Deep Learning Classifiers to identify the benign and malignant state from the Breast Cancer Histopathological Database

Renk Asik[1†], Leon Igal Inselberg Barbut[1†], Payal Priyadarshini[1†]

[1]Department of Mathematics and Computer Science, Freie Universität Berlin, Takustraße 9, Berlin, 14195, Germany.

Contributing authors: renka01@zedat.fu-berlin.de; leonigai01@zedat.fu-berlin.de; payap02@zedat.fu-berlin.de;
[†]These authors contributed equally to this work.

Abstract

Breast cancer is one of the leading types of cancer among women. Early detection and accurate classification of this disease can significantly improve patient outcomes and recovery. Advances in machine learning and deep learning techniques have enabled researchers to develop tools that can effectively detect tumors and abnormalities using image datasets, while also classifying them with precision. In this study, we focused on the BreakHis dataset, which consists of 7,909 valid observations, including 2,480 benign samples and 5,429 malignant samples. Three deep learning classifiers—Convolutional Neural Network (CNN), Shallow Neural Network (SNN), and ResNet50—were developed, and their performance was evaluated in terms of their effectiveness in classifying histopathological images into two categories: benign and malignant. Data augmentation was performed, and the results indicated that it enhanced the performance of the Shallow Neural Network classifier, which is particularly useful when drawing model accuracy from a limited dataset. Additionally, histograms and ROC curves were plotted, demonstrating that ResNet50, with its strong feature extraction capabilities, was the most accurate model, outperforming both the CNN and SNN on both augmented and non-augmented data. Furthermore, images captured at 100x magnification yielded better classification results compared to those captured at 400x magnification.

Keywords: Deep Learning, Breast cancer, Image classification, Histopathological image analysis, Computer aided diagnostics

1 Scientific Background

Breast cancer remains one of the most common and fatal diseases affecting women worldwide.[1] The death rates associated with breast cancer in many countries have consistently increased over the years, reaching alarming levels.[2] Breast cancer is characterized by the uncontrolled proliferation of cells, which can develop into benign (non-cancerous) or malignant (cancerous) tumors.[3] Treatment options for breast cancer are often less effective once the disease reaches advanced stages, making early detection a critical factor in improving patient outcomes. Early detection plays a vital role in preventing the rapid progression of the disease, as it allows for timely and targeted interventions, thereby increasing the chances of survival and recovery.[4]

In the detection of breast cancer, deep learning technologies have become indispensable, particularly when analyzing complex tabular and image datasets.[5] Deep learning, a subset of machine learning, leverages neural networks that mimic the learning patterns of the human brain. These models are capable of identifying intricate patterns and drawing conclusions from large datasets by processing features through multiple interconnected layers.[6] In this study, we utilized an image dataset containing intricate details and patterns that were challenging for traditional machine learning algorithms and experts to consistently interpret. Conventional methods are often time-consuming and prone to inaccuracies in predicting tumor cells from image datasets. Deep learning models, such as Convolutional Neural Networks (CNN), offer a more sophisticated architecture with a higher potential for learning from the data.[7]

The data used in this study was collected using standard histopathological analysis, a reliable and well-established method for understanding the progression of breast cancer.[8] Histopathology scans help medical professionals identify anomalies in tissue samples and provide crucial insights into cellular conditions. By using high-quality image data for deep learning models, this study aims to enhance the automatic classification of histological images, improving diagnostic consistency and accuracy.

In this research, we employed CNN, Shallow Neural Network (SNN), and ResNet50 classifiers to categorize images in the histopathological dataset. The resolution of images plays a significant role in model efficacy, and the performance of the classifiers was evaluated across different magnification levels. Additionally, data augmentation techniques were applied to increase data variability and enhance classifier performance, particularly for the Shallow Neural Network. This study highlights the impact, application, and results of using CNN, SNN, and ResNet50 classifiers for breast cancer detection.

2 Goal

The primary goal of this study was to implement deep learning neural networks to precisely segregate the breast cancer histopathological images into two groups: benign and malignant. The steps involved developing and then evaluating three deep learning classifiers: Convolutional Neural Network (CNN), Shallow Neural Network (SNN), and ResNet50. By implementing these models, we aim to further contribute to the automated diagnosis with minimal manual interruption.

3 Data

The Breast Cancer Histopathological Image Classification (BreakHis) database is a comprehensive collection which has been built in corporation with the P-D Laboratory, Parana, Brazil[8]. The creation of the BreaKHis dataset was intended to address the issue of histopathology image analysis being conducted on small datasets that are not accessible to the scientific community.

