Bioinformatics
doi.10.1093/bioinformatics/xxxxxx
Advance Access Publication Date: Day Month Year
Manuscript Category

BMI776 Project Report

Identification of Invasive Ductal Carcinoma in Breast Cancer

Krittisak Chaiyakul [1]

[1] Computer Science, University of Wisconsin-Madison, Madison, 53706, USA

Abstract

Invasive Ductal Carcinoma (IDC) is a very common subtype of breast cancers. In addition, deep learning becomes pretty common and powerful for image classification. There were two previous researches on detecting IDC (Cruz-Roa, A. et al. 2014) and (Janowczyk, A. and Madabhushi) that provided good baselines for applying deep learning on whole slide images. Now, I apply three different models (VGG, ResNet, and Inception) on the dataset. The dataset is splitted into training, validation, and testing set. The models are trained on training set, and the best models are chosen based on validation F1-score. The evaluation of VGG provides 0.7585 on F1-score and 0.8317 on Balanced accuracy score.The evaluation of ResNet provides 0.7811 on F1-score and 0.8419 on Balanced accuracy score. The evaluation of Inception provides 0.7896 on F1-score and 0.8515 on Balanced accuracy score. ResNet and Inception provide slightly better results than the ones in previous researches. However, Inception seems to provide a better overall prediction.

1 Introduction

Invasive Ductal Carcinoma (IDC) is a very common subtype of breast cancers. If we can automatically predict whether the cancer exists in the images accurately, we can reduce the amount of work doctors need to do and sometimes identify the cancer that the doctors miss. Moreover, we can adapt the idea to other related works in this field.

Deep learning is very common nowadays for classification. Especially, many different types of deep learning have been introduced to deal with image classification. There were two previous researche on detecting IDC. Cruz-Roaa, A. et al. (Cruz-Roa, A. et al. 2014) applied a simple 3-layer convolutional neural network architecture. This model got 0.718 (F-score) and 0.8423 (Balance accuracy). Janowczyk. A. (Janowczyk, A. and Madabhushi) applied a small version of AlexNet along with multiple preprocessing the inputs, such as cropping, resizing, and additional rotation. This model got about 0.75 (F-score) and 0.84 (Balance accuracy) on average.

I plan to introduce and test the performance on tree different models: VGG (Simonyan, K. and Zisserman), ResNet (He, K. et al. 2015), and Inception (Szegedy, C. et al. 2015). These models are more complex than ones previously testing on this dataset, so they can learn more complicated features and have higher accuracy.

This paper is followed by Methods, Results, and Discussion sections.

2 Methods

2.1 Dataset

The dataset was obtained from Kaggle Site. It consists of 279 whole mount slide images of Breast Cancer (BCa) specimens scanned at 40x. From that, 277,524 patches of size 50 x 50 were extracted (198,738 IDC negative and 78,786 IDC positive).

The dataset was splitted into three subsets of 167 training, 57 validation, and 55 testing dataset. Figure 1 shows an overview of dataset. The images are normalized between 0 and 1. The labels are encoded with one-hot head encoding into 2 nodes.

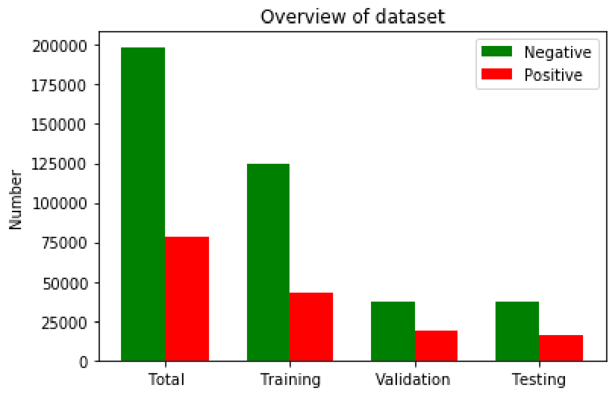

Fig. 1. The total dataset consists of 279 specimens with 198,738 IDC negative and 78,786 IDC positive. The training dataset consists of 167,382 images with 124,190 IDC negative and 43,192 IDC positive. The validation dataset consists of 56,331 images with 37,141 IDC negative and 19,190 IDC positive. The testing dataset consists of 53,811 images with 37,407 IDC negative and 16,404 IDC positive.

