Breast cancer detection in histopathological images using convolutional neural networks

Andrio Rodrigo Corrêa da Silva[1], **Iális Cavalcante de Paula Júnior**[1]
Márcio André Baima Amora [1]

[1]Graduate Program in Electrical and Computer Engineering –
Federal University of Ceará (UFC), Sobral Campus, Sobral-CE

andrio.rodrigo.silva@hotmail.com, ialis@sobral.ufc.br

marcio@sobral.ufc.br

Abstract. *Breast cancer is one of the biggest causes of death among women around the world. Diagnosing this disease early can offer better treatment to the patient. Intelligent systems have been used for the detection of diseases using images. In this work a convolutional neural network was used for the detection of breast cancer in histopathological images through Keras library and TensorFlow framework. Models were created for 4 datasets with different magnifying factors (40x, 100x, 200x and 400x). Using k-fold cross-validation, it was found that there was a better result for the set of 400x images with 98.44% accuracy in the training data. The set of 200x images obtained a better result for recall and f1-score.*

1. Introduction

Breast cancer has become the second most common cancer in the world, it has been caused deaths mainly in women. Men also have a number of deaths from this disease [Parkin 1998]. According to [Bray et al. 2018], 2.088.849 million new cases of breast cancer were diagnosed and 626.679 deaths were recorded around the world, these numbers correspond only to the year of 2018. These are quite expressive numbers.

Uncontrolled growth of the cells in the breasts is what characterizes the onset of breast cancer, this uncontrolled growth eventually leads to metastasis when the cancer spreads to other organs. The manifestation of this disease can be observed in different parts of the breasts, however, it is in the breast ducts that this disease is more widespread. Breast cancer has some symptoms that should be observed, such as: some sporadic mass in the breasts, changes in breast size and shape, differences in skin color of the breasts, change in skin texture, changes in nipples and other symptoms [Osareh e Shadgar 2010].

In modern medicine there are some methods that are used for the detection of breast cancer, such as biopsy, mammography and ultrasound. When the tumor is detected, it can be classified into two distinct types, the first is the benign one, that is, it has cells similar to those that originated it and there is no risk of metastasis, the second is the malignant one, that is, it has the ability to spread to other organs[Gayathri e Sumathi 2016].

For existing breast cancer detection methods (biopsy, mammography, and ultrasound), there are technologies that can be implemented aiming at an early diagnosis of the

disease, such as, machine learning algorithms. Some techniques such as image analysis have been implemented for applications such as disease detection even before it manifests itself in the patient [Jangade e Chauhan 2006].

One of the main tools that has emerged in recent years in the category of machine learning is the TensorFlow, created by Google, which operates on a large scale and in heterogeneous environments. TensorFlow supports a variety of applications focused primarily on the training and inference of deep neural networks. A great advantage is the ability to train machine learning models using GPU (Graphics Processing Unit), thus allowing the training cost to be reduced compared to CPU usage (Central Process Unit)[Abadi et al. 2016].

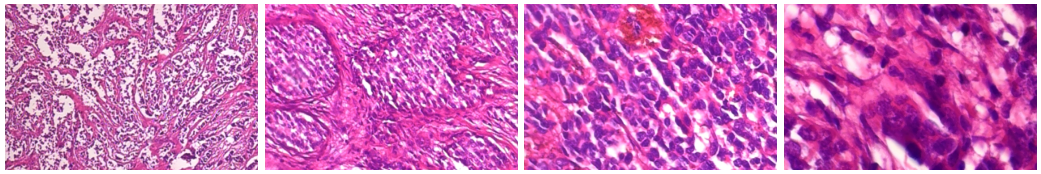
This work aims to use Keras, a Python library, which uses the TensorFlow as a backend, for the creation of the neural network model that will be used for the classification of breast cancer using histopathological images. In order to classify these images, it was used a special type of deep neural network, the convolutional neural network (CNN), it has been increasingly used for image recognition problems. More specifically, a variant of VGGNet has been used.