Magnification	Benign	Malignant	Total
40x	652	1,370	1,995
100x	644	1,437	2,081
200x	623	1,390	2,013
400x	588	1,232	1,820
Total images	2,480	5,429	7,909

Table 1: The BreakHis dataset consists of four magnificant levels as shows in the table

The dataset consists of 7909 valid observations from 82 patients out of which 2480 are benign tumor images and 5429 are malignant tumor images. Both benign and malignant tumor images are categorized into four subtypes, namely adenosis, fibroadenoma, phyllodes tumor and tubular adenoma for benign tumors; ductal carcinoma, lobular carcinoma, mucinous carcinoma and papillary carcinoma for malignant tumors. These subcategories are essential for identifying the various tissues where the tumors may be located. All images were captured at four magnification levels (40x, 100x, 200x, 400x) with a resolution of 700x460 pixels, formatted in PNG.

For this study, from the available magnification levels, we focused on the 100x and 400x magnification levels and we did not target any specific subcategory. We trained our models to classify each given image into benign or malignant tumors.

4 Preprocessing

In this study, data augmentation was the sole preprocessing step employed. This step was used to expand the training dataset by applying difference transformations to the images. Our goal was to demonstrate the difference in performance of the various classifiers before and after data augmentation.

4.1 Data Augmentation

Data augmentation refers to a set of algorithms designed to generate synthetic data from an existing dataset, effectively increasing the dataset. The synthetic data includes small modifications that the model's predictions should remain consistent against.[9] The goal is to improve model optimization and generalization.

We employed 3 data augmentation strategies for our training set in this study. First we used horizontal random flip, which randomly flips the images horizontally, helping the model learn features that are invariant to left-right orientation. Random rotation(x) rotates images by up to the specified percentage x, allowing the model to recognize patterns regardless of rotations. We used a rotation cap of 20 percent. Lastly, we employed random zoom(x), which applies a random zoom to the images up to the specified percentage x, training the model to handle varying scales in object sizes. We also specified a cap of 20 percent for the random zoom.[10]

5 Methods

5.1 Exploratory analysis

To familiarize ourselves with the data, we visualized the distribution of images across subcategories within the dataset. This step primarily involved comparing images and understanding the variations in color, texture, and size in the images.

5.2 Train, Test and Validation

We created our training, testing, and validation sets using the*train_test_split* function from the scikit-learn library in Python[11]. This function splits arrays into random training and testing subsets. Seventy percent of the images in the dataset were randomly allocated to the training set, while the remaining 30 percent were evenly divided between the testing and validation sets. The *test_size* parameter was used to ensure the correct proportions, and the *stratify* parameter was applied to maintain the original ratio of benign to malignant images in each subset.

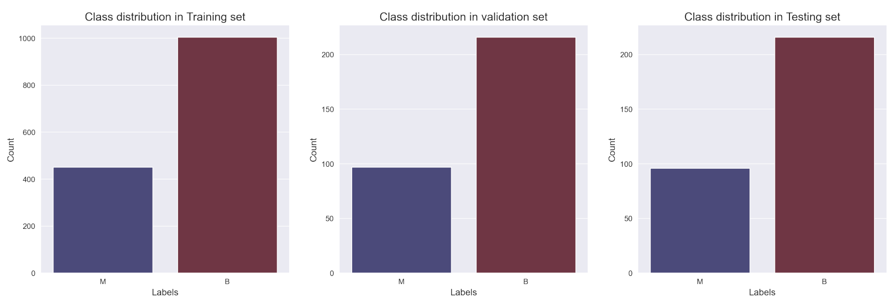


Fig. 1: Malignant(blue) and Benign(brown) samples distribution in Train,test and validation,for with magnification level 100x.

Fig. 2: Malignant(blue) and Benign(brown) samples distribution in Train,test and validation,for with magnification level 400x.

5.3 Classifiers

The classifiers in this study are deep learning models designed to learn patterns from sample data, enabling them to categorize new data into predefined classes. These classifiers play a critical role in tasks such as image classification, providing accurate results for important and timely medical diagnoses. We evaluated three classifiers in our study based on key performance *metrics: accuracy, loss, and recall (sensitivity)*. The evaluations considered variations in 100x and 400x magnification levels as well as the impact of data augmentation.

5.3.1 Convolutional Neural Network

Convolutional Neural Network (CNN) is a type of deep learning neural network designed to process and analyze data. It has the ability to detect spatial hierarchies through convolutional layers. A typical CNN structure works by first using convolutional layers with filters to extract features from images, followed by pooling layers to reduce spatial dimensions, thereby making the data more efficient to process. Finally, fully connected layers combine the extracted and compressed features to classify images into designated categories, functioning as the decision-making part of the model.[12]

In this study, the CNN model was designed with multiple layers to analyze raw input image data and learn patterns that distinguish between benign and malignant tumors. It was built using TensorFlow and Keras libraries, with an aim to balance the efficiency of the outcome. In order to generalize and avoid over fitting, augmentation tools such as zooming, flips and rotations were implemented.

