Deep Convolutional Neural Networks for Breast Cancer Histology Image Analysis

Böckenhoff Bernhard, Dzmanashvili Demetre, Glowacz Jakub, Urcelay Lucía[1]

[1] Master in Artificial Intelligence, Universitat Potilècnica de Catalunya, {bernhard.bockenhoff, demetre.dzmanashvili, jakub.glowacz, lucia.urcelay}@estudiantat.upc.edu

Abstract

Breast cancer is one of the main causes of death worldwide, being the second leading cause of death among women. New recent Machine Learning techniques have helped to make a more accurate diagnosis, increasing the level of interobserver agreement and reducing the workload of medical practitioners. In this work we perform in the first place a study of State of the Art approaches for classification of histopathological images from breast tissue and in the second place an implementation of [Rakhlin 2018] approach with some modifications. This approach performs data augmentation in order to increase the number of images and uses several deep neural networks such as VGG-16 or InceptionV3 for feature extraction; moreover gradient boosted trees classifier is used in order to classify the images based on its features. The obtained accuracy for 4 class classification is 80,7%, which is 6,5 points lower than the original paper's accuracy 87,2 %.

***Keywords**: Breast Cancer, Histopathology, Computer Aided Diagnostic, Convolutional Neural Networks and Deep Learning*

1. Problem statement and goals

As a final work of the Computational Intelligence course, it has been decided to carry out both a comparative study of deep learning techniques for the early detection of breast cancer and the self-implementation of a Convolutional Neural Network for cancer detection from histopathological images based on state of the art literature.

In relation to the first objective, the comparative study of state-of-the-art literature, the first step has been to define the method for the search of papers related to the field of study. First, the following keywords were defined: Breast Cancer, Histopathology, Computer Aided Diagnostic, Convolutional Neural Networks and Deep Learning. Secondly, a search using these words was performed on the following platforms: Researchgate, IEEE Xplore and Mendeley. Finally, the search was limited to publications not prior to 2017 in order to obtain the most recent approaches. As a result, three papers published between 2017 and 2018 which propose different methods for histopathological image processing using Convolutional Neural Networks have been selected for study. These publications have been studied and analyzed in order to provide a comparative study, which can be read in the second section entitled "State of the Art".

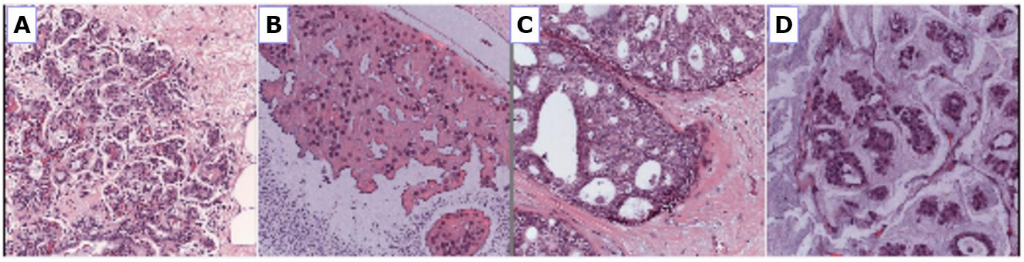
On the other hand, and in relation to the second objective which is the self-development of a Convolutional Neural Network, one of the papers studied has been selected in order to recreate its implementation and compare the results obtained. This section begins in point number three, "Implementation of a model for cancer classification", where an explanation of the methodology used to recreate the paper for the design of the Convolutional Neural Network and the obtained results are provided.

2. State of the Art

2.1. Motivation

Breast cancer is one of the main causes of death worldwide, being the second leading cause of death among women. Therefore, an early diagnosis is vital due to the significant increase in treatment success. Traditionally, morphological assessment and tumour grading is done by pathologists, however this process is tedious, increases the workload of the health specialists, is sometimes subjective and if the pathologist is not well trained this can lead to a wrong diagnosis. Computer-aided-diagnosis tools play an important role in empowering medical practitioners to make a more accurate diagnosis, increasing the level of interobserver agreement, reducing the workload and burnout and consequently making an impact in the health of patients.

