Using Convolutional Neural Networks (CNN) to Detect Invasive Ductal Carcinoma (IDC)

Seyed Hosseini Darabi, Nathan Acosta, Mohammad Zadeh, Shaz Maknojia

Abstract

This study proposes a convolutional neural network (CNN) approach to invasive ductal carcinoma (IDC) detection and compares the effectiveness of various CNN architectures against different non-CNN machine learning algorithms. All the described architectures were guided using a large, public dataset of roughly 277,000, 50x50 pixel images. Validation tests are carried out for all quantitative results in accordance with the respective performance metrics for each method. We utilize Random Forest, Support Vector Machine (SVM), Decision Tree, and K-Nearest Neighbors (KNN), as well as four distinct CNN models. We find this proposed system to be successful, with an average CNN accuracy of 81% and potential to reduce human error in IDC diagnoses. Moreover, our proposed CNN approach also outperforms the non-CNN machine learning algorithms' average accuracy of 72%. We therefore find that our proposed CNN approach to IDC detection improves accuracy by 9 percentage points when compared to other machine learning approaches.

1. Introduction

Breast cancer remains the most prevalent form of cancer for women in the United States [1]. The most common type of breast cancer is invasive ductal carcinoma (IDC) IDC), which comprises approximately 80% of all female breast cancer diagnoses [2]. Moreover, not only is breast cancer more common among women over the age of 50, it is especially life-threatening when joined with a number of comorbidities [3]. An example of this is women who have endured lung cancer—for them, a breast cancer diagnosis is particularly more dangerous than for the average person who is diagnosed.

Consequently, the accurate identification and diagnosis of IDC is of huge importance within the medical field, and any approaches that can provide viable, high quality classification are crucial insofar as they can save time, reduce clinical error, and improve upon existing diagnostic infrastructure. Thus, exploring ways in which the detection of such a prevalent and potentially life-altering disease can be improved—in our case, through the incorporation of machine learning techniques like CNN—holds the potential for enormous societal benefit. A successful, effective model would be able to raise the standard for breast cancer detection and, in doing so, provide key medical and economic benefits.

The benefits of a successful, effective model are numerous. Perhaps most obviously, improvements in diagnostic models can

reduce the number of deaths due to breast cancer—which presently sits at more than 42,000 annually [1]. Moreover, more effective diagnostic approaches can lead to increased life expectancy and 5-year survival rates for women, improving women's quality of life and long term health. Another area of particular value in which our model—and other effective breast cancer classification models—can provide meaningful societal benefit is in the form of more effective early screenings. By improving the quality and consistency of diagnoses, early screenings would be more effective and yield a reduction in the number of women diagnosed with late-stage breast cancer [4]. Lastly, there is also an economic benefit attributable to improving breast cancer diagnoses. By improving diagnostic accuracy and ensuring more breast cancer patients are treated early, the total yearly cost of breast cancer treatment—which presently sits at $16.5 billion [5]—can be reduced without any sacrifice in quality.

Thus, with all of this taken into consideration, our goal in this project is to utilize machine learning algorithms and evaluate the respective accuracies of different models with a focus on breast cancer—specifically, IDC—detection. Using a large public dataset of over 277,000 breast histopathology images, we implement 8 distinct models and evaluate their results. These models consist of Random Forest, Support Vector Machine (SVM), Decision Tree, and K-Nearest Neighbors (KNN), and CNN approaches. For our CNN approach, we utilize 4 different models—three of which are pre-trained, with one being trained from scratch. We divide our experiment into two parts: in the first half, we obtain the model accuracies of each approach; then, to evaluate whether our CNN models are truly more effective than our non-CNN models, we compare the respective accuracies of both.

2. Related Work

Given the prevalence of machine learning applications in the medical field, there is a wealth of relevant literature from which we were able to draw guidance for this project. For example, a 2021 study in the *Journal of Healthcare Engineering* examines the difference between CNN and non-CNN approaches to breast cancer detection [6]. While this research utilizes different non-CNN models and CNN architectures than our project, the results are surprisingly similar insofar as it also finds a roughly 9 percentage point difference between CNN and non-CNN average model accuracies [6].

