International Journal of Healthcare Information Systems and Informatics
Volume 16 • Issue 4

Transfer Learning for Highlighting Diagnosis in Pathological Anatomy Based on Immunohistochemistry

Mohamed Gasmi, Larbi Tebessi University, Algeria*
https://orcid.org/0000-0001-5872-9142

Issam Bendib, Larbi Tebessi University, Algeria
https://orcid.org/0000-0001-9153-8161

Yasmina Benmabrouk, Larbi Tebessi University, Algeria

ABSTRACT

In the medical field, the diagnostic phase is the most important, as the entire treatment process will be based on this step. Oncological diseases such as breast cancer require a precise anatomopathological study accompanied most of the time by an immunohistochemical study whose goal is to know the sensitivity of tumor tissues to hormone therapy and targeted therapy. This study relies on antibodies and their interpretation requires significant time as it can suffer from poor reproducibility which negatively influences the treatment stage. In this work, the objective is to classify histopathological images stained with E-cadherin antibody to help pathologists in their work in order to facilitate oncologists in the choice of the most appropriate therapeutic protocol. The realization of this task is based on the choice of transfer learning as techniques and data augmentation due to the minimal number of images gathered. The results obtained are very satisfying on accuracy where they reached a rate of 97.27% with a reduced number of parameters and very close to the basic model.

KEYWORDS

Breast Cancer, CNN, Deep Learning, Immunohistochemistry, Pathological Anatomy, Transfer Learning

1 INTRODUCTION

Breast cancer is the most common malignant tumor in women and the most deadly. Breast cancer is a major public health problem in both industrialized and developing countries where its rate is increasingly growing year after year according to the 2020 statistics of the international agency for research on cancer[1]. Its diagnosis is always based on the histo-radio-clinical tripod and its management must absolutely be multidisciplinary.

Among the many different histological types that breast cancer involves, ductal and lobular carcinomas are distinguished and often oppose each other on different pathological criteria. Ductal and lobular carcinomas are the two most common histological types of invasive breast cancer. The ductal form dominates all histological types and accounts for more than half of all invasive carcinomas. The lobular form, second in frequency, accounts for only 5 to 10% of cancers.

DOI: 10.4018/IJHISI.301232 *Corresponding Author

Immunohistochemistry provided an appreciable comfort both to confirm the diagnosis of malignancy and to evaluate the loss of expression for E-cadherin for lobular carcinomas and positivity for other forms, notably ductal carcinomas.

However, traditional manual diagnosis requires an intense workload by competent experts. Misdiagnosis is likely to occur with pathologists who do not have sufficient diagnostic experience.

In this case, the need to evolve towards the use of computer-assisted-diagnosis (CAD) for the automatic classification of ductal and lobular cancers in order to improve the efficiency of the diagnosis will be noticed, but also physicians will be provided with results of more objective and accurate diagnostic results.

Deep learning techniques, such as convolutional neural networks (CNN), have shown great success in the detection of mitotic cells from histopathological images stained with hematoxylin and eosin (Ben Cheikh et al., 2017). This is a valuable source of inspiration for the development of an algorithm for classifying cancerous tissues from histopathological images colored with "E-cadherin".

The objective of this work is to create a classification model dedicated to anapathic slides of cancerous breast tissue treated with E-cadherin antibodies to classify ductal and lobular carcinomas to help the pathologist to specify the sensitivity of cancerous tissues to the antibodies used and on the other hand to facilitate the work of oncologists for appropriate management and monitoring of patients

The adopted approach to solve this problem is based on transfer learning, which is a deep learning technique that a model trained for one task is reused for another related task in.

While it is demonstrated that all models are working well in practice, it is not clear that they work well when modified and used to accommodate small datasets. There has been considerable interest in this issue.

Small-SE-ResNet (Yun et al., 2019) is the basic model for this work. It is a very deep learning model trained on a large dataset (BreakHis) (Fabio et al., 2016) and it gives a very good performance both on the accuracy and the reduced number of parameters.