Although many machine learning (ML) methods have been developed for the analysis of pathological images, recent deep learning (DL) based approaches have outperformed these last methods. In the first part of this project, the aim is to analyse three different deep learning approaches found in the literature following the method described in the above section. All the papers use as datasets histopathological images of breast tissue stained with hematoxylin and eosin (H&E). The image below shows the look of each histopathology image class.



Fig, 1: Random histopathology image set. (A) normal; (B) benign; (C) in situ carcinoma; and (D) invasive carcinoma. [Rakhlin 2018]

2.2. Study of the literature

The table below is a summary of the three approaches that are studied in this section, more specifically, it points out the method that has been followed in each case, the dataset that has been used, as well as the performance of the model.

Paper	Approach	Dataset	Classes	Accuracy	Sensitivity	Specificity
[Rakhlin 2018]	CNNs and gradient boosted trees	BACH (400 images)	Benign/ Malignant	93,8 %	95,6 %	88 %
			Normal/Benign/Insitu/Invasive	87,2 %	-	-
[Nazeri 2018]	2 CNN (patch bases and image based)	BACH (400 images)	Normal/Benign/Insitu/Invasive	95 %	-	-
[Araújo 2017]	CNN and SVM	Bioimaging 2015 (269 images)	Carcinoma/ non-carcinoma	83,3 %	95,6 %	-
			Normal/Benign/Insitu/Invasive	77,8 %	-	-

Table 1: Overview of the different approaches to be analysed

The analysis will be conducted in the following way: a pairwise analysis will be made between [Rakhlin 2018] and [Nazeri 2018] which use the same dataset, due to the data being a conditionat over the model development, and then [Araújo 2017] will be explained and overall comparison and conclusions will be made.

BACH dataset. Rakhlin 2018 and Nazeri 2018

BACH dataset consists of 400 H&E stain images of size 2048 × 1536 pixels, moreover, each image is labeled with one of the four balanced classes: normal, benign, in situ carcinoma, and invasive carcinoma. Even though the dataset used by [Rakhlin 2018] and [Nazeri 2018] is the same, the approach proposed by each one is different and thus, the results. On the one hand, [Rakhlin 2018] utilizes several deep neural networks architectures and gradient boosted trees for classification, achieving 87,2 % accuracy. On the other hand, [Nazeri 2018] proposes a patch-based technique which consists of two consecutive Convolutional Neural Networks, the accuracy of which is 95 %. In the following paragraphs it will be explained in more detail how each model is constructed; nonetheless **[Rakhlin 2018]** will be analysed in more depth as this paper has been selected by us to be self-implemented (see section 3).

The first problem that both approaches face is the low data quantity. To solve this problem, both approaches conduct data preprocessing and augmentation. Regarding to [Rakhlin 2018] they follow the next steps: for each image 50 random color augmentations are performed followed by an adjustment of the H&E. Then images are downscaled and crops of 400 x 400 pixels and 650 x 650 pixels are extracted. The approach of [Nazeri 2018] is slightly different. In the first place, to create the patches, a sliding window of size 7 x 7 and a stride of 256 is used; as a result, 35 overlapping image patches are obtained from each individual image. Then data augmentation is implemented the following way: rotation by 4 multiples of 90 degrees and random color perturbations are applied to each image.

The next step of [Rakhlin 2018] is to use 3 CNN encoders to ecode the previously obtained crops into 20 descriptors, which are in turn combined through 3-norm pooling into a single descriptor. The result of the preprocessing and augmentation process is the obtention of 50 (number of color augmentations) x 2 (crop sizes) x 3 (CNN encoders) = 300 descriptors, for each original image. See Fig. 2 below.