2.2 VGG

I used a simple architecture based on VGG (Simonyan, K. and Zisserman). Figure 2 shows an overview of the architecture. Table 2 shows a configuration of the model.

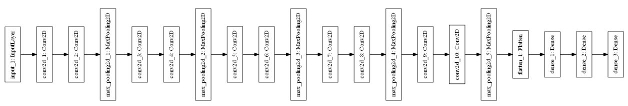

Fig. 2. VGG architecture.

Table 1. Sublayers of VGG configuration with "same" padding on every convolutional and pooling layers and "ReLu" activation on every layer except the last layer with "SoftMax" activation. * is determined by Num Kernels in the full model.

Layer	Type	Num Kernels	Kernel size	Stride
0	Convolution	*	3x3	1
1	Convolution	*	3x3	1
2	Max pool	-	2x2	2

Table 2. VGG configuration with "same" padding on every convolutional and pooling layers and "ReLu" activation on every layer except the last layer with "SoftMax" activation.

Layer	Type	Num Kernels	Kernel size	Stride
0	Input	3	50x50	-
1	Table 1	16	-	-
2	Table 1	32	-	-
3	Table 1	64	-	-
4	Table 1	128	-	-
5	Table 1	256	-	-
6	Fully connected	256	-	-
7	Fully connected	256	-	-
8	Fully connected	2	-	-

2.3 ResNet

I used a simple architecture based on ResNet (He, K. et al. 2015). Figure 3 shows an overview of the architecture. Table 4 shows a configuration of the model.

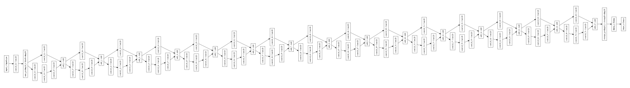

Fig. 3. ResNet architecture.

Table 3. Sublayers of ResNet configuration with "same" padding on every convolutional and pooling layers and "ReLu" activation on every layer except the last layer with "SoftMax" activation. * is determined by Num Kernels in the full model. ** is determined by Stride in the full model

Layer	Type	Num Kernels	Kernel size	Stride	Connect from
0	Input	-	-	-	-
1-1	Convolution	*	3x3	**	0
1-2	Convolution	*	3x3	1	1-1
1-3	Convolution	*	3x3	1	1-2
1-4	Convolution	*	3x3	1	1-3
1-5	Convolution	*	3x3	1	1-4
1-6	Convolution	*	3x3	1	1-5
1-7	Convolution	*	3x3	1	1-6
1-8	Convolution	*	3x3	1	1-7
1-9	Convolution	*	3x3	1	1-8
1-10	Convolution	*	3x3	1	1-9
1-11	Convolution	*	3x3	1	1-10
1-12	Convolution	*	3x3	1	1-11
1-13	Convolution	*	3x3	1	1-12
1-14	Convolution	*	3x3	1	1-13
1-15	Convolution	*	3x3	1	1-14
2-1	Convolution	*	1x1	**	0
3	Add	*	*	*	1 and 2

Table 4. ResNet configuration with "same" padding on every convolutional and pooling layers and "ReLu" activation on every layer except the last layer with "SoftMax" activation.

Layer	Type	Num Kernels	Kernel size	Stride
0	Input	3	50x50	-
1	Convolution	16	5x5	2
2	Max pool	-	2x2	2
3	Table 3	16	-	1
4	Table 3	32	-	2
5	Table 3	64	-	2
6	Table 3	128	-	2
7	Table 3	256	-	2
8	Mean pool	-	full size	full size
9	Fully connected	2	-	-

2.4 Inception

I used a simple architecture based on Inception (Szegedy, C. et al. 2015). Figure 4 shows an overview of the architecture. Table 6 shows a configuration of the model.