The proposed approach is to adapt it with a small dataset based on simple modifications but appropriate for better performance. Identical to (Yun et al., 2019), the increase in data was used to expand the number of images to improve the learning operation.

Our work is more original compared to previous related works in terms of time and economy. Existing works try to help doctors either by automatic classification or by detection and segmentation, which can give a clearer view of the slide. The novelty of our work is to make an immunohistochemical classification without using antibodies. Which, of course, will allow us to save the cost of these antibodies and at the same time save the time of fixation that will in return influence the speed of diagnosis.

To clarify this study, the paper has been organized as follows:

After the introduction an overview was given on image classification, deep learning, transfer learning and some related work. Then methods and materials section describe the objectives and design of the study, also the dataset used in this work and how to increase it with data augmentation were presented. Afterwards the Section 5 shows the proposed transfer learning methods. The article ends with a discussion and validation of the results. Finally, Section 6 concludes this study.

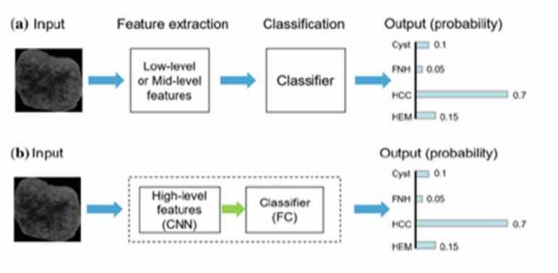
2 PRELIMINARY

2.1 Image Classification

Image classification consists of assigning one or more labels to an image. It is considered one of the most fundamental problems of computer vision and pattern recognition. In general, the image classification process consists of extracting features and then classifying them. Traditional classification methods use low-level or intermediate-level features to represent an image. Low-level features are typically based on human-defined density, color, texture, shape, and grayscale position

information (also known as hand-crafted features). Intermediate-level features, as well as learning-based features, are typically distilled by bag of visual word algorithms (BoVW) (Gong et al., 2013) (Gabriella et al., 2004) which have been effective and popular for image classification in recent years. In computer vision, after feature extraction, a classifier (e.g. SVM (Chih-Chung & Chih-Jen, 2011), random forest (Leo, 2001), etc.) is usually used to assign the label to different types of objects. The traditional classification of images is illustrated in (Figure 1.a).

Figure 1. Image classification. a Traditional classification method. b Deep learning method (Geert et al., 2017).



Unlike the traditional image classification method, the deep learning method combines the process of extracting and classifying image features on the same network. The classification process in deep learning is illustrated in (Figure 1.b).

2.2 Deep Learning

Several inventions were inspired by nature. It therefore seems logical to take inspiration from the architecture of the brain to build an intelligent machine. This is the key idea that triggered artificial neural networks, which are at the heart of deep learning, either by implementing supervised methods, as in (Alex et al., 2012) (Carneiro & Vasconcelos, 2005), or by implementing unsupervised methods, as in (Fergus et al., 2003) (Yuval et al., 2011).

CNN architectures stack a few convolutional layers (each usually followed by a ReLU layer), then a grouping layer, then a few other convolutional layers (+ ReLU), then another grouping layer, and so on. The image becomes smaller and smaller as it progresses through the network, but it usually gets deeper (i.e. with more feature maps) due to the convolution layers.

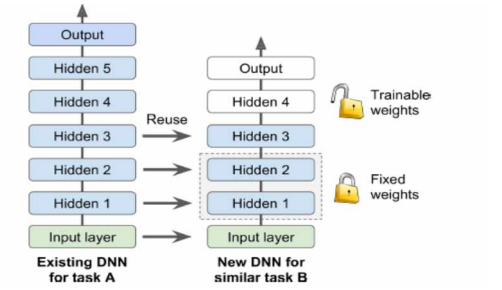
AlexNet from (2012), VGG net from Karen & Andrew, (2015), GoogleNet from Christian et al., (2016), ResNet from Kaiming et al., (2016), ResNeXt from Saining et al., (2017), se-resnet from Yun et al., (2019), small-se-resnet from Yun et al., (2019), RCNN (Region Based CNN) from Shaoqing et al., (2015), YOLO (You Only Look Once) from Joseph et al., (2016) are advanced models of deep learning. But in this paper, I focus mainly on Small-SE-ResNet.