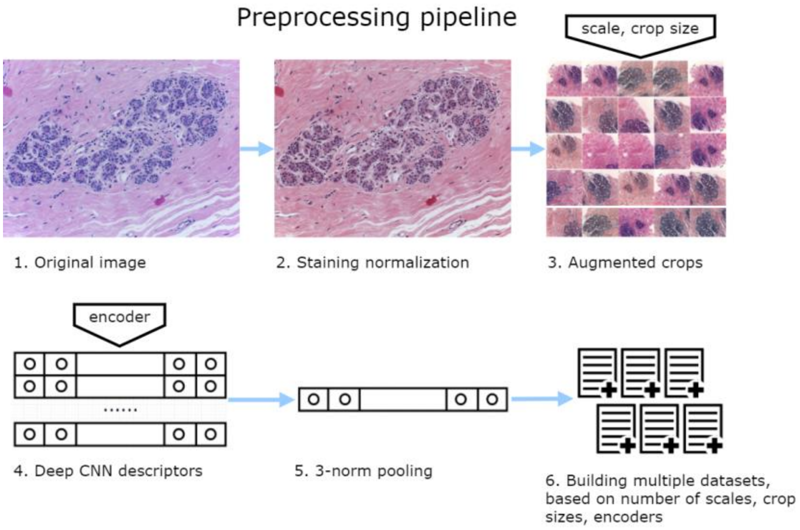


Fig. 2: Overview of the preprocessing pipeline of [Rakhlin 2018]

Then, feature extraction is conducted. [Rakhlin 2018] uses standard pre-trained ResNet-50 [He 2016], InceptionV3 [Szegedy 2016] and VGG-16 [Simonyan 2014] networks from Keras library. First, the fully connected layers from each model are removed to allow the network to consume images of an arbitrary size. Then, GlobalAveragePooling operation is applied to the last convolutional layer in ResNet-50 and InceptionV3, and to the four internal convolutional layers in VGG-16. Finally the features are concatenated into one feature vector of length 1408, see Fig. 3 below.

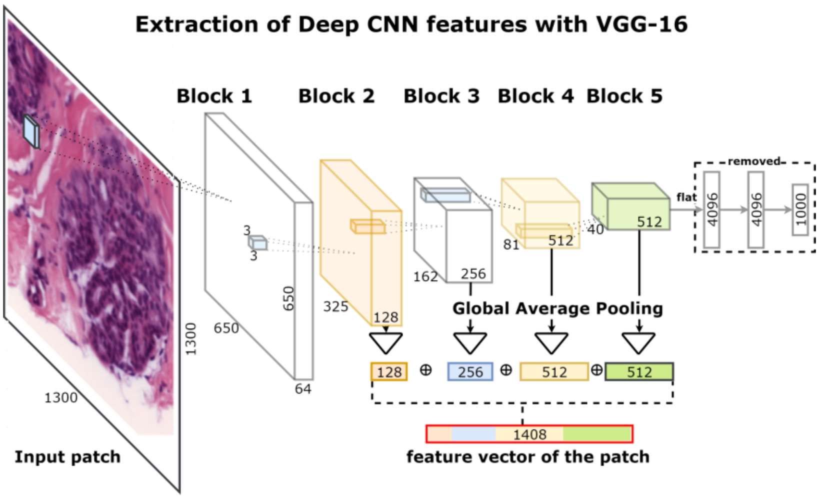


Fig. 3: Overview of the network architecture for feature extraction of [Rakhlin 2018]

Finally, the training phase is done in the following way. Data is splitted into 10 stratified folds which preserve class distribution and the model is trained for each combination of the encoder, crop size and scale using gradient boosted with 10-fold cross-validation. Furthermore, each dataset is recycled 5 times with different random seeds in LightGBM adding augmentation on the model level. For the test data, the same preprocessing and data augmentation steps are followed. The 300 descriptors extracted for each image are used to train the model for particular patch size and encoder; and predictions are averaged over all augmentations. As a final step, the predicted class is defined by the maximum probability score.