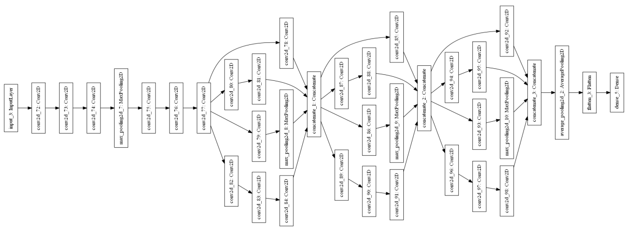

Fig. 4. Inception architecture.

Table 5. Sublayers of Inception configuration with "same" padding on every convolutional and pooling layers and "ReLu" activation on every layer except the last layer with "SoftMax" activation. * is determined by Num Kernels in the full model.

Layer	Type	Num Kernels	Kernel size	Stride	Connect from
0	Input	-	-	-	-
1-1	Convolution	*	1x1	2	0
2-1	Convolution	*	1x1	1	0
2-2	Max pool	-	2x2	2	2-1
3-1	Convolution	*	3x3	2	0
3-2	Convolution	*	1x1	1	3-1
4-1	Convolution	*	3x3	2	0
4-2	Convolution	*	3x3	1	4-1
4-3	Convolution	*	1x1	1	4-2
5	Concatenate	*	*	*	1, 2, 3 and 4

Table 6. Inception configuration with "same" padding on every convolutional and pooling layers and "ReLu" activation on every layer except the last layer with "SoftMax" activation.

Layer	Type	Num Kernels	Kernel size	Stride
0	Input	3	50x50	-
1	Convolution	32	3x3	2
2	Convolution	32	3x3	1
3	Convolution	32	3x3	1
4	Max pool	-	2x2	2
5	Convolution	64	3x3	1
6	Convolution	64	3x3	2
7	Convolution	64	3x3	1
8	Table 5	64	-	-
9	Table 5	128	-	-
10	Table 5	256	-	-
11	Mean pool	-	full size	full size
12	Fully connected	2	-	-

2.5 Hyperparameter Setting

Table 7 shows the settings for three deep learning models.

2.6 Evaluation

For each model type, I train the model on training set and choose the best model baed on F1-score on validation set.

Table 7. Deep learning hyperparameter settings for all models

Variable	Setting
Batch size	128
Initial learning rate	0.001
Learning rate schedule	Adam
Beta 1	0.9
Beta 2	0.999
Epsilon	None
Weight decay	0.0
AMSGrad	False
Number of iterations	30

Define F1-score and Balanced Accuracy.

$$F1 - score = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} \quad (1)$$

$$BalancedAccuracy = \frac{Sensitivity + Specificity}{2} \quad (2)$$

I test the performance of the model on F1 and BAC to be able to compare with the two previous researches.

Moreover, three specimens from testing set will be randomly selected to plot the prediction from the models on the whole sliding images.

3 Results

Table 8 shows f1-score and balanced accuracy across all three models as well as the models from previous researches.

Table 8. F1-score and Balance accuracy for VGG, ResNet, and Inception models. * is from (Janowczyk, A. and Madabhushi). ** is from (Cruz-Roa, A. et al. 2014)

Method	F1-score	Balance accuracy
VGG	0.7585	0.8317
ResNet	0.7811	0.8419
Inception	0.7896	0.8515
Resize*	0.7648	0.8468
Resize + Dropout*	0.7570	0.8423
Cropping*	0.7533	0.8415
Cropping + Additional Rotations*	0.7558	0.8368
**	0.7180	0.8423

Table 9 shows the confusion matrix of the results predicted by VGG.

Table 9. Confusion matrix on VGG

	Predict IDC (-)	Predict IDC (+)
IDC (-)	32473	4934
IDC (+)	3347	13006

Table 10 shows the confusion matrix of the results predicted by ResNet.

Table 10. Confusion matrix on ResNet

	Predict IDC (-)	Predict IDC (+)
IDC (-)	33953	3454
IDC (+)	3660	12693

Table 11 shows the confusion matrix of the results predicted by Inception.