2.3 Transfer Learning

Transfer learning is a technique by which a neural network model is first trained on a problem similar to the problem being solved. One or more layers of the trained model are then used in a new model trained on the problem of interest. It has been shown that transfer learning can work better in many cases with a smaller training data set Yen-Wei & Lakhmi, (2020).

(Figure 2) gives us an overview of the reuse of parts of the pre-trained network. The output layer of the original model usually has to be replaced because it is probably not useful at all for the new task, and may not even have the right number of outputs for the new task.

Figure 2. Reuse of the pre-entrained layer in deep learning methods Adrian, (2019).



The pre-trained model or the desired part of the model can be integrated directly into a new neural network model. In this use, the pre-trained model weights can be frozen so that they are not updated when learning the new model (the descending gradient does not change them). Then an attempt to unfreeze one or two of the top hidden layers to allow back-propagation to modify them and see if performance improves. The more training data you have, the more layers you can unfreeze. Adrian, (2019).

2.4 Related Work

Over the past decade, significant efforts have been made to recognize breast cancer (BC) from histological images based on deep learning. CNN variants are applied for this classification. Some of these experiments are conducted with the BreakHis dataset Fabio et al., (2016).

3. METHODS AND MATERIALS

3.1 Approach Overview

In practice there are many different approaches of Transfer Learning. Some techniques make the transfer by weighting and/or choosing certain data from the "source" set to introduce them into the "target" set: this is called an instance based transfer. Other approaches modify the "source" and/or "target" descriptor spaces to make them similar: this is called feature based transfer.Finally, still other methods modify the models themselves (whether neural networks, SVM, decision trees...) in order to transfer knowledge. in this case, it is a model-based transfer or parameter-based transfer.

The approach proposed in this work is based on the model based transfer. This is due to the small number of images available in the subject area. So we are going to take a high-performance model pre-trained on a large dataset with a very high precision, then we played in their layers by removing some of them, adding others, freezing some and training the rest.

concerning the data set we opted for the data increase because it is largely sufficient for our images (Either by rotation, horizontal flip, vertical flip, height shift range, width shift range ...). the thing that does not influence the content of these images.

3.2 Data Set

In this work, two datasets are central, the first one is an open source dataset used for the pre-trained model Fabio et al., (2016), and the second dataset is collected with the help of pathologists from the Pathological Anatomy Department of the CHU Constantine -Algeria- . This dataset will be used for the training of the model based on transfer learning.

It is a collection of histopathological images taken from actual samples of people with his own cancer. The samples are labeled with the antibody "E-cadherin". The author of the dataset uses breast tissue biopsy slides to generate these samples. The samples treated with "E-cadherin" are labeled in

another phase by the pathologists of the pathological anatomy laboratory of CHU Constantine, and are evaluated by immunohistochemistry.

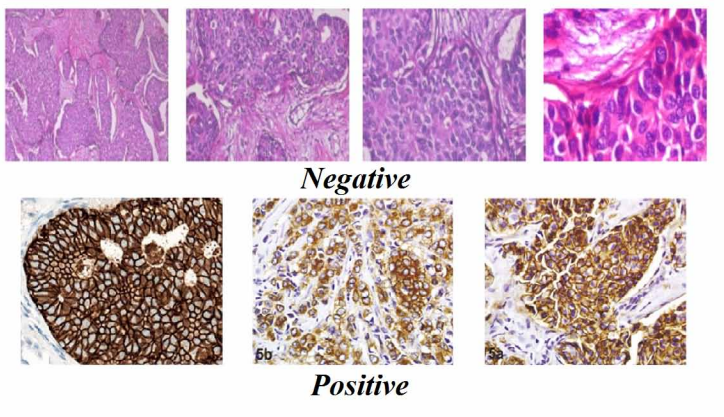
The collection of anapath images is done via a confocal laser scanning microscope, which is a fluorescence microscope whose light beam is generated by a laser. The transmitted signals are captured, digitized and saved as a jpeg image.