In contrast, feature extraction from [Nazeri 2018] proposes a patch-based technique which consists of two consecutive Convolutional Neural Networks: patch-wise CNN followed by an image-wise CNN that classifies images into four classes. The patch-wise CNN is designed using 9 convolutional layers, each three followed by a pooling layer. Moreover, all convolutional layers are followed by batch normalization and ReLU activation functions. Once this network has been trained, the classification layer is discarded and the last convolutional layer is used to extract feature maps. Then, the extracted feature-maps are concatenated and used as input for the image-wise network, which is also trained against ground truth labels using categorical cross-entropy loss. This network learns to classify images based on local features extracted from the patches and global information which is shared between the different patches. Its architecture is similar to the patch-wise one: it is composed of a series of 3 x 3 convolutional layers, each followed by batch normalization and ReLU activation function. The convolutional layers are followed by 3 fully connected layers with a softmax classifier at the end. Besides, as overfitting is a problem within this network, regularization is done using dropout and early stopping if accuracy does not improve.

Bioimaging 2015 challenge dataset. Araújo 2017

This dataset is composed of 269 high-resolution (2040 × 1536 pixels) annotated H&E stain images from the Bioimaging 2015 breast histology classification challenge. As in the previous dataset, each image is labeled with one of the four balanced classes: normal, benign, in situ carcinoma, and invasive carcinoma. [Araújo 2017] proposes a CNN which architecture is designed to retrieve information of the images at different scales, including both nuclei and overall tissue organization. Moreover, the features extracted by the CNN are also used for training a Support Vector Machine (SVM). The accuracy obtained by Araújo et al. for four class classification is 77,8 %.

The first step conducted is data preprocessing and augmentation. To start with, images are preprocessed using techniques such as optical density conversion, singular value decomposition (SVD) and histogram stretching. Also, data is augmented by dividing images in 512 x 512 patches and applying to them normalization, rotation as reflections.

In the second place, image-wise classification is performed in the following way: first the original image is divided into twelve contiguous non-overlapping patches. The patch class probability is computed using two methods: the patch-wise trained CNN and CNN + Support Vector Machines classifiers. The proposed patch-wise classification CNN is composed of several convolutional-pooling layers followed by three fully connected layers. The final fully-connected network performs the integration of the information for the whole image patch, and provides the final classification. All layers use ReLU as activation functions except for the output layer, which uses softmax function. The network also uses an adaptive learning rate for gradient descent back-propagation; besides, the selected loss function is categorical cross entropy. Finally, the image-wise classification is obtained by combining the classification results of all the image patches.

Moreover, as stated before, the features extracted by the CNN are also used for training a SVM classifier but the 77,8 % of accuracy obtained by the CNN remains constant of using CNN + SVM.

Summary of previous studies

A common step followed by the three approaches is data augmentation. In this case this is a necessary step due to the low number of images that are available (400 or 269). When CNNs are trained with a low number of images model overfitting can occur and as a consequence, the model can present a poor generalization ability and thus a bad performance in the testing phase. Another common technique used by the three approaches is the use of patches. Dividing the image into smaller parts not only benefits data augmentation but also allows to extract information in different scales and thus discover more silent features. While [Rakhlin 2018] uses 3 Convolutional Neural Networks to extract features and classify them using gradient boosted trees and [Nazeri 2018] uses a patch-based CNN before feeding the features to the image-wise CNN, [Araújo 2017] uses a single CNN to extract the features of the patches and also to perform classification. This simpler approach, compared to the other papers, can be one cause of its poorer performance. Moreover, the use of pre-trained models from [Rakhlin 2018] show worse performance in terms of accuracy than fully trained CNNs from [Nazeri 2018], which results in a gain of 7,8 points in accuracy over its competitor, this may also be due to the fact that [Nazeri 2018] extract features both patch-wise and image-wise, thus obtaining more features.

3. Implementation of a model for cancer classification

The second objective of this project is to implement a model from the literature. [Rakhlin 2018] paper titled “Deep Convolutional Neural Networks for Breast Cancer Histology Image Analysis” has been chosen to base or work on. The final goal is to make some modifications to this approach and get similar or better results to the ones of the paper. To develop this work, PyTorch library has been mainly used.