The model expects input images with dimensions of 32×32 pixels and 3 color channels (RGB). Initially, a convolutional layer with 32 filters of size 3×3, activated by the *ReLU* function, captures local patterns in the images. This is followed by a max-pooling layer with a 2×2 window, which reduces the spatial dimensions by half, summarizing key features. The model continues with a second convolutional layer, now with 64 filters, and another max-pooling layer to further compress and extract deeper features. A third convolutional layer with 64 filters refines feature extraction.

Next, a flattening layer converts the 2D feature maps from the convolutional layers into a 1D vector, preparing the data for the fully connected layers. A dense layer with 64 neurons and *ReLU* activation then interprets the flattened features, helping the model to identify more abstract patterns. Finally, a single-neuron dense layer with *sigmoid* activation outputs a probability, enabling binary classification by indicating whether an image is benign or malignant. As each image is processed through these layers, the model progressively learns complex features, thereby enhancing predictive accuracy. Adam optimizer and binary cross entropy function were used while compiling the model and 10 epochs were utilized while fitting the model.

5.3.2 Shallow Neural Network

Shallow Neural Network (SNN) is a type of neural network in deep learning with a straightforward architecture, with one or few hidden layers. This simplicity differentiates it from deeper neural networks (such as CNNs) by focusing on fewer transformations of the input data, which can sometimes make it faster to train while still providing meaningful results in basic classification tasks. For this study, we used *ReLU* (Rectified Linear Unit) as the activation function in the hidden layer, which is effective for learning in image data contexts due to its ability to model complex patterns without saturating. [13]

The structure of this shallow neural network consisted of three main layers: the input layer, a single hidden layer, and the output layer. The input layer was configured for images with dimensions of 32×32 pixels, with the data reshaped into a 1D vector to prepare it for the fully connected (dense) layers, much like the preprocessing in CNNs. The hidden layer contained 64 filters and used *ReLU* activation, allowing it to extract core features from the data. Finally, the output layer, with a single unit and *sigmoid* activation, provided a binary classification output, indicating whether the image was classified as benign or malignant.

The model was trained for 10 epochs using the Adam optimizer to minimize binary cross-entropy loss.

5.3.3 ResNet50

ResNet50, also known as Residual Network-50, is a deep convolutional neural network architecture developed by Microsoft. This advanced model enables detailed analysis and highly accurate classification of complex image data. ResNet50 is composed of 50 layers, with 48 convolutional layers followed by a max pooling layer and an average pooling layer. The architecture is organized into *residual blocks*, which introduce a unique shortcut connection allowing the model to learn residual mappings instead of directly learning features. This residual approach helps the network mitigate issues of vanishing gradients, enabling it to train effectively even with a significant depth. Each residual block contains multiple convolutional layers, batch normalization, and ReLU activation functions, allowing it to capture intricate patterns and features.[14]

In our study analyzing breast cancer data, we included ResNet50 as the base model, inputting images with a shape of 224×224×3. We configured the model without the final classification layer, so that we could add custom layers tailored to our binary classification task. Specifically, we followed the base model with a *GlobalAveragePooling2D* layer to reduce the output to a single vector, a structure we used in prior classifiers for efficient dimensionality reduction. This was followed by a dense layer with a *sigmoid* activation function for binary classification to distinguish between benign and malignant tumor images.

The model was initially trained for 10 epochs using the Adam optimizer, minimizing binary cross-entropy loss. During this phase, we froze the base ResNet50 model layers to ensure that only the added layers were updated.

After this initial phase, we proceeded to fine-tune the model by unfreezing the ResNet50 layers and retraining it with a reduced learning rate. Fine-tuning allows the model to adapt its pre-trained features to the specific patterns in our dataset, improving its overall performance.

5.4 Libraries

The *Pandas*[15] library in Python was utilized for its *DataFrame* structure, and the *os*[16] library was primarily used for its *walk* function to locate individual images within the dataset. We employed the *numpy* library for its *array*[17] structure and its ability to count unique labels.

For data splitting, the *sklearn* library, specifically the *model_selection*[18] submodule, was used with its *train_test_split* function to divide the dataset into random training, testing, and validation sets. We also utilized the *metrics*[19] submodule for generating confusion matrices, ROC curves, and calculating the area under the curve (AUC).

The *matplotlib.pyplot*[20] and *seaborn*[21] libraries were applied for visualizations such as heatmaps and bar plots.

Our primary library for model creation and evaluation was *tensorflow*[22]. *Tensorflow.keras*[23] facilitated the building of models and layers, and it includes a built-in *ResNet50* function, which allowed us to construct the base of our ResNet50 classifier, which we then customized for our specific requirements

5.5 Evaluation

To evaluate our models, we used the test set that was created alongside the training and validation sets. After training, each model was assessed on this test set to measure its performance on data it had not seen during training, with the *evaluate*[24] function from the tensorflow.keras library. For this evaluation, we used a set of key metrics—Loss, Accuracy, Precision, and Recall (Sensitivity)—to provide a comprehensive understanding of the model's predictive capabilities.