The image set contains 177 images, divided into Positive E-cadherin: 107 images et Negative E-cadherin: 70 images (Figure 3)

Table 1. The different related works

Reference	Architecture	Proposed
Fabio A (2017)	AlexNet	used AlexNet to extract the deep feature and combined different feature fusion strategies for breast cancer recognition. The performances of the proposed model are much better than the traditional ones.
Wei B Benzheng et al., (2017)	BiCNN	proposed a new method based on deep CNN (called BiCNN) to deal with the classification of pathological breast cancer images into two classes. This model considered breast cancer class and subclass labels as prior knowledge.
Hrushikesh G(2017)	GoogleNet	presented a classification model based on the GoogLeNet architecture for the diagnosis of cell samples using their high-magnification microscopic views.
Zhongyi H (2017)	GoogleNet	used GoogLeNet as a core network and proposed a method for multiple classification of breast cancer based on histopathological images. The structured model had performed remarkably well on a large-scale data set.
Shazia A (2017)	DNN	proposed a regularization technique called transition module, which captures filters at multiple scales and then reduces them by global average clustering to facilitate the reduction of the network size.
Weiming Z (2017)	VGGNet	used of transfer learning on convolutional neural networks (VGGNet and the personalized model) to diagnose breast cancer from histopathological images
Xianli Z (2019)	ResNet	provide a method for using the hop connection in Resnet to solve optimization problems when the network gets deeper.
Duc M (2019)	ResNet v2	apply augmentation techniques such as rotation, slicing, image transformation to increase the training data before extracting the deep function of the Inception-ResNet-v2.
Yun J (2019)	Small-SE-ResNet	proposed a new CNN network based on a small SE-ResNet module based on the combination of the residual module and SE block. Compared to the SE-ResNet module, the parameters of the small SE-ResNet module are reduced to 29.4%. The results show that this model achieves accuracy between 98.87% and 99.34% for binary classification and achieves accuracy between 90.66% and 93.81% for multi-class classification.

Figure 3. Positive & Negative E-cadhérine



3.3 Data Augmentation

Having a large data set is crucial for the performance of the deep learning model. However, improving the performance of the model can be done by increasing the collected data. One of the most widely used methods to address this problem is Data Augmentation (DA) - understanding the artificial increase in dataset size using image manipulation methods. This DA is achieved by performing operations that change the appearance of the image, without changing its semantics: for example, by performing a rotation.

The model requires 224 x 224 x 3 size input images, but the images used have different sizes. So resizing the learning images, increasing a data to enrich the dataset and improving the result are compulsory.

Figure 4. Generation of new images from existing ones

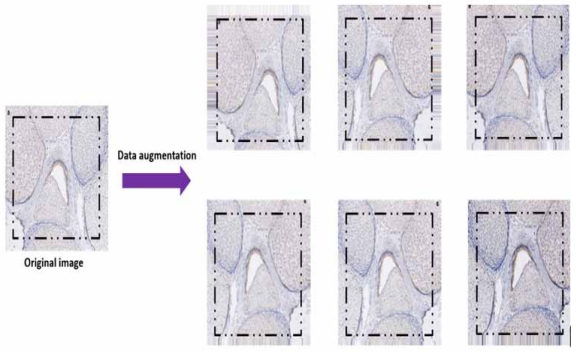


Table 2:Division of DataSet

	Dataset	
	Before Data Augmentation	**After Data Augmentation**
Train data	123	851
Test data	54	366
Total data	177	1217

The augmentation will be based on transformations to generate artificial samples for one or more classes of the dataset by rotating, zooming, mirroring, blurring on the original set of images.