3.1. Dataset

BACH dataset consists of 400 H&E stain images of size 2048 × 1536 pixels with pixel size of 0.42 µm x 0.42 µm., moreover, each image is labeled with one of the four balanced classes: normal, benign, in situ carcinoma, and invasive carcinoma. Each of the classes contains 100 images of their class. This dataset is publicly available in the following direction[1]. Moreover, the code of this implementation is available in the *.ipynb* file attached to the delivery task.

3.2. Data preprocessing and augmentation

In our implementation we developed a method for loading, preprocessing and augmenting the data. The data loader class has two responsibilities: first loading the high resolution histological

[1] https://www.sciencedirect.com/science/article/abs/pii/S1361841518307941/

images, and second augmenting them to generate more data in order to overcome overfitting of the model. Regarding the data augmentation the following process is done:

- Randomly applying a flip along the x axis of the image
- Randomly applying a flip along the y axis of the image
- Randomly applying a rotation of 0° 90° 180° 360°
- Randomly manipulate the following attributes with a factor between (0.5,1.5)
    - Image contrast
    - Image color
    - Image brightness

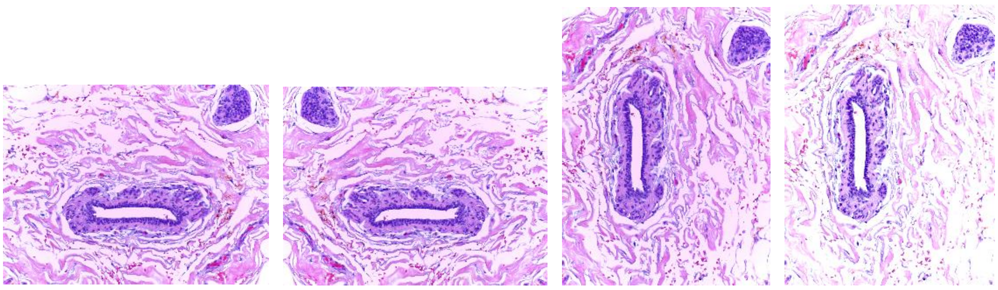


Fig. 4: Data augmentation of histopathological images from dataset. From left to right: x flip, y flip, rotation, color enhancement

Each of these augmented images will be cropped into twenty pieces. In order to generate these crops a random point inside the image is selected. Around this point a 512 pixels times 512 pixels crop is cut out. With this combination of different parameters we can generate enough unique feature vectors to properly train a classifier. The features of these twenty crops will be combined to make up one instance for the classification process.

3.3. Feature extraction

For the feature extraction we have used three pre-trained CNNs and modified each to our needs. The feature vectors which output from each model are then concatenated for each crop. In the next sentences the main ideas behind the three CNNs will be explained.

The **VGG16** is a CNN which follows an arrangement of convolution and max pooling layers. The architecture is pictured on the following image:

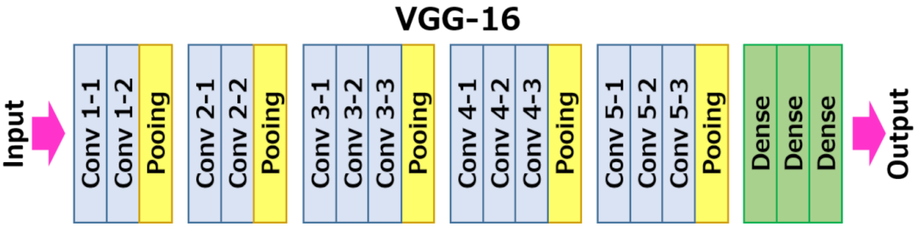


Fig. 5: VGG-16 Convolutional Neural Network architecture

Nevertheless, in order to only get the features from this network and not the final classification, only the feature extraction sub-architecture has been used, which does not include the last 3 fully connected layers of the network. Compared to the implementation in the paper [Rakhlin 2018], we did not use the global average pooling method for the last four convolutional layers. Instead of that, we kept the original way of average pooling in these layers.

