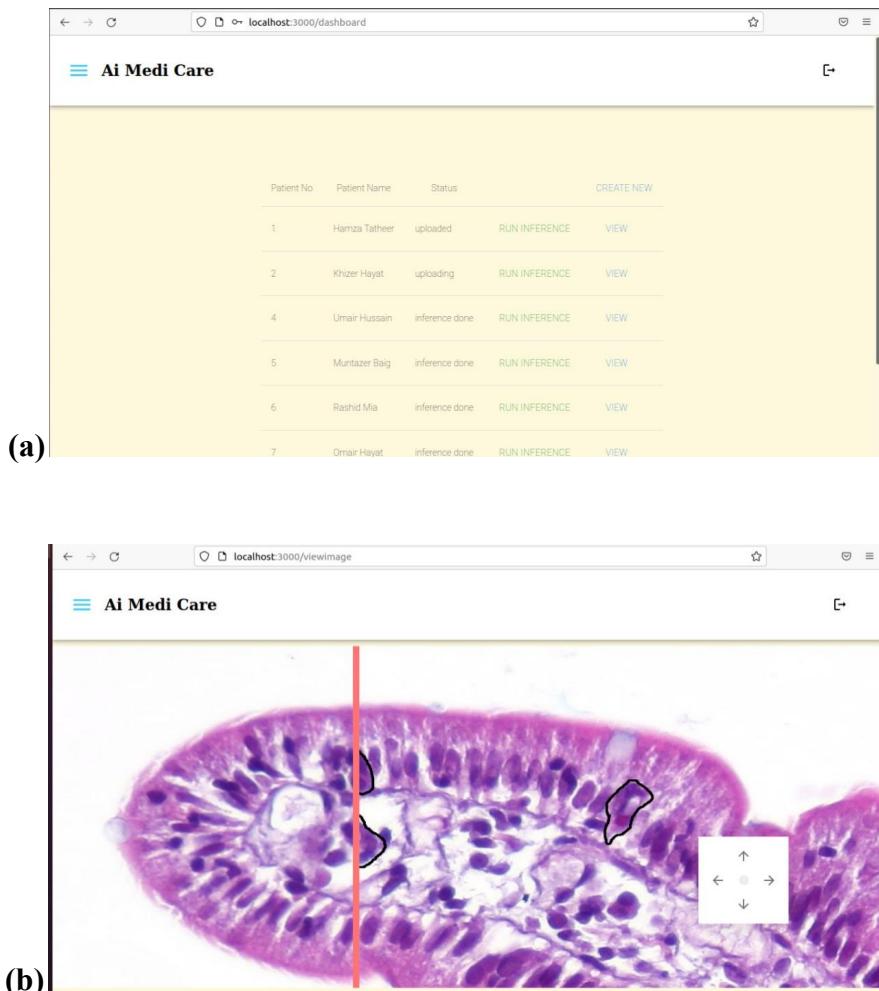


Figure 17: Screenshots from proposed Expert System

- (a) *Patients list fetched from database.*
- (b) *Real-time inference performed by the expert system using a slider window.*

4.2 Test Case Design and Description

All the test cases mentioned here require Google Chrome version 45+, Mozilla Firefox version 38+, Opera version 9+, Internet Explorer 10+ and Microsoft Edge 12+ to run. For Windows operating systems Windows 7 and later versions are supported. Moreover, an internet connection is required for all test cases.

4.2.1 Segmentation Ground Truth Validation

Type Map Validation			
Expert System			
Step No.	Execution description	Procedure result	
1	User converts the mask to numpy and matches the color region with the class	Generates type map which allocates class to each pixel of the image	
Comments:	Passed <input checked="" type="checkbox"/> Passed <input type="checkbox"/> Failed <input type="checkbox"/> Not Executed		

4.2.2 Upload and View WSI

User Interface			
Expert System			
Step No.	Execution description	Procedure result	
1	User uploads WSI to Expert System	Renders visible portion of WSI onto screen	
Comments:	Passed <input checked="" type="checkbox"/> Passed <input type="checkbox"/> Failed <input type="checkbox"/> Not Executed		