Similarly, other research such as a 2018 study presented at the *International Conference on Computational Science and Computational Intelligence* (CSCI) solely focuses on different CNN architectures without any comparison to non-CNN approaches [7]. The highest CNN model accuracy obtained in this study is slightly higher than ours (89% vs. our ~87%) and utilizes a dataset that is approximately the same size as ours [7].

There are also certain studies we looked at that displayed potential warning signs. A 2019 study in the open-access *Informatics in Medicine Unlocked* journal reported a

staggering 99.86% CNN test accuracy, which we interpreted as a possible indication of overfitting [8]. Learning from these existing studies and the broader CNN literature, we were able to apply specific best practices and precautions to our models.

3. Proposed Method

In this project, we implement 8 models: 4 non-CNN algorithms and 4 distinct CNN architectures. For our non-CNN machine learning algorithms, we utilize Random Forest, SVM, Decision Tree, and KNN. Our CNN architectures—which we use to compare the effectiveness of our overall CNN approach to IDC classification against its non-CNN counterparts—include ResNet_50V2, Inception_V3, and EfficientNet_B0, as well as a simple sequential CNN model. The simple sequential CNN, as well as all non-CNN models, is trained solely on our breast histopathology dataset, while the other 3 CNN approaches are imported as keras pre-trained models. We use model accuracy—which can simply be defined as the number of correct predictions divided by the number of total predictions—as our performance metric for all 8 models.

3.1 Random Forest

Random Forests, also known as Random Decision Forests, are non-parametric ensemble learning methods that can be applied to classification, regression, and other tasks through the application of various Decision Trees. In the case of classification, each of these individual Decision Trees (which together comprise the 'forest') generate a class prediction, and the class with the most votes becomes the Random Forest model's overall prediction. For the purposes of our project, we utilize a Random Forest model to conduct classification with respect to breast histopathology images. Our Random Forest model classifies such images as either negative (no IDC detected) or positive (IDC diagnosed).

3.2 Support Vector Machine (SVM)

Support Vector Machines, or SVM, are supervised learning methods that can be applied to both classification and regression situations. The benefits of SVM include its versatility (via different Kernel functions that can be customized to a specific decision function) and memory efficiency. SVM is capable of conducting multi-class classification, but for the purposes of our project we are using it for binary classification between IDC and non-IDC images. Like our other non-CNN machine learning approaches (and simple sequential CNN), this model is trained using the full histopathology dataset.

3.3 K-Nearest Neighbor (KNN)

K-Nearest Neighbor is a lazy learning, non-parametric algorithm. The algorithm will find the K "nearest neighbors" of a given data point, and then it will assign the given point a class depending on the predominant plurality of surrounding point classes. For distance metrics, we will use the Euclidean metric, and finally, the input x gets assigned to the class with the largest probability. For the purposes of our project, we train this model using the breast histopathology dataset and use it to classify

images as either negative (no IDC) or positive (IDC).

3.4 Decision Tree

We utilize a Decision Tree model because it can be rewritten as a set of discrete rules to make it easier to understand. The main advantage of the Decision Tree classifier is its ability to use different feature subsets and decision rules at different stages of classification. In the Decision Tree, we calculate the entropy of the target, then the dataset is split into different attributes. The entropy for each branch is calculated and is added proportionally to get the total entropy for the split. The resulting entropy is subtracted from the entropy before the split. The result is the information gain or decrease in entropy. Then we choose the attribute with the largest information gain as the decision node, divide the dataset by its branches, and repeat the same process on every branch. Finally, if a branch with an entropy of 0 is a leaf node, otherwise a branch with an entropy more than 0 needs further splitting. As with our other non-CNN models, we train this model using our breast histopathology dataset.

3.5 Convolutional Neural Network (CNN)

Convolutional Neural Networks (CNN) are a specialized set of artificial neural networks which are adept at analyzing visual data. In place of normal matrix multiplication, CNN's utilize what is known as a convolution operation. Convolution operations aid the network in identifying feature maps embedded within image data. Our approach consists of testing four separate CNN architectures, three of which—EfficientNet_B0, ResNet_50v2, and Inception_V3—are keras pre-trained networks, with the fourth being constructed and implemented via the keras package in python and trained from scratch on the histopathology dataset.