Table 11. Confusion matrix on Inception

	Predict IDC (-)	**Predict IDC (+)**
IDC (-)	33568	3839
IDC (+)	3180	13173

Figure 5 shows the heat map probability predicted by three models on the specimen with id 12751.

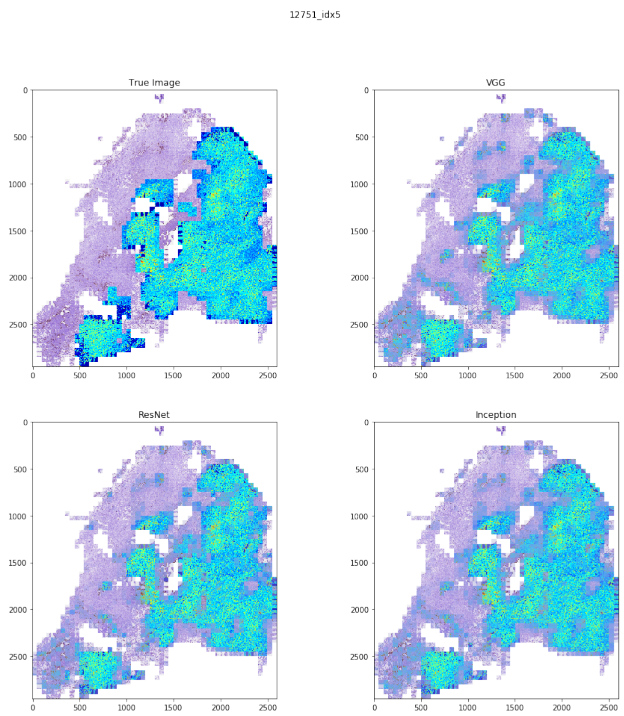

Fig. 5. Heat map of specimen with id 12751 on true positive, VGG, ResNet, and Inception

Figure 6 shows the heat map probability predicted by three models on the specimen with id 10303.

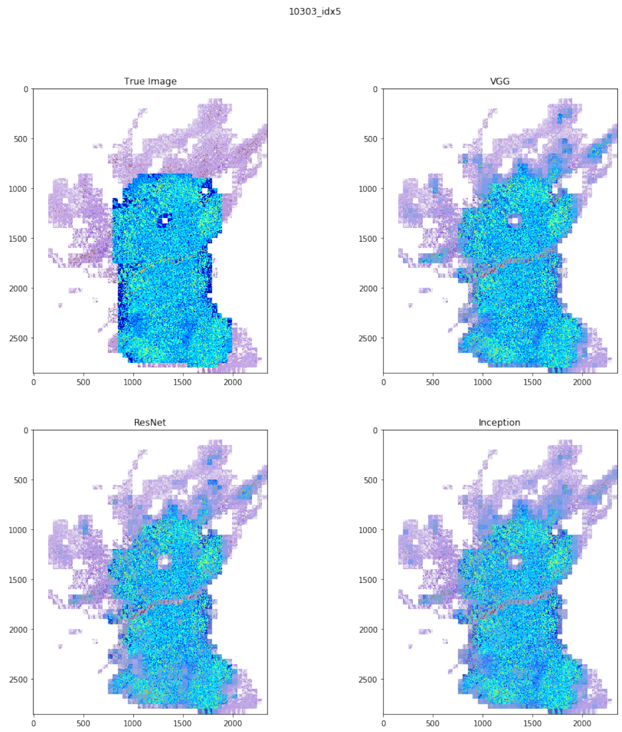

Fig. 6. Heat map of specimen with id 10303 on true positive, VGG, ResNet, and Inception

Figure 7 shows the heat map probability predicted by three models on the specimen with id 13462.