The evaluation process involved running the test data through the model and calculating these metrics. Loss, a measure of the model's prediction error, gives insight into how well the model generalizes. Accuracy reflects the overall percentage of correct predictions, while Precision measures the proportion of true positive predictions among all positive predictions made. Lastly, Recall (or Sensitivity) indicates the model's ability to correctly identify all true positive cases.

6 Results and Discussion

6.1 Classifier Comparison

Firstly, we compare classifiers using non-augmented, 100x magnified data.

The barplot (Figure 3) comparing loss, accuracy, precision and recall of different classifiers, highlights that ResNet50 is the best performing classifier across all metrics. ResNet50 has an accuracy of 0.93, recall of 0.98 and precision of 0.92. Moreover, SNN has higher accuracy and precision compared to CNN. However, CNN shows a slightly better recall than SNN.

Figure 4 compares the ROC curves of our three classifiers. ResNet50 again performs significantly better with an AUC of 0.98 compared to Shallow NN (AUC = 0.86) and CNN (AUC = 0.88).

The superior performance of ResNet50 can be attributed to its deeper architecture and therefore ability to learn more complex features. Its skip connections (shortcuts in neural networks that allow information to bypass one or more layers) mitigate the vanishing gradient problem, enabling better optimization and feature extraction.[25] In contrast, SNN and CNN, being simpler architectures, may struggle to capture intricate patterns in the data.

6.2 Effect of Augmentation

In this section we investigate the effect of using augmented data instead of the original data to train our classifiers.

Table 2 and 3 depicts the metrics calculated for augmented and original data. The metrics measured for ResNet50 change only slightly. However CNN and SNN get affected more significantly from the data augmentation. Accuracy (0.79 vs. 0.82) and precision (0.77 vs. 0.89) of CNN increases with the augmentation. However recall rate drops dramatically from 0.98 to 0.82. For the SNN, accuracy and precision increase slightly, whereas recall rate mildly decreases.

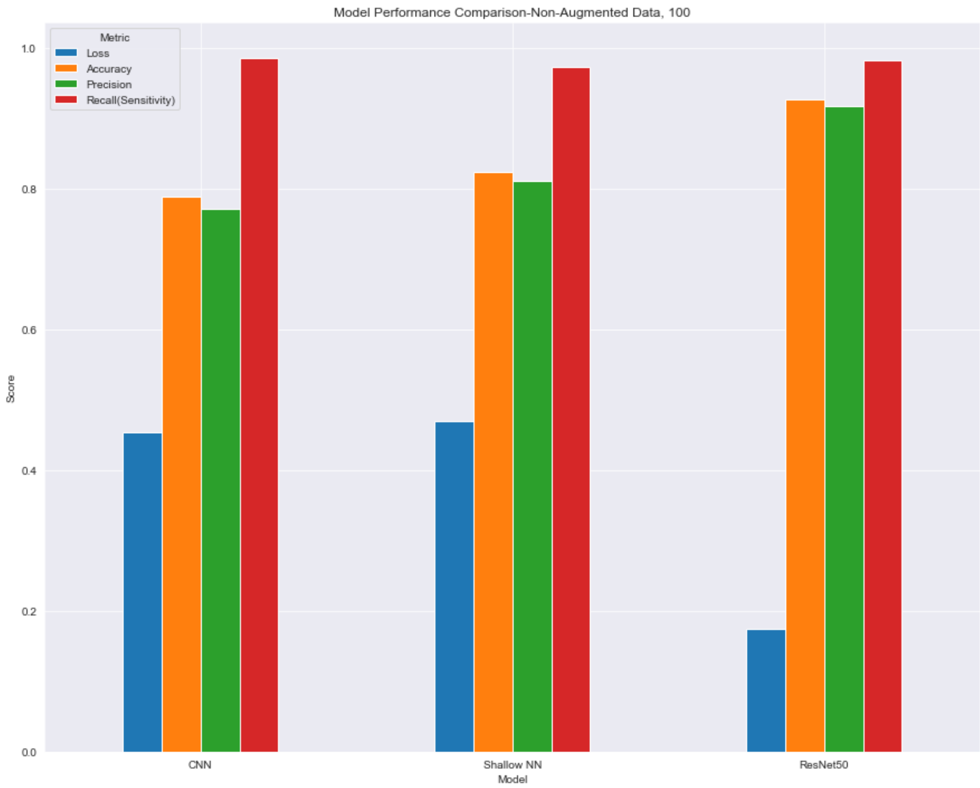


Fig. 3: Metrics of different classifiers for non-augmented, 100x magnified data. Blue bar shows the loss, yellow shows the accuracy, green shows the precision and red shows the recall.