The **Inception V3** is a CNN which uses Label Smoothing and Factorized 7 x 7 convolutions. It is also emphasizing the use of an auxiliary classifiers, which propagate label information lower down the network. In our implementation we change the last layer to a flattening one.

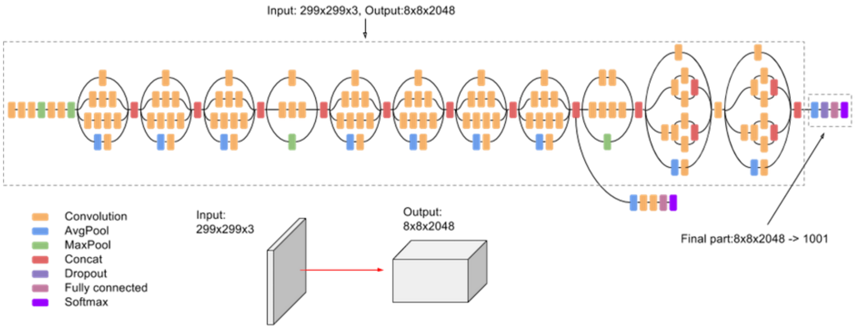


Fig. 6: Inception V3 Convolutional Neural Network architecture

The **Resnet 50** consists of 5 stages each with a convolution and Identity block. Each convolution block has 3 convolution layers and each identity block also has 3 convolution layers. We replaced the last fully connected layer, with a flattening layer in order to receive the feature vector for each crop.

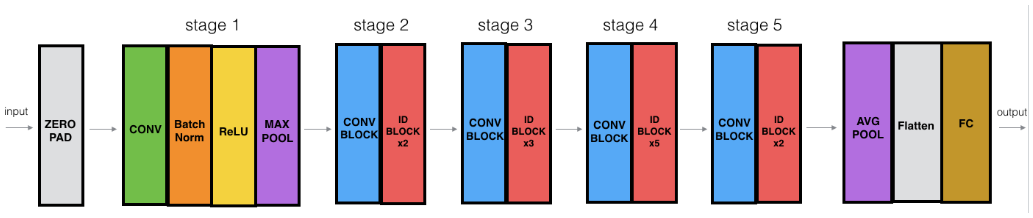


Fig. 5: Resnet-50 Convolutional Neural Network architecture

Each of the twenty crops is passed to the feature extractor. We combine the twenty resulting feature vectors via norm pooling:

$$d_{pool} = (\frac{1}{N}\sum_{i=1}^{N} \quad (d_i)^p \ )^{1/p}$$

The combined feature vector which result from the previous step is used in the next step in order to train a classifier which is able to classify the images in the four different cell types.

3.4. Classification

The previously constructed feature vectors belong to one of the four classes Normal, Benign, InSitu, Invasive. Hence it is a multiclass classification problem. The training dataset contains 15000 feature vectors based on 150 images. The test dataset consists of 1000 feature vectors constructed from 40 different images.

We choose a gradient boosted classifier as our approach to solve this problem. In the class of ensemble learning, many weak models are combined to get a strong predictive model. Typically decision trees are iteratively added to the model to improve the overall model. In gradient boosted models this idea is supplemented with the ability to optimize an arbitrary differentiable loss function. The implementation we used is part of the SciKit-learn library.

The hyperparameters of the algorithm are the number of classifiers, learning rate and max depth. For the initial run we set the number of classifiers to 100, the learning to 1.0 and the max depth of the trees to 1.

4. Results and discussion

Table below shows the confusion matrix achieved by the model that has been deployed, where the amount of correctly classified cases for each class can be observed. It can be seen that images labeled as InSitu cancer cells are just negatively predicted as non cancer cells 21 times and images labeled as Invasive cancer cells are just negatively predicted as healthy cells 8 times in the testing; in the medical field is important to have a low false negative rate in order to not diagnose sick patients as unhealthy ones as this could be fatal for them. It can also be observed that people who are healthy are diagnosed as having cancer just five times, these cases are called false positives. Moreover, the accuracy obtained by our model is 80,7 %, which is 6,5 points lower than the accuracy obtained by the paper on which we have based our work on [Rakhlin 2018].