EfficientNet

Stage i	Operator $\hat{\mathcal{F}}_i$	Resolution $\hat{H}_i \times \hat{W}_i$	#Channels $\hat{C}_i$	#Layers $\hat{L}_i$
1	Conv3x3	224 × 224	32	1
2	MBConv1, k3x3	112 × 112	16	1
3	MBConv6, k3x3	112 × 112	24	2
4	MBConv6, k5x5	56 × 56	40	2
5	MBConv6, k3x3	28 × 28	80	3
6	MBConv6, k5x5	14 × 14	112	3
7	MBConv6, k5x5	14 × 14	192	4
8	MBConv6, k3x3	7 × 7	320	1
9	Conv1x1 & Pooling & FC	7 × 7	1280	1

ResNet50v2

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$
conv3_x	28×28	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times8$
conv4_x	14×14	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times23$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times36$
conv5_x	7×7	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

InceptionV3

Type	Kernel size/stride	Input size
Convolution	3 × 3/2	299 × 299 × 3
Convolution	3 × 3/1	149 × 149 × 32
Convolution	3 × 3/1	147 × 147 × 32
Pooling	3 × 3/2	147 × 147 × 64
Convolution	3 × 3/1	73 × 73 × 64
Convolution	3 × 3/2	71 × 71 × 80
Convolution	3 × 3/1	35 × 35 × 192
Inception module	Three modules	35 × 35 × 288
Inception module	Five modules	17 × 17 × 768
Inception module	Two modules	8 × 8 × 1,280
Pooling	8×8	8 × 8 × 2,048
Linear	Logits	1 × 1 × 2,048
Softmax	Output	1 × 1 × 1,000

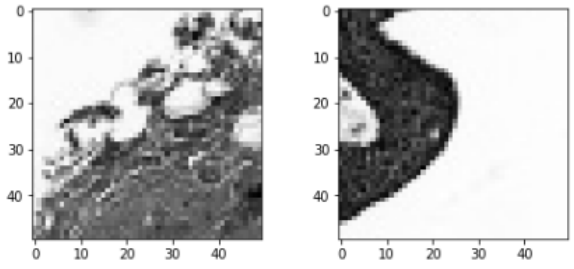
Keras Sequential CNN

Layer	Output Dimension	Kernel Size / Stride	Activation	Output Shape
Convolution	16	3x3 / 1	ReLU	48x48x16
Max Pooling	-	2x2 / 2	-	24x24x16
Convolution	32	3x3 / 1	ReLU	22x22x32
Max Pooling	-	2x2 / 2	-	11x11x32
Convolution	16	3x3 / 1	ReLU	9x9x16
Max Pooling	-	2x2 / 2	-	4x4x16
Flatten	-	-	-	256
Dense	50	-	ReLU	50
Dense	1	-	Sigmoid	1

4. Experiments

4.1 Dataset

The dataset we used consisted of 277,524 PNG files of pixel dimension 50x50. Of the 277,524 images, 198,738 represented test negative (absence of cancer), and the remaining 78,786 images represented test positive. Image preprocessing methods consisted of Principal Component Analysis (PCA) and rescaling color channel values to the interval [0, 1]. Additionally, image pixel dimensions were rescaled in accordance with the input dimensionality requirements for certain pre-trained CNN models.



(1) (0)

Preprocessed Images

4.2 Random Forest

The Random Forest model is trained using the entirety of our breast histopathology image dataset and is imported from Python's sklearn library (as RandomForestClassifier). The number of Decision Trees used in this forest, or n_estimators, is set to 200. Cross validation (via the *cross_val_score* method imported from sklearn.model_selection) is used to estimate the model's predictive accuracy with respect to unseen data and thereby protect against overfitting.

4.3 Support Vector Machine (SVM)

The SVM model is using the entirety of our breast histopathology image dataset and is imported from Python's sklearn library (as SVC, given this is a classification task). With respect to parameters, we set gamma (our hyperparameter for determining how much curvature we want in our decision boundary) to 'auto,' which uses 1/(number of features). Moreover, we use the radial basis function, or RBF, and linear for our kernels. The model's accuracy is assessed via a validation set approach and the ultimate accuracy is obtained via the *accuracy_score* method from sklearn.metrics.