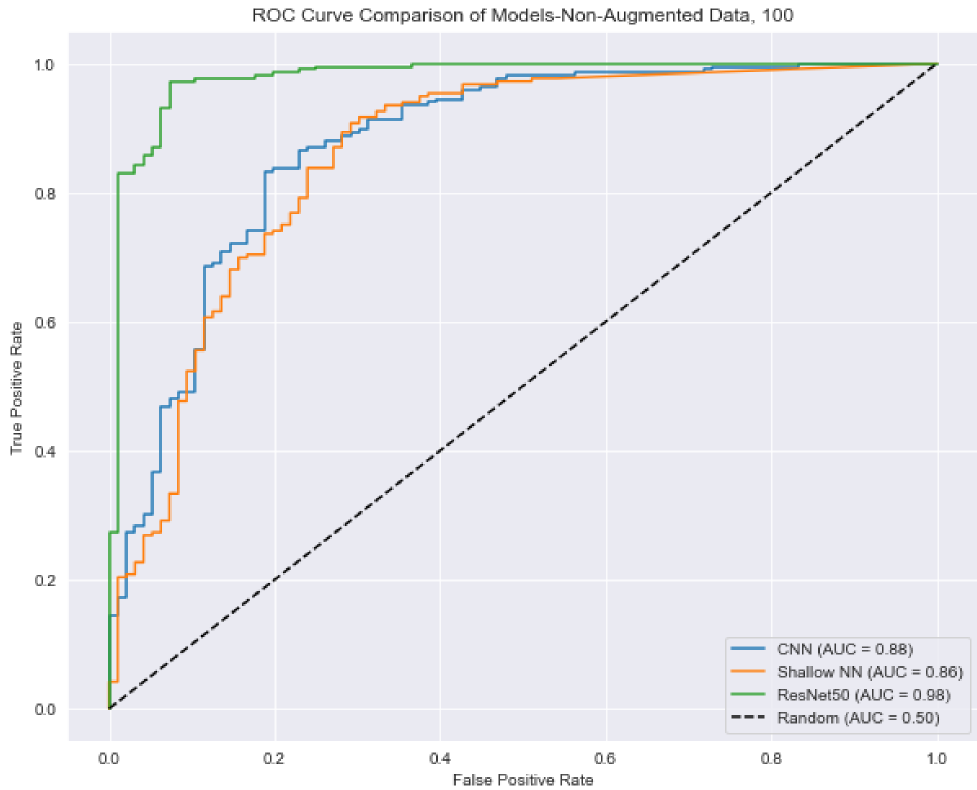


Fig. 4: ROC Curves for different classifiers. Green curve is for ResNet50, blue curve is for CNN and orange curve is for SNN. Non-augmented, 100x magnified data

Model	Loss	Accuracy	Precision	Recall (Sensitivity)
CNN	0.453616	0.788462	0.771739	0.986111
Shallow NN	0.469739	0.823718	0.810811	0.972222
ResNet50	0.174460	0.926282	0.917749	0.981481

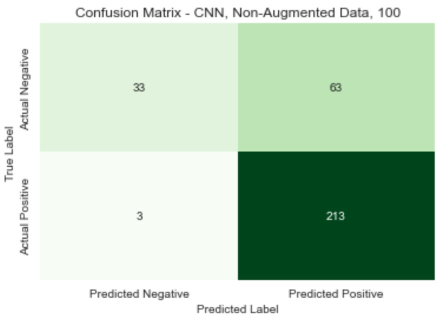
Table 2: Comparison of classifiers based on loss, accuracy, precision, and recall for **non-augmented data, 100x magnification**.

Model	Loss	Accuracy	Precision	Recall (Sensitivity)
CNN	0.458682	0.814103	0.898990	0.824074
Shallow NN	0.432735	0.836538	0.836735	0.949074
ResNet50	0.184443	0.919872	0.920705	0.967593

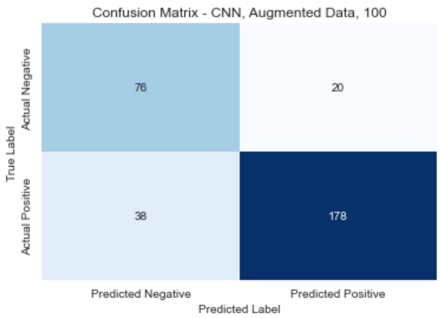
Table 3: Comparison of classifiers based on loss, accuracy, precision, and recall for **augmented data, 100x magnification**.

Comparing the confusion matrices for CNN (Figure 5), it is clear that both false and true negatives increase notably with the augmentation, while false positives decrease.

The changes may stem from augmented data introducing variations that help CNN and SNN generalize better, improving precision and accuracy, but also adding complexity that reduces their sensitivity (recall) to true positives. This trade-off may suggest a shift toward prioritizing precision over recall.



(a) Confusion matrix of CNN for **non-augmented**, 100x magnified data



(b) Confusion matrix of CNN for **augmented**, 100x magnified data

Fig. 5: Comparison of classifiers trained with non-augmented and augmented data.

6.3 Effect of Magnification

This section investigates whether the chosen magnification level has an effect on the training of our classifiers.