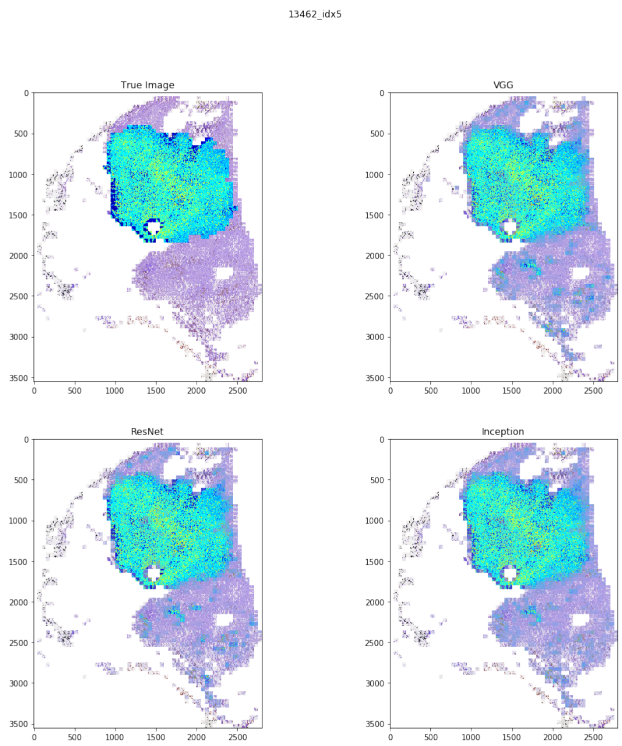

Fig. 7. Heat map of specimen with id 13462 on true positive, VGG, ResNet, and Inception

4 Discussion

Since the current data have more specimens than the ones in two previous paper, it is hard to compare them directly. VGG provides a similar result or slightly less accuracy than the result from this paper (Janowczyk, A. and Madabhushi) due to increase in the number of layers causing overfitting. However, ResNet and Inception models provide a quite better result than other models. These are the cases because those two models are consist of multiple paths of layers for allowing the model to learn different paths to fit the data, reducing the overfit of the model. In term of confusion matrix, based on VGG, ResNet tends to predict toward negative than positive. However, Inception seems to predict both negative and positive more equally correctly than previous two models. Therefore, if we want the model to prevent false positive, ResNet will reduce the noise on the whole image, as shown in Figure 5, 6, and 7, so that we can focus on the large group of positive IDC and know the general location. In contrast, if we want the model to provide more predicted positive, Inception will predict more positive on 50x50 images that might have IDC, as shown in Figure 5, 6, and 7. Therefore, Inception will focus on all potential IDC so that we can identify each small group of positive IDC. In addition, Inception is better in classification for single 50x50 image.

For future work, it will be great to try incorporate the nearby images along with the 50x50 image into consideration on training and predicting the outputs. This might be helpful to identify the location of IDC in the whole slide images.

References

Cruz-Roa, A., Basavanhally, A., GonzÃ¡lez, F., Gilmore, H., Feldman, M., Ganesan, S., Shih, N., Tomaszewski, J. and Madabhushi, A. 2014. Automatic detection of invasive ductal carcinoma in whole slide images with convolutional neural networks. In: *Medical Imaging 2014: Digital Pathology*. doi: 10.1117/12.2043872.

He, K., Zhang, X., Ren, S. and Sun, J. 2015. Deep Residual Learning for Image Recognition. In: ArXiv e-prints. arXiv: 1512. 03385 [cs.CV].

Janowczyk, A. and Madabhushi, A. 2016. Deep learning for digital pathology image analysis: A comprehensive tutorial with selected use cases. In: *Journal of Pathology Informatics* 7.1, p. 29. doi: 10.4103/2153-3539.186902. eprint: http://www.jpathinformatics.org/article.asp?issn=2153-3539;year=2016; volume=7;issue=1;spage=29;epage=29;aulast=Janowczyk;t=6. url: http://www.jpathinformatics. org/article.asp?issn=2153-3539;year=2016;volume=7;issue=1;spage=29;epage=29;aulast= Janowczyk;t=6.

Simonyan, K. and Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. In: *ArXiv e-prints*. arXiv: 1409.1556 [cs.CV].

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J. and Wojna, Z. 2015. Rethinking the Inception Architecture for Computer Vision. In: *ArXiv e-prints*. arXiv: 1512.00567 [cs.CV].