4.2.3 Scroll to different portion of WSI

User Interface Expert System					
Test Case ID:	3	QA Test Engineer:	<i>Mohammad Ahmed Irfan</i>		
Test case Version:	1.0	Reviewed By:	<i>Hamza Tatheer</i>		
Test Date:	18/5/2022	Use Case Reference(s):	<i>N/A</i>		
Revision History:	<i>None</i>				
Objective	<i>To test the ability to scroll to different portions of WSI</i>				
Product/Ver/Module:	<i>Data Augmentation</i>				
Environment:	<ul style="list-style-type: none"> • Ubuntu 20.04.2 LTS • Pytorch version 1.7.0 • CUDA Version 11.0 				
Assumptions:	<i>User familiar with use of scroll pad</i>				
Pre-Requisite:	<i>WSI has been uploaded successfully</i>				
Step No.	Execution description	Procedure result			
1	<i>User clicks on scroll pad to move up, down, left or right</i>	<i>View changes to render previously invisible portion of WSI as per user scroll request</i>			
Comments:	<i>Passed</i>				
<input checked="" type="checkbox"/> Passed <input type="checkbox"/> Failed <input type="checkbox"/> Not Executed					

4.2.4 WSI Nuclei Segmentation

HoVerNet Expert System					
Test Case ID:	4	QA Test Engineer:	<i>Khizar Hayat Saani</i>		
Test case Version:	1.0	Reviewed By:	<i>Hamza Tatheer</i>		
Test Date:	18/5/2022	Use Case Reference(s):	<i>N/A</i>		
Revision History:	<i>None</i>				
Objective	<i>To test the segmentation of WSI using HoVerNet</i>				
Product/Ver/Module:	<i>Backend</i>				
Environment:	<ul style="list-style-type: none"> • Ubuntu 20.04.2 LTS • Pytorch version 1.7.0 • CUDA Version 11.0 				
Assumptions:	<i>User familiar with nuclei classes</i>				
Pre-Requisite:	<i>WSI successfully uploaded</i>				
Step No.	Execution description	Procedure result			
1	<i>User clicks on 'Perform Inference' button</i>	<i>Renders same portion of WSI with nuclear segmentation and classification</i>			
Comments:	<i>Passed</i>				
<input checked="" type="checkbox"/> Passed <input type="checkbox"/> Failed <input type="checkbox"/> Not Executed					

Conclusion

4.2.5 Scroll to different portions of segmented WSI

User Interface					
Expert System					
Test Case ID:	5	QA Test Engineer:	Mohammad Taifoor Saleem		
Test case Version:	1.0	Reviewed By:	Hamza Tatheer		
Test Date:	18/5/2022	Use Case Reference(s):	N/A		
Revision History:	None				
Objective	To test the ability to scroll to other portions of WSI after inference				
Product/Ver/Module:	Backend				
Environment:	<ul style="list-style-type: none"> • Ubuntu 20.04.2 LTS • Pytorch version 1.7.0 • CUDA Version 11.0 				
Assumptions:	User familiar with scrolling pad use				
Pre-Requisite:	Inference has been performed on the WSI				
Step No.	Execution description	Procedure result			
1	User clicks on the scroll pad to move up, down, left or right.	Renders relevant portions of WSI with segmentation and classification performed.			
Comments:	Passed				
<input checked="" type="checkbox"/> Passed <input type="checkbox"/> Failed <input type="checkbox"/> Not Executed					

4.2.6 Response Time Testing

HoVerNet					
Expert System					
Test Case ID:	6	QA Test Engineer:	Mohammad Ahmed Irfan		
Test case Version:	1.0	Reviewed By:	Hamza Tatheer		
Test Date:	18/5/2022	Use Case Reference(s):	N/A		
Revision History:	None				
Objective	To test the time taken for inference is less than or equal to 1 minute				
Product/Ver/Module:	Backend				
Environment:	<ul style="list-style-type: none"> • Ubuntu 20.04.2 LTS • Pytorch version 1.7.0 • CUDA Version 11.0 				
Assumptions:	User familiar with Expert System				
Pre-Requisite:	User has successfully uploaded WSI				
Step No.	Execution description	Procedure result			
1	User clicks on 'Perform Inference' button	Renders relevant portion of WSI with nuclear segmentation and classification under 1 minute			
Comments:	Passed				
<input checked="" type="checkbox"/> Passed <input type="checkbox"/> Failed <input type="checkbox"/> Not Executed					

4.3 Test Metrics

Following are the test metrics and their results after the testing process.