We compare Table 3 which is generated with 100x magnification data to Table 4 which reflects metrics for 400x magnification data. It can be observed that every metric of every model decreases slightly when using 400x magnified data, except recall of ResNet50 which increases from 0.97 to 0.98. These metrics indicate that performances of the models decrease slightly.

Highlighting this further, AUC for CNN decreases from 0.87 to 0.84, AUC for SNN decreases from 0.86 to 0.81 and AUC for ResNet50 drops from 0.98 to 0.96. (Figure 6)

The results indicate that increasing the magnification level from 100x to 400x has a slight negative impact on the performance of the classifiers. The reason could be that

Model	**Loss**	**Accuracy**	**Precision**	**Recall (Sensitivity)**
CNN	0.496596	0.783883	0.875000	0.794595
Shallow NN	0.505514	0.805861	0.826733	0.902703
ResNet50	0.278410	0.897436	0.882927	0.978378

Table 4: Comparison of different classifiers based on loss, accuracy, precision, and recall for **augmented data, 400x magnification**.

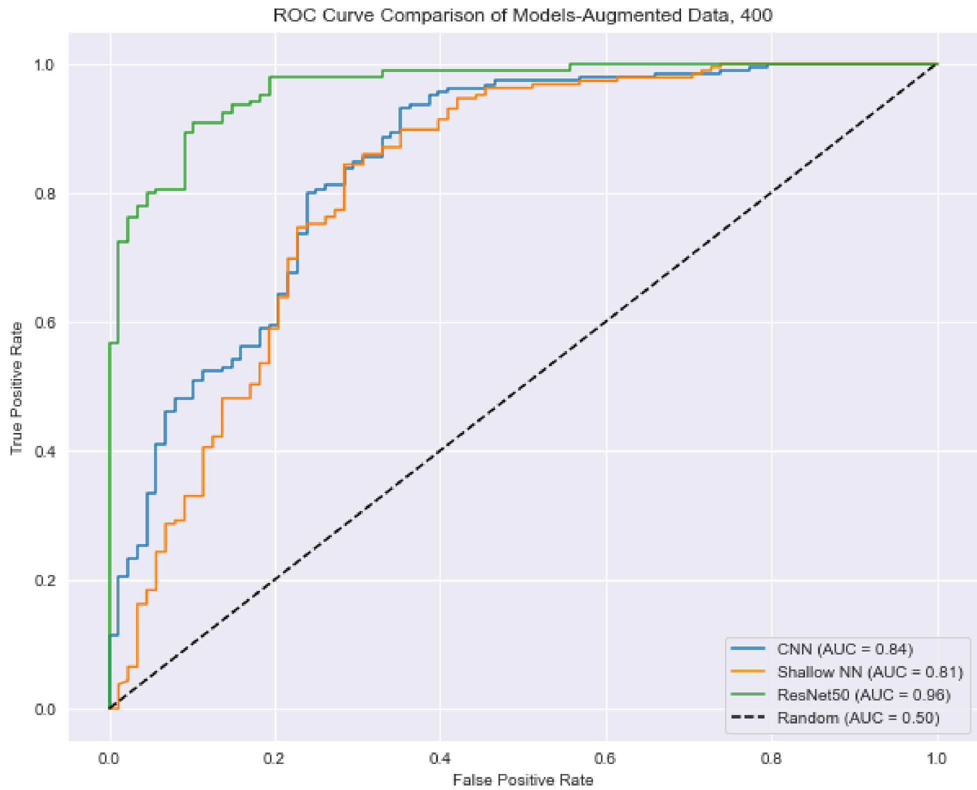


Fig. 6: ROC Curves for different classifiers. Green curve is for ResNet50, blue curve is for CNN and orange curve is for SNN. Augmented, 100x magnified data.

the frames captured with 400x magnification are too zoomed in, making it difficult to extract meaningful features. A higher magnification can also potentially increase noise.

7 Conclusion

In this study, we evaluated the performance of three deep learning classifiers—ResNet50, CNN, and Shallow Neural Network (SNN)—for classifying breast cancer histopathological images into benign and malignant categories using the BreakHis dataset. Among the models, ResNet50 performed the best, outperforming both CNN and SNN due to its superior feature extraction capabilities. Interestingly, images captured at 400x magnification yielded poorer classification results compared to those at 100x magnification, indicating that lower magnification images may retain features more relevant for effective classification. Additionally, data augmentation proved to be particularly beneficial for the CNN, significantly enhancing its performance. These findings highlight the importance of choosing appropriate magnifications and leveraging augmentation techniques for optimizing deep learning models in histopathological image analysis.