In Section 2, is made a description of the tools and dataset that were used in this work. In Section 3, is made a detailing about the steps that were accomplished in the neural network training methodology. In Section 4, the results of the training are presented. In Section 5, an overview is presented on the application of the methodology used, if it presented and met the objectives initially discussed.

2. Materials and Methods

2.1. Dataset

The data set used in this work has a total of 7909 images, collected from 82 patients, distributed in 4 degrees of magnification (40x, 100x, 200x and 400x), divided into 2480 for the benign type and 5429 for the malignant type. All images have the following format: 700x460 pixels, 3-channel RGB, 8-bit depth in each channel, PGN extension. This dataset was built by P&D *Laboratory - Pathological Anatomy and Cytopathology* located in the State of Paraná, Brazil [Spanhol et al. 2016]. Figure 1 shows an example of a malignant tumor at 4 degrees of magnification.



(a) 40x **(b)** 100x **(c)** 200x **(d)** 400x

Figure 1. Malignant tumor at 4 degrees of magnification

2.2. Keras

Keras is a Python library developed to facilitate the creation of high level neural networks, such as deep neural networks, for example. One of the main advantages is the ease and speed with which a neural network can be constructed, differently from the case of only

using TensorFlow [Chollet 2015]. Both tools allow the neural networks to be trained using either a CPU or a GPU, the latter being important because the training time ends up being reduced.

In this work TensorFlow will be used as a backend tool since using only Keras does not allow low level operations such as tensor and convolution multiplication.

3. Methodology

For this work, a network based on VGGNet was used, except that the one implemented here has a smaller number of layers, only 6 of convolution and 2 of fully connected layers. Figure 2 shows the network architecture used.

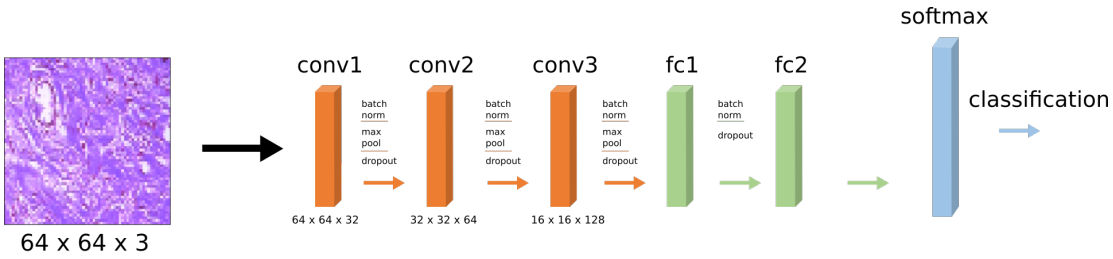


Figure 2. Network architecture

In the convolution layers, filters of 32, 64 and 128 were applied and each filter has a window of size 3x3. For each activation layer, after the convolution layer, it was used the ReLu (Rectified Linear Unit) function, widely used in neural network problems. Then a layer called Batch Normalization was allocated after the activation layer, allowing to improve network performance. After this layer is also added another, called MaxPooling, which allows to reduce the dimensionality of the input characteristics. It was also used 2 fully connected layers with 512 hidden neurons in the first layer and 2 hidden neurons in the second layer (number of classes). And finally, a Softmax layer was added, this layer was used for classification, benign or malignant.

Before performing the network training, it was necessary to apply a simple pre-processing in the images to fit the input parameters of the algorithm. The size of the images has been reduced from 700x460 pixels to 64x64 pixels. The pixels were also normalized, each was divided by the value 255, resulting in values between 0 and 1. A final adjustment made before performing the training was to apply ImageDataGenerator function from Keras, which aims to increase the number of data for the training simply by varying the existing data, using horizontal/vertical inversions, rotations, variations in brightness, shifts etc. This function is applied only after separation of the data in training and test.

In order to perform the training of the network, the data were divided into 80% for training and 20% for test, and some images were removed, that is, they were not used in the training, so that it was possible to perform a classification validation test at the end of the training, a kind of real situation simulation.