The data augmentation is made due to the predefined class of Keras " ImageDataGenerator " which allows to apply geometrical transformations. Just browse them and choose the most relevant for the study. (Figure 4)

This Augmentation allowed us to increase the total number of images from 177 to 1217 images, and they are divided into two parts: 70% for training data and 30% for test data. (Table 2). Python "Scikit-Learn" offers a very handy function to split datasets: train_test_split.

3.4 The Pre-Trained Model

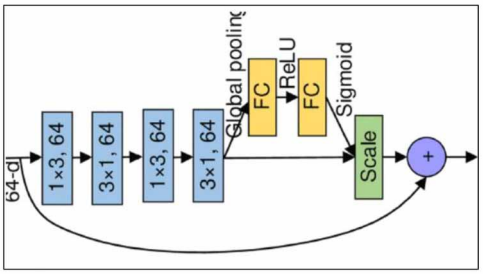
The pre-trained model used in this study is a combination of Squeezeand block-Excitation (SE block) and the residual block of ResNet.

An SE-ResNet module designed to reduce network parameters called the Small SE-ResNet module. It is an improvement of the combination of the residual module and the Squeeze-and-Excitation block, and achieves the same performance with fewer parameters Yun et al., (2019).

In the SE-ResNet module, there are two consecutive 1×3 and 3×1 convolutions with batch normalization and ReLU preceding the convolution and then combined with the SE block:

conv 1×3— conv 3×1— conv 1×3— conv 3×1 - bloc SE (Figure 5).

Figure 5. Small SE-ResNet module architecture Yun et al., (2019).



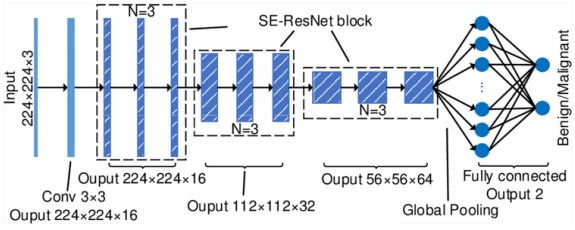
Only the total number of parameters in the convolutional layers is taken into consideration. The total number of parameters for a convolutional layer is:

T(conv)=kernel size * * kernel number = (C*H*W)*K Yun et al., (2019).

Where $C \times H \times W$ is the size of the kernel and K is the number of kernels. Then, the number of parameters for the Small SE-ResNet module could be obtained by the following formula:

T(Small)= (64 X 1 X 3 X 64) X 2+(64 X 3 X 1 X 64) X 2=12 X 64^2

Figure 6. The BHCNet-3 architecture for the benign and malignant classification of histopathological images of cancer of the breast. Yun et al., (2019)



The BHCNet "(breast cancer histopathology image classification network)" is a new CNN architecture for the classification of breast cancer histopathology images using the small SE-ResNet module. It consists of a single convolutional layer, three SE-ResNet blocks and a fully connected layer. Each SE-ResNet block is stacked by N small SE-ResNet modules, designated by BHCNet- N. When N= 3, The BHCNet architecture is shown in (Figure 6) Yun et al., (2019).

This model has been tested on the BreakHis dataset for the binary classification of breast cancers (benign and malignant) with competitive experimental results (accuracy between 98.87% and 99.34% for the binary classification) Yun et al., (2019).

4 TRANSFER MODEL

4.1 Classification With a Pre-Trained CNN Model

The first thing to do before starting to use transfer learning techniques on the pre-trained model, whether by removing or adding new layers or by training or fixing other layers, is to train the model even with the new database.

To do this, the BHCNet model using Keras' class "load_model" is loaded before starting the learning without any modification with the use of stochastic gradient descent as an optimization algorithm to correct the predictions and guide the network to precise weights. Table 3 describes the results obtained for the two datasets used.