Supplementary information. The complete code for the study, including classifier training and plots can be found in the GitHub repository that can be accessed through this link: https://github.com/igalins/ifa.git

References

[1] Sun, Y.-S., Zhao, Z., Yang, Z.-N., Xu, F., Lu, H.-J., Zhu, Z.-Y., Shi, W., Jiang, J., Yao, P.-P., Zhu, H.-P.: Risk factors and preventions of breast cancer. International journal of biological sciences **13**(11), 1387 (2017)

[2] DeSantis, C.E., Ma, J., Gaudet, M.M., Newman, L.A., Miller, K.D., Goding Sauer, A., Jemal, A., Siegel, R.L.: Breast cancer statistics, 2019. CA: a cancer journal for clinicians **69**(6), 438–451 (2019)

[3] Pike, M.C., Spicer, D.V., Dahmoush, L., Press, M.F.: Estrogens, progestogens, normal breast cell proliferation, and breast cancer risk. Epidemiologic reviews **15**(1), 17–35 (1993)

[4] Ginsburg, O., Yip, C.-H., Brooks, A., Cabanes, A., Caleffi, M., Dunstan Yataco, J.A., Gyawali, B., McCormack, V., Anderson, M., Mehrotra, R., *et al.*: Breast cancer early detection: A phased approach to implementation. Cancer **126**, 2379–2393 (2020)

[5] Dar, R.A., Rasool, M., Assad, A., *et al.*: Breast cancer detection using deep learning: Datasets, methods, and challenges ahead. Computers in biology and medicine **149**, 106073 (2022)

[6] LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. nature **521**(7553), 436–444 (2015)

[7] Benhammou, Y., Achchab, B., Herrera, F., Tabik, S.: Breakhis based breast cancer automatic diagnosis using deep learning: Taxonomy, survey and insights. Neurocomputing **375**, 9–24 (2020)

[8] Spanhol, F.A., Oliveira, L.S., Petitjean, C., Heutte, L.: A dataset for breast cancer histopathological image classification. Ieee transactions on biomedical engineering **63**(7), 1455–1462 (2015)

[9] Shorten, C., Khoshgoftaar, T.M., Furht, B.: Text data augmentation for deep learning. Journal of big Data **8**(1), 101 (2021)

[10] Maharana, K., Mondal, S., Nemade, B.: A review: Data pre-processing and data augmentation techniques. Global Transitions Proceedings **3**(1), 91–99 (2022)

[11] Developers, S.-l.: Sklearn.model_selection.train_test_split — Scikit-learn 1.0.2 Documentation. https://scikit-learn.org/dev/modules/generated/sklearn.model_selection.train_test_split.html

[12] Tharsanee, e.a. R. M.: Deep convolutional neural network–based image classification for covid-19 diagnosis. Academic Press (2021)

[13] Prinzi F, G.S.V.S. Currieri T: Shallow and deep learning classifiers in medical image analysis. Eur Radiol Exp. (2024)

[14] Theckedath, S.R.R. D.: 2020. SN COMPUT. SCI. 1 (2020)

[15] Developers, P.: Pandas Documentation. https://pandas.pydata.org/docs/

[16] Foundation, P.S.: Os — Miscellaneous Operating System Interfaces. https://docs.python.org/3/library/os.html

[17] Developers, N.: Array Manipulation Routines — NumPy Documentation. https://numpy.org/doc/2.1/reference/routines.array-manipulation.html

[18] Developers: Sklearn.model_selection Module — Scikit-learn 1.5.0 Documentation. https://scikit-learn.org/1.5/api/sklearn.model_selection.html

[19] Developers: Sklearn.metrics Module — Scikit-learn 1.5.0 Documentation. https://scikit-learn.org/1.5/api/sklearn.metrics.html

[20] Developers, M.: Matplotlib.pyplot — Matplotlib 3.5.3 Documentation. https://matplotlib.org/3.5.3/api/_as_gen/matplotlib.pyplot.html

[21] Developers, S.: seaborn.objects.Plot — Seaborn Documentation. https://seaborn.pydata.org/generated/seaborn.objects.Plot.html

[22] Developers, T.: TensorFlow — All Symbols. https://www.tensorflow.org/api_docs/python/tf/all_symbols

[23] Developers, T.: TensorFlow Keras — All Symbols. https://www.tensorflow.org/api_docs/python/tf/keras

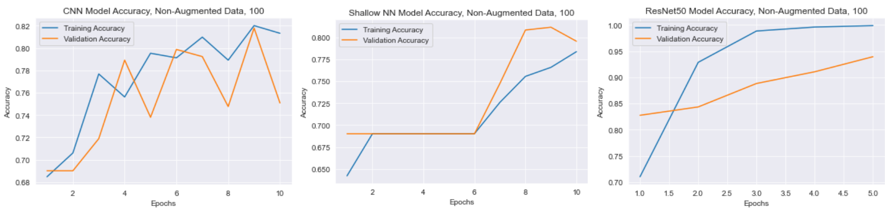
[24] Developers, T.: Training with Built-in Methods. https://www.tensorflow.org/guide/keras/training_with_built_in_methods