		True labels			
		Normal	**Benign**	**InSitu**	**Invasive**
Predicted labels	**Normal**	**283**	63	21	8
	Benign	28	**160**	3	7
	InSitu	5	26	**222**	16
	Invasive	0	12	4	**142**

Table. 4: Confusion matrix from our implementation

As it is visible on table 4, there are some misclassifications in our implementation. The most important thing in this classification task is a high recall on all of the classes that indicate cancer. Therefore it is possible to simplify the problem by reducing the problem to two classes: Cancer or No cancer. The resulting confusion matrix will look as following:

		True labels	
		No cancer	**Cancer**
Predicted labels	**No cancer**	**283**	92
	Cancer	33	**582**

Table. 5: Simplified confusion matrix from our implementation

From table 5 we can compute the precision and recall for both classes: Cancer and No cancer. The results are shown on a plot below:

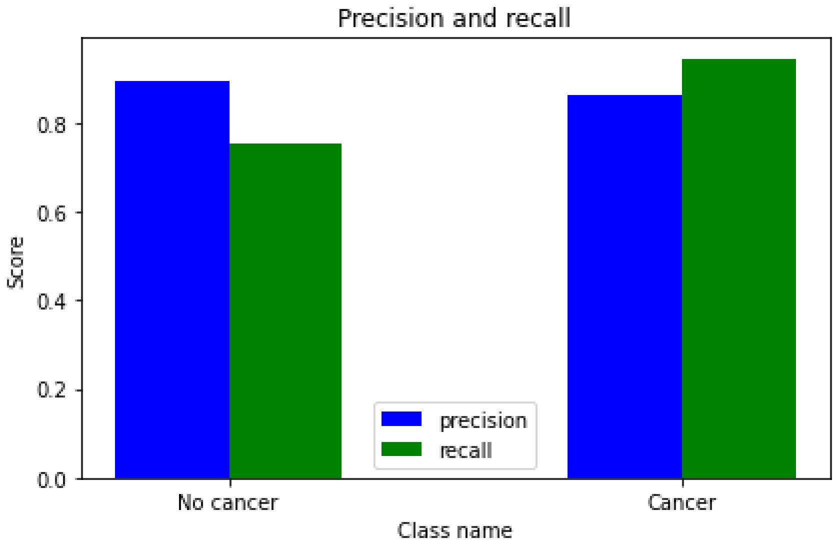


Fig. 7: Precision and recall for simplified classifier

From figure 7, it is possible to see that the precision is higher for No Cancer class, and for Cancer class recall is higher. This is exactly what we wanted to achieve. In cancer classifiers it is important to have high recall on the cancer classes, to avoid situations where cancer is not detected, when in fact it is present.

On the other hand, the plot below shows the loss plot over the gradient boosted trees classifier. It can be seen that the error of the classification decreases over time and thus the performance of the classifier improves.

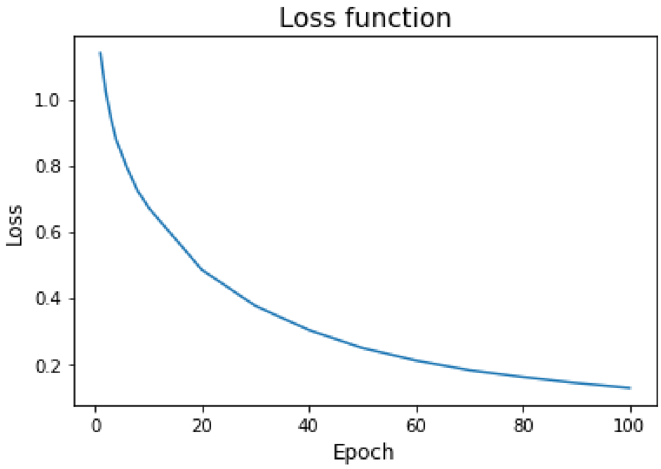


Fig. 7: Training loss from gradient boosted trees classifier

5. Strengths and weaknesses

The biggest problem we faced solving this task is the low data quantity. Only a couple hundred images are usually not enough to train a deep neural network from scratch. To combat this we utilised various data augmentation techniques to generate a dataset big enough to properly train a classifier on and avoid overfitting the model; the data augmentation has consisted in doing color alterations as well as flips and rotations.