For each of the image sets, a training was performed using k-fold cross-validation. Table 1 shows the total of images available in the original dataset, the total that was used for training, for testing, and the total that was used for the final validation, respectively.

Table 1. Distribution of images in Training, Final Testing and Validation

Dataset	Total images (Benign - Malignant)	Total images training	Total images testing	Total images final validation (Benign - Malignant)
40x	652 - 1370	1531	383	24 - 57
100x	644 - 1437	1599	400	24 - 58
200x	623 - 1390	1544	387	24 - 58
400x	588 - 1232	1401	351	24 - 58

After dividing the data for training and for testing, some parameters were adjusted for the training of the network. The Dropout layer, which is used for a better generalization of the model and avoiding an overfitting of the data, is configured with a rate of 25% in conv1, conv2 and conv3. In fc1 this value increases to 50%. Table 2 shows other parameters that have been set up for the network.

The learning rate will determine how fast the weights change in the neural network. The number of epochs refers to the number of times the data will be presented to the algorithm, at each new epoch the value of the weights is also changed. Batch size refers to the amount of data that will be used to each iteration of the training. The value of the optimizer indicates the algorithm used to perform the update of the weights.

Table 2. Adjusted parameters

Parameter	Value
Learning rate	0.01
Number of epochs	500
Batch size	32
Optimizer	Stochastic gradient descent optimizer (SGD)

The flowchart of Figure 3 shows the methodological step-by-step method used to perform this work.

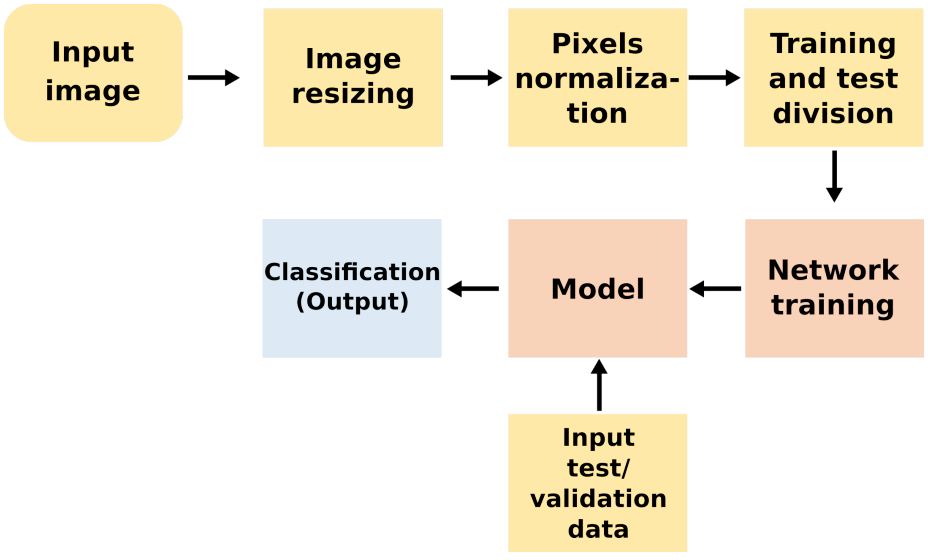
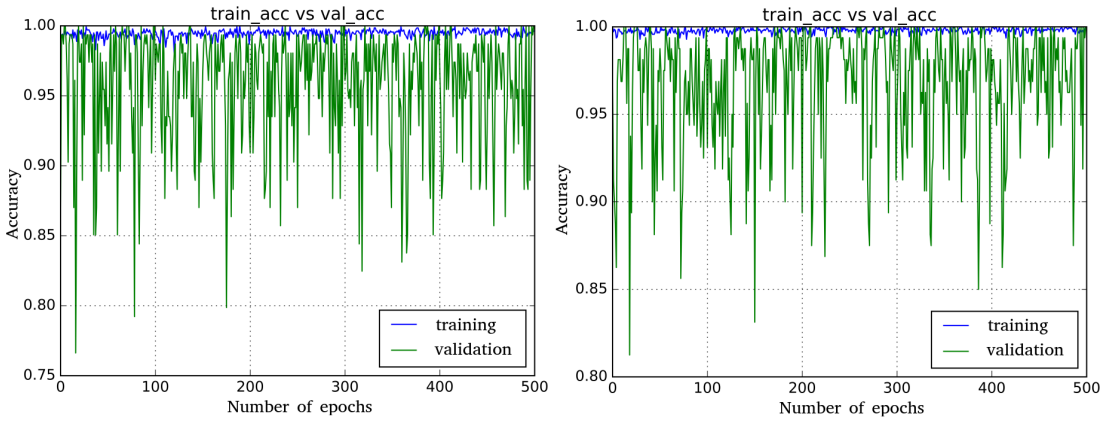