4.4 K-Nearest Neighbor (KNN)

The KNN model was created using the preprocessed images. 70% of the images were used as a training set, and the remaining 30% as a testing set. The KNN algorithm utilized, *KNeighborsClassifier*, was imported from sklearn.neighbors. The classification algorithm operated with the n_neighbors parameter set to 3.

4.5 Decision Tree

The Decision Tree classification model utilized the entire preprocessed dataset.

Implementation of the model was achieved through the sklearn package in python. The gini-index criterion was used to prune the tree and the max_depth was set to 3 with a random_state of 0. The model's accuracy was assessed via a validation set approach and the final accuracy score of the model was obtained via *accuracy_score*, imported from sklearn.metrics.

4.5 Convolutional Neural Network (CNN)

Three of the CNN models used [EfficientNet_B0, ResNet_50v2, Inception_V3] are Keras pre-trained models. The specific architecture of these models is displayed in table format under Section 3, Methods. The fourth CNN model was a straightforward sequential CNN constructed using Keras layers via Python. All models were trained on local machines. A batch size of 32 was implemented to ease memory costs. Pre-trained CNN models underwent 10 to 15 training iterations, while the sequential CNN underwent 40 training iterations (variance in the number of iterations was due primarily to computational resources and time). Test accuracy was predicted via a validation set approach, wherein 70% of the dataset was used for training, and the remaining 30% was used for testing.

5. Results

The Simple Sequential CNN model, which had 9 layers, gave us our highest accuracy of 86.96%. The next highest accuracy model was our Inception_V3 model, which used 42 layers and achieved an accuracy of 83.37%. The ResNet_50V2 model, which has 50 layers, yielded an accuracy of 80.04%. With our top three highest accuracies being CNN models, the next most accurate model is our Random Forest model—which utilized 200 trees (i.e., n_estimators = 200) and achieved an accuracy of 76.21%. Furthermore, our SVM model achieved a result of 76.15%, while the EfficientNet_B0 CNN model—which had 237 layers—achieved an accuracy of just 72.70%. Finally, the two lowest model accuracies belong to our Decision Tree, with 72.30%, and KNN, with 62.40%. A tabulation of these results (sorted by descending accuracy) can be seen below:

Model	Accuracy ↓	Brief Description
Simple Sequential CNN	**86.96%**	9 Layer Convolutional Neural Network (Implemented using tensorflow.keras)
Inception_V3 Pretrained	**83.37%**	42 Layer Convolutional Neural Network Pretrained on Imagenet dataset
ResNet_50V2 Pretrained	**80.04%**	50 Layer Residual Neural Network Pretrained on Imagenet dataset
Random Forest	**76.21%**	Non-Parametric Ensemble Learning Classifier
SVM	**76.15%**	Deep Learning Supervised Learning Classifier
EfficientNet_B0 (Pretrained)	**72.70%**	237 Layer Convolutional Neural Network Pretrained on Imagenet dataset
Decision Tree	**72.30%**	Non-Parametric Supervised Learning Classifier
KNN	**62.40%**	Non-Parametric Supervised Learning Classifier

5.1 Random Forest

The Random Forest model achieved a test accuracy score of 76.21%.

5.2 Support Vector Machine (SVM)

The SVM model achieved a test accuracy score of 76.15%.

5.3 K-Nearest Neighbor (KNN)

The KNN model achieved a test accuracy score of 62.40%.

5.4 Decision Tree

The Decision Tree model achieved a test accuracy score of 72.30%.

5.5 Convolutional Neural Network (CNN)

Three out of the four CNN models—specifically, ResNet_50v2, Inception_V3, and the Keras Sequential CNN—performed better than all non-CNN models, with an average test accuracy of 83.46%. Given the intended function of CNN's with respect to computer vision and image classification, the results line up with initial expectations. The EfficientNet_B0 model did not perform as well as the other CNN models, but still outperformed half of the non-CNN models. The table below outlines the details of each CNN model:

Model	Test Accuracy
Sequential	86.9%
InceptionV3	83.37%
ResNet50v2	80.04%
EfficientNetB0	72.70%

5.6 Discussion

Our KNN, Decision Tree, and EfficientNet_B0 models all yielded accuracies below 75%. While this level of accuracy is not ideal, it is not within the realm of being dismissible. The remaining models all exceeded 75% test accuracy scores, with the top three models consisting of two pre-trained CNN models (ResNet_50v2 and Inception_V3), as well as the Simple Sequential CNN model. Interestingly, our best performing model was the Simple Sequential CNN, with a test accuracy of 86.9%. We believe this high level of accuracy clearly showcases the predictive power of CNN models with respect to image classification.