Table 3. Accruacy for classification with the BHCNet model without any modification

DataSet	Accuracy
DataSet (BreakHis)	97.87%
DataSet	87.70%

It is obvious that the accuracy obtained for dataset is less accurate compared to the rate obtained for the BreakHis dataset. So to try to optimize the accuracy, the first thing to do is to play with the training options which are: batch_size and the number of epochs.

- The batch_size: is a gradient descent hyper-parameter that controls the number of training samples to be processed before updating the internal parameters of the model.
- Epochs: The number of epochs is a hyper parameter of gradient descent that controls the number of complete runs in the training dataset.

The first time, 32 batches and 100 epochs are used, the accuracy obtained in this case is higher than the rate obtained previously, but in the case of increasing the number of batch_size and the number of epochs, the rate is decreased.

Table 4. Accuracy by changing batch size and epochs

	Batch_size	Epochs	Batch_size	Epochs	Batch_size	Epochs
	32	**100**	**64**	**100**	**128**	**200**
Accuracy	88,52%		87,90%		87,70%	

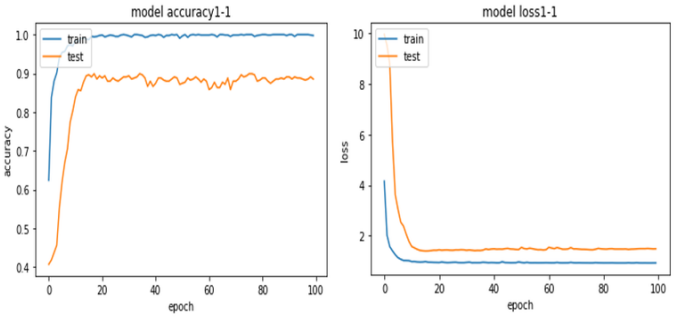
the results obtained by changing the batch size and the number of epochs are shown in the following table 4

In order to interpret the results correctly, it is practical to use graphs that show the learning metrics and their automatic update after each epoch whereas the "Matplotlib" library offers many different tools to help us in this visualization process.

(Figure 7) shows the two graphs of accuracy and loss which allows us to see that the accuracy during learning is very high and the loss is very minimal, but during the test the results do not resemble the learning results.

These two graphs show that the accuracy and loss is acceptable, so the training of all the layers of the model do not need to be done again.

Figure 7. Accuracy and loss graph for scenario 0

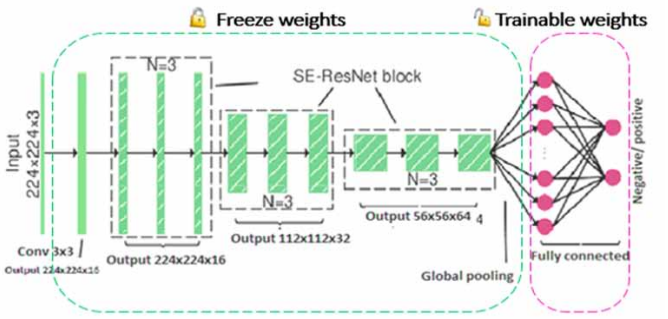


To train certain layers of the pre-trained model, transfer learning is a reliable way, which will allow us to reuse the pre-trained model to adjust its parameters to adapt it to dataset with the aim of increasing the accuracy rate and reducing the loss.

To determine whether the use of transfer learning makes the pre-trained network more effective for data, different types of experiments are conducted:

- Fix the weights of all layers of the pre-trained model, and train the last layer (dense_2).
- Fix the weights of all upper layers of the pre-trained model, and replace the last two layers of the model (dense_2, flatten_2) by new trainable layers.
- Fix one set of top layers, and train the rest of the model.

Figure 8. The transfer model architecture for scenario 1



4.2 Scenario 1

The pre-trained model contains 219 layers initialized by weights optimized for the BreakHis dataset. To preserve and take advantage of the parameters and weights of the pre-trained model, the first idea was to freeze the three SE-ResNet blocks by keeping the initial weights of the pre-trained model and applying the training only on the fully connected layer as shown in (Figure 8).