Figure 3. Methodological flowchart

4. Results

In this work the training of a deep neural network was carried out, with an average duration of 70 minutes for each k-fold, all with the same parameter, but with different datasets. In total, 40 models were generated, 10 models for each dataset. In this section will be evaluated which were the best models.

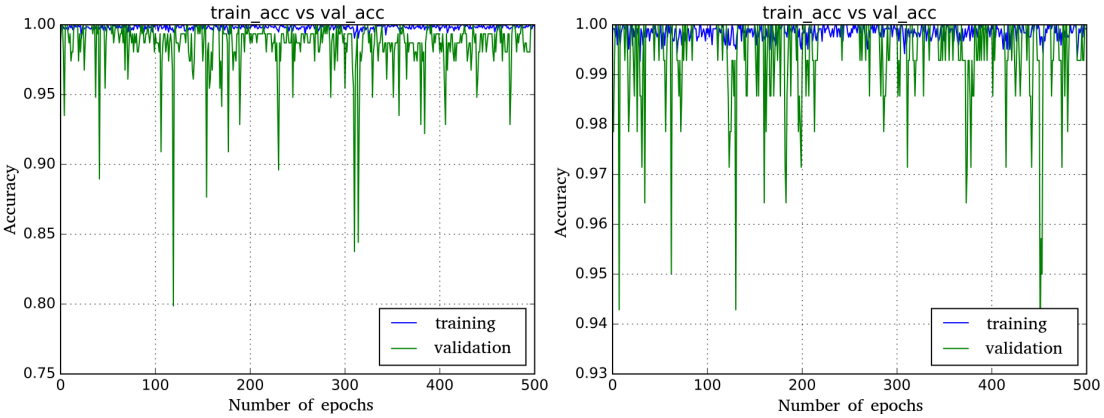
Figures 4 and 5 show the training and test rate reached in the best k-fold for each of the datasets (40x, 100x, 200x and 400x).



(a) Training 40x - k=4 **(b)** Training 100x - k=6

Figure 4. Training and test rate (40x and 100x)

It can be noted that in Figure 4.a the training and validation rates are very close to each other. In Figure 4.b, there is a much smaller variation, and unlike Figure 4.a, where the lowest index reached was around 77%, the lowest index reached with the 100x set was around 82%.



(a) Training 200x - k=7 **(b)** Training 400x - k=8

Figure 5. Training and test rate (200x and 400x)

Figure 5.a shows excellent values of training and validation, it is possible to note that at the time 500 the validation rate reached a value higher than 95%. Figure 5.b also presents an excellent result for the trained network, also reaching a value higher than 95% in the validation of the data and with a smaller variation in relation to the other sets. In order to perform the evaluation of the generated models, some important values were

used, such as the accuracy of training data (train_acc) and accuracy of the validation data (val_acc), Table 3 shows these values for the best k-fold of each of the datasets.

Table 3. Training and test values

Dataset	Training accuracy	Validation accuracy
40x (k=4)	99,23%	100%
100x (k=6)	99,88%	100%
200x (k=7)	99,86%	100%
400x (k=8)	99,89%	100%

In Table 3 it can be seen that all models obtained 100% accuracy in validation data, this does not necessarily mean that the model will actually perform 100%, there may be certain data that will be presented to the model and may be classified incorrectly. This misclassification in a so-called optimal model can occur precisely in data representing outliers, data that have unusual characteristics when related to others of the same dataset.