Metric:	Purpose
Number of Test Cases:	6
Number of Test Cases Passed:	6
Number of Test Cases Failed:	0
Test Case Defect Density:	0
Test Case Effectiveness:	0

Chapter 5: Experimental Results and Analysis

5.1 Results and Discussion

Table 4 summarizes the result obtained in the comparative analysis of BreastNet using the BreaKHis and BACH datasets.

Table 4: Result summary for BreastNet using BreaKHis and BACH

The performance of BreastNet, in terms of F1-Score, accuracy, sensitivity and precision, as proved in its original research.

Test Metric	BreaKHis	BACH
F1- Score	98.59	96.67
Accuracy	98.80	96.60
Sensitivity	98.35	96.62
Precision	98.84	96.71

Here, we observe that the model performs better on the BreaKHis dataset. However, it is worth noting that the BACH dataset is considerably smaller. Even so, the accuracy obtained is considerably high showing promise that the model holds the capacity to be generalized for a variety of input types.

Table 5 summarizes the results obtained in the comparative analysis of BreastNet versus PMNet using the BACH dataset.

Table 5: Result summary for BreastNet vs PMNet using BACH dataset

The performance comparison, of F1-Scores, between PMNet using BACH dataset and the crossover of Breastnet using BACH dataset.

Test Metric	BreastNet	PMNet
F1- Score	96.67	88.9

Here, we observe that BreastNet performs better than PMNet on the BACH dataset. The F1-score obtained using BreastNet is considerably higher.

Meanwhile, HoVerNet has proven to perform significantly well in terms of both segmentation and classification compared to PMNet on the CoNSeP dataset as shown in Table 6.

Table 6: Result summary for HoVerNet vs PMNet using CoNSeP dataset

The performance comparison, in terms of detection and segmentation quality, between HoVerNet and PMNet using the CoNSeP dataset.

Test Metric	HoVerNet	PMNet

Detection Quality	77.4	11.0
Segmentation Quality	77.8	69.8

Chapter 6: Conclusion

In order to perform successful comparative analysis, the criteria to select and involve any deep learning models and datasets would highly be based on how similar the nature of our datasets is. The nature of these datasets would include a number of measurable properties like number of images, number of classes and subclasses in datasets, image formats, pixel sizes, number of pixels, image dimensions, availability of annotations, magnification rates, memory scale..

Our team has been able to come up with many ways to perform comparative analysis between models and datasets. Two of those options were as follows: either to make appropriate changes in our selected deep learning models to be compatible with each selected dataset individually, or make appropriate augmentations in our selected datasets to make them feedable into each of our selected models. However, we have realized that the safest way to perform comparative analysis is to keep the program codes of our selected deep learning models as close to their original state as possible.

While developing our Expert System, we were able to make the following deduction which are key to developing a system that can be used on a large scale within the medical industry:

1. The system cannot be a black-box, i.e a mere classifier.
2. The system must be generalized, not limited to particular type of cancer only
3. The system must be robust, responsive, easy to use as most medical professionals lack in depth tech knowledge.
4. The system must produce results instantly, ideally in less than 1 minute while using minimal resources.
5. The system must be accurate, given that people's lives could be on stake.

These deductions and the prior comparative analysis led us to use HoVerNet as the inference model for AIMediCare, the proposed Expert System. HoVerNet is a generalized nuclei segmentation model that segments nuclei within an histopathological image in four classes. Using the segmented image, pathologists are able to efficiently diagnose the presence and stage of the cancer.

Owing to its generalized training on the Kumar Dataset, HoVerNet shows the most promise of being used as the goto inference system for Expert Systems in cancer diagnosis. Our system provides a complete, easy to use interface compatible with HoVerNet. Future researchers may use this scalable interface to test newer, better models without worry of

Conclusion

developing an efficient front end. They may also continue to work on improving the existing model to provide better results.