While we are not fully sure of why the pre-trained models were outperformed by the more rudimentary sequential CNN model, we believe it to be at least partly attributable to the fact that the pre-trained models' convolutional layers were attuned to the Imagenet dataset as opposed to the dataset we were analyzing. In the future, given better hardware and more time, we believe a more thorough investigation regarding the discrepancy between the pre-trained models' accuracy as compared to the sequential CNN could be better outlined.

6. Conclusion

The best model we tested was the simple sequential CNN model. It had the highest accuracy, decisively beating out the pre-trained models while also demanding less in terms of computational cost as compared to the other CNN models. Although we got good results with our models, we had a few limitations. One of these is merely the fact that our dataset was quite large, containing over 277,000 images in total. Another noticeable limitation is that most of the images in our dataset were not especially clear, meaning that our models ran the risk of potentially misclassifications as a consequence of naturally difficult-to-distinguish dataset images. Going forward, we would utilize better hardware to enable improved testing and training, which would thereby allow us to employ sophisticated models more easily. We would likewise use other pre-trained models that would be better suited for our

dataset and seek out better quality images for our datasets, thereby making our models' jobs easier when it comes to classification.

7. Contribution

Shaz Maknojia obtained the original dataset from Kaggle. Mohammad Mahmoudzadeh handled the majority of the data preprocessing code and constructed the Decision Tree and KNN models, while Nathan Acosta contributed the code for image scaling and ResNet_50v2, Inception_V3, and Simple Sequential CNN models. Shaz also wrote the code for the EfficientNet model, while Seyed Hosseini Darabi programmed the Random Forest and SVM models. Regarding the written components, Seyed wrote the abstract and original project proposal, as well as the introduction and related works sections in the final report; Seyed also wrote the methods, experimentation, and results sections for his Random Forest and SVM models. Mohammad wrote the methods, experimentation, and results for his KNN and Decision Tree models. Nathan described the dataset and results sections, as well as the methods, experimentation, and results sections for his CNN models. Finally, Shaz completed the discussion, conclusion and contribution sections.

References

[1] U.S. Cancer Statistics Working Group. U.S. Cancer Statistics Data Visualizations Tool, based on 2020 submission data (1999-2018): U.S. Department of Health and Human Services, Centers for Disease Control and Prevention and National Cancer Institute; www.cdc.gov/cancer/dataviz, released in June 2021.

[2] "Invasive Ductal Carcinoma (IDC)." Invasive Ductal Carcinoma (IDC), https://www.breastcancer.org/types/invasive-ductal-carcinoma.

[3] Derks, Marloes GM, et al. "Impact of Comorbidities and Age on Cause‐Specific Mortality in Postmenopausal Patients with Breast Cancer." *The Oncologist* 24.7 (2019): e467-e474.

[4] Gangnon RE, Sprague BL, Stout NK, et al. The contribution of mammography screening to breast cancer incidence trends in the United States: an updated age-period-cohort model. *Cancer Epidemiol Biomarkers Prev.* 2015;24(6):905–912.

[5] Mariotto AB, Yabroff KR, Shao Y, Feuer EJ, Brown ML. Projections of the cost of cancer care in the United States: 2010–2020. *J Natl Cancer Inst.* 2011; 103(2):117–128.

[6] Alanazi, Saad Awadh, et al. "Boosting breast cancer detection using convolutional neural network." *Journal of Healthcare Engineering* 2021 (2021).

[7] Wang, Justin L., et al. "A study on automatic detection of IDC breast cancer with convolutional neural networks." *2018 International Conference on Computational Science and Computational Intelligence (CSCI)*. IEEE, 2018.

[8] Dabeer, Sumaiya, Maha Mohammed Khan, and Saiful Islam. "Cancer diagnosis in histopathological image: CNN based approach." *Informatics in Medicine Unlocked* 16 (2019): 100231.

[9] Our Compiled Python Code