[25] Koonce, B.: ResNet 50, pp. 63–72. Apress, Berkeley, CA (2021). https://doi.org/10.1007/978-1-4842-6168-2_6 . https://doi.org/10.1007/978-1-4842-6168-2_6

Appendix A Supplementary Figures

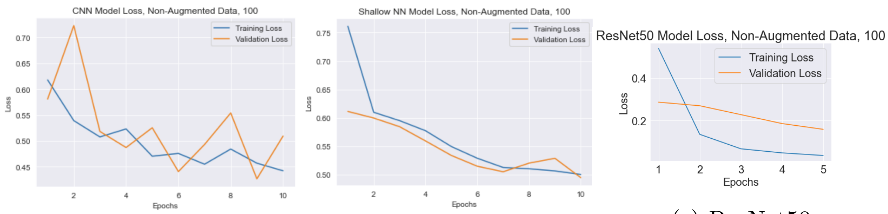
A.1 Learning Curves

A.1.1 Non Augmented Data, 100x Magnification



(a) CNN (b) SNN (c) ResNet50

Fig. A1: Accuracy learning curves for 3 Classifiers for non augmented data, 100x magnification



(a) CNN (b) SNN (c) ResNet50

Fig. A2: Loss learning curves for 3 Classifiers for non augmented data, 100x magnification

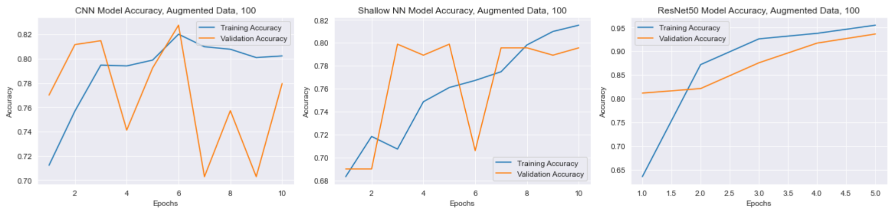
Figures A1 and A2 show the accuracy and loss learning curves for the non augmented data at 100x magnification.

A.1.2 Augmented Data, 100x Magnification

Figures A3 and A4 show the accuracy and loss learning curves for the augmented data at 100x magnification.

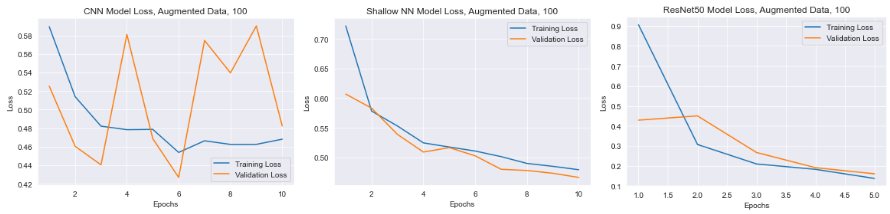
A.1.3 Augmented Data, 400x Magnification

Figures A5 and A6 show the accuracy and loss learning curves for the augmented data at 400x magnification.



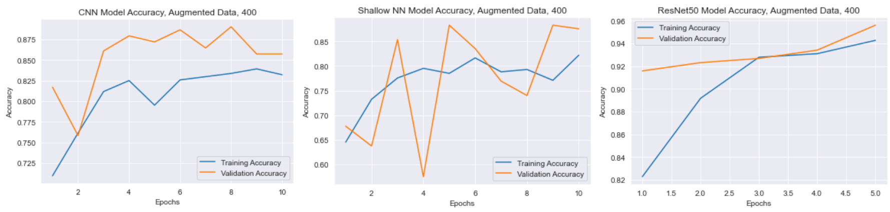
(a) CNN (b) SNN (c) ResNet50

Fig. A3: Accuracy learning curves for 3 Classifiers for augmented data, 100x magnification



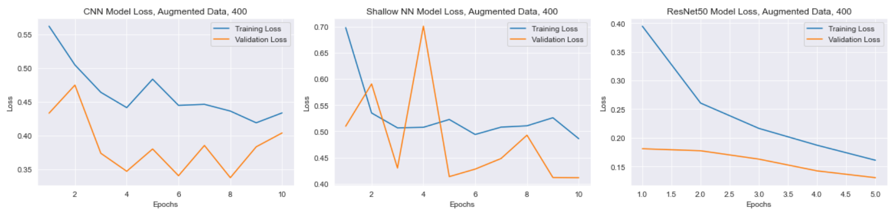
(a) CNN (b) SNN (c) ResNet50

Fig. A4: Loss learning curves for 3 Classifiers for augmented data, 100x magnification



(a) CNN (b) SNN (c) ResNet50

Fig. A5: Accuracy learning curves for 3 Classifiers for augmented data, 400x magnification



(a) CNN (b) SNN (c) ResNet50

Fig. A6: Loss learning curves for 3 Classifiers for augmented data, 400x magnification