References

- [1] T.R. Geiger, D.S. Peepo, et. al., “Metastasis Mechanisms,” *Biochimica et Biophysica Acta*, 1796.
- [2] X.Y. Liew, N. Hameed, J. Clos. et. al., “A Review of Computer-Aided Expert Systems for Breast Cancer Diagnosis,” *Cancers* 2021, vol. 2764, no. <https://doi.org/10.3390/cancers13112764>
- [3] M.Z. Alom, C. Yakopcic, et. al., “Breast Cancer Classification from Histopathological Images with Inception Recurrent Residual Convolutional Neural Network,” *J. Digit. Imaging*, vol. 32, 2019.
- [4] F.A. Spanhol, L.S. Oliveira, et. al., “A Dataset for Breast Cancer Histopathological Image Classification,” *IEEE Trans. Biomed. Eng.*, vol. 63, 2016.
- [5] T. jo Arau, G. Aresta et. al., “Classification of breast cancer histology images using Convolutional Neural Networks,” *PLoS ONE*, vol. 12, no. 6, 2017.
- [6] Guilherme Aresta, Teresa Araújo, et al, BACH: Grand challenge on breast cancer histology images, Medical Image Analysis, Volume 56, 2019
- [7] M. Macenko, M. Niethammer, et. al., “A method for normalizing histology slides for quantitative analysis,” *IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, 2009.
- [8] D.M. Vo, N.Q. Nguyen, et. al., “Classification of Breast Cancer Histology Images Using Incremental Boosting Convolution Networks,” *Inf. Sci.*, vol. 482, 2019.
- [9] I. Ud Din, N. Islam, et. al., “A Novel Deep Learning based Framework for the Detection and Classification of Breast Cancer Using Transfer Learning,” *Pattern Recognition Letters*, vol. 125, pp. 1-6, 2019.
- [10] R. Kumar, R. Srivastava, et. al., “Detection and Classification of Cancer from Microscopic Biopsy Images Using Clinically Significant and Biologically Interpretable Features,” *J. Med. Eng.*, 2015.
- [11] M. Toğançar, K.B. Özkurt, et. al., “BreastNet: A Novel Convolutional Neural Network Model through Histopathological Images for the Diagnosis of Breast Cancer,” *Phys. A Stat. Mech. Its Appl.*, p. 545, 2020.
- [12] M. Geist, “Soft-max boosting,” *Mach. Learn*, vol. 2958, p. 305–332, 2015. [16] I. K. e. a. H. Kataoka, “Feature Evaluation of Deep Convolutional Neural Networks for Object Recognition and Detection,” 2015.
- [13] I. K. e. a. H. Kataoka, “Feature Evaluation of Deep Convolutional Neural Networks for Object Recognition and Detection,” 2015.
- [14] K. He , X. Zhang , et. al., “Deep residual learning for image recognition,” *IEEE conference on computer vision and pattern recognition*, p. 770–778, 2016.
- [15] Salman Ahmed, Maria Tariq, Hammad Naveed, PMNet: A probability map based scaled network for breast cancer diagnosis, Computerized Medical Imaging and Graphics, Volume 89, 2021.
- [16] “Gleason Score and Grade Group,” [Online]. Available: <https://www.pcf.org/about/prostate-cancer/diagnosis-staging-prostate-cancer/gleason-score-isup-grade>.

- [17] R. Cao, and A. M. Bajgiran and et al., “Joint Prostate Cancer Detection and Gleason Score Prediction in mp-MRI via FocalNet,” *IEEE Transactions on Medical Imaging*, vol. 38, no. 11, Nov. 2019.
- [18] T. Nolan, “SPIE-AAPM-NCI PROSTATEx Challenges (PROSTATEx),” SPIE AAPM-NCI PROSTATEx Challenges (PROSTATEx), 21 November 2016. [Online]. Available: <https://wiki.cancerimagingarchive.net/pages/viewpage.action?pageId=23691656#23691656d4622c5ad5884bdb876d6d441994da38>.
- [19] G. Yang, G. Li, et. al., “Automatic Segmentation of Kidney and Renal Tumor in CT Images Based on 3D Fully Convolutional Neural Network with Pyramid Pooling Modul,” in *24th International Conference on Pattern Recognition (ICPR)*, , August 2018.
- [20] Simon Graham, Quoc Dang Vu, Shan E Ahmed Raza, Ayesha Azam, Yee Wah Tsang, Jin Tae Kwak and Nasir Rajpoot, HoVer-Net: Simultaneous Segmentation and Classification of Nuclei in Multi-Tissue Histology Images, 2019