Unfortunately only the training and test accuracy values are not sufficient to verify the performance of a neural network model. Table 4 provides additional metrics for assessing whether a network performs efficiently or not.

Table 4. Metrics obtained

Dataset	Precision benign - malignant	Recall benign - malignant	f1-score benign - malignant	Quantity benign - malignant
40x (k=4)	0.92 - 0.98	0.95 - 0.96	0.93 - 0.97	116 - 267
100x (k=6)	0.96 - 0.93	0.82 - 0.99	0.89 - 0.96	119 - 281
200x (k=7)	0.97 - 0.96	0.91 - 0.99	0.94 - 0.98	116 - 271
400x (k=8)	0.96 - 0.92	0.86 - 0.98	0.91 - 0.95	126 - 225

The precision value indicates how precious the model is in relation to the predicted positive values, that is, how many values are actually positive. In Table 4 it is possible to notice that the model for the 200x set obtained an excellent precision for the benign type, and the model for the 40x set obtained an excellent precision for the malignant type. Another important value, recall, indicates the frequency that a given example was classified as being of a given class, the 40x and 200x sets presented good results when compared to the 100x and 400x sets. The f1-score value is generally used when there is a need to obtain a balance between precision and recall values, thus indicating a general quality of the model.

As previously mentioned, to perform a better evaluation of the models, k-fold was used, taking 10 as the value of k, for the validation datasets. Table 5 presents the values of each of the folds for each of the datasets.

Table 5. K-fold values obtained

Dataset	k=1	k=2	k=3	k=4	k=5
40x	94,81%	96,75%	98,05%	100%	97,40%
100x	87,50%	90,00%	95,00%	84,38%	98,75%
200x	87,18%	95,48%	94,84%	98,70%	96,75%
400x	92,91%	96,45%	99,29%	99,29%	99,29%

k=6	k=7	k=8	k=9	k=10	Mean Standard-Deviation
98,69%	91,45%	95,39%	98,03%	91,45%	96,20%±2,77%
100%	94,38%	98,75%	97,50%	96,23%	94,25%±5,00%
97,40%	100%	100%	98,70%	100%	96,91%±3,69%
98,57%	99,29%	100%	99,28%	100%	98,44%±2,07%

It is notable to notice that the 400x set presents better results in relation to the accuracy variation in the validation data, since the standard deviation is only 2.07%, the set 40x presents the second smallest standard deviation, 100x shows a slightly higher standard deviation, 5%. Table 6 presents the mean values of the k-fold with the validation data compared to the results obtained in [Spanhol et al. 2016], which uses the following classifiers: 1-nearest neighbor (1-NN), quadratic linear analysis (QDA), support vector machine (SVM) e random forests (RF) and that uses an abstract model called oracle, which selects the classifier that predicted the correct label for a given query in a given sample [Spanhol et al. 2016].

Table 6. Evaluation with k-fold

Method	Dataset			
	40x	100x	200x	400x
1-NN	91,50%	91,50%	93,10%	91,50%
QDA	100%	96,90%	96,20%	97,70%
RF	92,30%	91,50%	90,80%	92,30%
SVM	95,40%	95,60%	94,60%	97,70%
CNN	**96,20%**	**94,25%**	**96,91%**	**98,44%**

The model generated for 40x set with CNN obtained better results than 1-NN, RF and SVM, only below the classifier QDA. The model for the 100x set obtained inferior performance in relation to the QDA and SVM classifiers, but remained above the 1-NN and RF classifiers. For the 200x and 400x images, there was a better result than all the classifiers. It is worth noting the mean of 98.44% obtained by the last set of images, 400x.

After the training was performed with the four sets of images using the cross-validation k-fold, the model that presented the best value of accuracy on the training data was selected, these values are shown in Table 5. For the 40x set, the best training was where k=4. For the 100x set, the best training was where k=6. For the 200x set, the best training was where k=8. For the 400x set, the best training was where k=8. With these models the dataset that was initially separated for final validation was used. The results of this validation are shown in Table 7.