Another issue that we faced was the excessively high time required to fully train a network from scratch. This issue has been solved by using pre trained networks VGG16, Resnet50 and InceptionV3, which have all been trained on ImageNet. The use of these networks has not only been helful to reduce the time but also to be used as feature extractors.

Despite these difficulties we were able to generate acceptable results with the explained setup.

6. Conclusions and future work

We implemented a method for the classification of breast cancer detection. We used data augmentation techniques and pre-trained CNNs to make up for the low quantity of data in the BACH dataset. The augmentation techniques were the key to achieve high accuracy. Without an artificial enlargement of the dataset, the amount of images that were in the dataset would be clearly too small to train an usable classifier. As a rule of thumb, to train a model, it needs around 1000 images per class. In our case it would be 4000 of images and as the histopathological images are harder to interpret it would still not be enough. That shows that the augmentation of the data is really important and it has been proved to be working in the context of breast cancer detection.

Comparing our results to the original work it becomes clear that our model performs slightly worse. This could either be due to the light changes we did in the feature extraction process or due to suboptimal hyperparameters of the classifier. It is possible to perform a grid search to further improve our results.

Because of the nature of the problem, it is preferable to falsely detect breast cancer than falsely not detecting breast cancer. Therefore it could be desirable to implement a custom loss function which reflects these properties in a future work. Lastly, regarding the classification step other different ensemble or non-ensemble classifiers could be tested based on the same constructed data set to try to get a more accurate classification of the images.

7. References

[Nazeri 2018] Nazeri, K., Aminpour, A., & Ebrahimi, M. (2018). Two-Stage Convolutional Neural Network for Breast Cancer Histology Image Classification. Image Analysis and Recognition, 717–726. https://doi.org/10.1007/978-3-319-93000-8_81

[Rakhlin 2018] Rakhlin, A., Shvets, A., Iglovikov, V., & Kalinin, A. (2018). Deep Convolutional Neural Networks for Breast Cancer Histology Image Analysis. https://doi.org/10.1101/259911

[Araújo 2017] Araújo, T., Aresta, G., Castro, E., Rouco, J., Aguiar, P., Eloy, C., Polónia, A., & Campilho, A. (2017). Classification of breast cancer histology images using Convolutional Neural Networks. PLOS ONE, 12(6), e0177544. https://doi.org/10.1371/journal.pone.0177544

[Simonyan 2014] Simonyan, K., & Zisserman, A. (2014). Very Deep Convolutional Networks for Large-Scale Image Recognition. ArXiv. https://arxiv.org/abs/1409.1556

[Szegedy 2016] Szegedy, C., Vanhoucke, V., Ioffe, S., & Shlens, J. (2016). Rethinking the Inception Architecture for Computer Vision. Computer Vision and Pattern Recognition 2016. https://doi.org/10.1109/CVPR.2016.308

[He 2016] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). https://doi.org/10.1109/CVPR.2016.90

VGG-16 Convolutional Neural Network architecture. (2018). [Illustration]. Neurohive. https://neurohive.io/en/popular-networks/vgg16/

Inception V3 Convolutional Neural Network architecture. (2022). [Illustration]. Google Cloud. https://cloud.google.com/tpu/docs/inception-v3-advanced

Dwivedi, P. (2019). Resnet-50 Convolutional Neural Network architecture [Illustration]. Towards Data Science. https://towardsdatascience.com/understanding-and-coding-a-resnet-in-keras-446d7ff84d33