Table 7. Final validation values

Dataset	Total images (Benign - Malignant)	Benign classification (rights - wrongs)	Malignant classification (rights - wrongs)
40x	24 - 57	23 - 1	54 - 3
100x	24 - 58	18 - 6	58 - 0
200x	24 - 58	23 - 1	58 - 0
400x	24 - 58	21 - 3	56 - 2

It can be observed that the 200x set obtained excellent results both for the classification of benign and malignant, the latter being the most important result in a final diagnosis, called true positive (TP). The 400x set also showed good results, even though it was the best on the mean of k-folds, it still fell below in classifying malignant ones when compared to the 200x. The 100x set presented the highest number of wrong classifications for the benign type, which was not surprising since the recall value for benign presented in Table 4 also presented a low index, as well as the 400x. The 40x set presented a higher number of wrong classifications for malignant, and obtained a similar performance to the set 200x for right classification of benign.

5. Conclusion

In this article was used a convolutional neural network derived from a VGGNet network, the one used in this article presents a reduced number of layers. With the network implemented, the training and generation of models used for the classification of breast cancer using histopathological images were performed, the image was classified as benign or malignant.

The use of convolutional neural networks allows certain aspects, such as feature extraction, to be performed automatically, unlike common neural networks. It was also possible to perform the increase of the data set with the internal function of Keras that allowed the images to be rotated, inverted and that there was variation in the brightness, which allowed new information to be generated.

The performance of the models created allows to conclude that the use of a CNN for problems of image classification can be of great value, obtaining even superior results compared to common neural networks. For the datasets used (40x, 100x, 200x and 400x) it was possible to notice that the metrics evaluated obtained excellent results for 200x and 400x in the training metrics. For other evaluative metrics, such as f1-score, the 40x and 200x sets obtained better results.

In general, the use of the CNN architecture to evaluate images that have 200x of approximation is seen as ideal, since it presents values of Recall and f1-score more balanced when compared to the other sets. The results can be further improved by using better computational resources and a larger number of images for better network learning.

References

Parkin, D. (1998). Epidemiology of cancer: global patterns and trends. In *Toxicology Letters*, pages 227–234. ScienceDirect, Lyon, France.

Jangade, R. e Chauhan, R. (2006). Applications of machine learning in cancer prediction and prognosis. In *Departments of Biological Science and Computing Science*. Libertas Academica, Thousand Oaks, California.

Osareh, A. e Shadgar, B. (2010). Machine learning techniques to diagnose breast cancer. In *5th International Symposium on Health Informatics and Bioinformatics*. IEEE, Antalya, Turkey.

Chollet, F. (2015). *Keras. https://github.com/keras-team/keras*.

Gayathri, B. e Sumathi, C. (2016). Comparative study of relevance vector machine with various machine learning techniques used for detecting breast cancer. In *IEEE International Conference on Computational Intelligence and Computing Research*, pages 1–5. IEEE, Chennai, India.

Spanhol, F., Oliveira, L., Petitjean, C., e Heutte, L. (2016). A dataset for breast cancer histopathological image classification. In *IEEE Transactions on biomedical engineering (TBME)*, pages 1455–1462.

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., Kudlur, M., Levenberg, J., Monga, R., Moore, S., Murray, D., Steiner, B., Tucker, P., Vasudevan, V., Warden, P., Wicke, M., Yu, Y., e Zheng, X. (2016). Tensorflow: A system for large-scale machine learning. In *12th USENIX Symposium on Operating Systems Design and Implementation*. Usenix, Savannah, GA, USA.

Bray, F., Ferlay, J., Soerjomataram, I., Siegel, R., Torre, L., e Jemal, A. (2018). Global cancer statistics 2018: Globocan estimates of incidence and mortality worldwide for 36 cancers in 185 countries: Global cancer statistics 2018. In *CA: A Cancer Journal for Clinicians*.