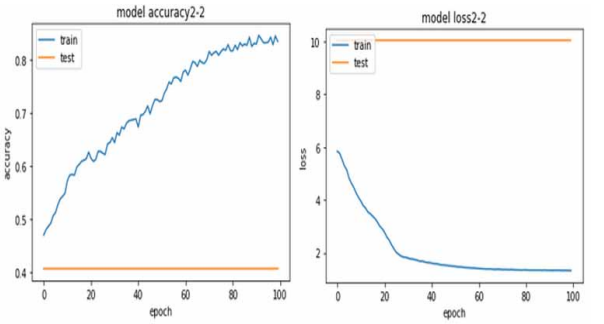
After applying transfer learning by fixing the three SE-ResNet blocks and training the fully connected layer, learning by several values of the batch-size and by different number of epochs started. The results obtained are shown in Table 5.

The first remark extracted from this table is that the different batch size values and the number of epochs have no effect on accuracy. It is also noticable that the accuracy rate is less than the results of the scenario 0.

Table 5. Accuracy for different batch_size and epochs for scenario 1

	Batch_size	**Epochs**	**Batch_size**	**Epochs**	**Batch_size**	**Epochs**
	32	**100**	**128**	**100**	**16**	**50**
Accuracy	40.71%		40.71%		40.71%	

Figure 9. Graphs of accuracy and loss of scenario 1 (batch_size=128, epochs=100).



For a better understanding of the results obtained, the learning and testing process through precision and loss graphs are demonstrated (Figure 9).

From this graph, the model is under-adjusted and this can be seen through the minimal rate of accuracy. In this case, this is due to an offset between the weights obtained from the pre-trained model for the 3 SE-ResNet blocks and the re-trained fully connected layer.

The following code shows the instructions used to freeze all layers except the fully connected layer.

```
For layer in new_model.layers[:-1]:
Layer.trainable = False
Newmodel.compile(optimizer='SGB',loss='binary_crossentropy',metrics=['accuracy'])
```

4.3 Scenario 2

The weak results of the first scenario led us to think about introducing a modification in the structure of the pre-trained model. The idea is to remove some layers and add others and then play on the choice of fixation and training of the layers.

From the results of scenario 0 and scenario 1, the addition of several fully connected layers makes the transfer model more efficient.

a. The first idea is to change the architecture of the last two layers of the pre-trained model. To do this, these two layers are removed and replaced by a new "flatten layer" and two new fully connected layers with two neurons for each (Figure 10).

Figure 10. Changing the last two layers of the BHCNet model

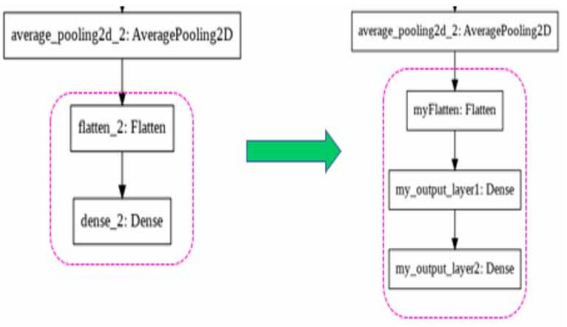
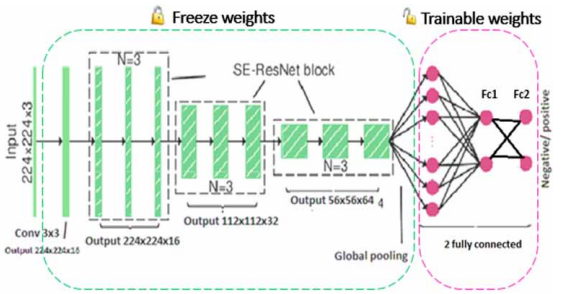


Figure 11. transfer learning Architecture for scenario 2.a



The architecture of the new model after the new modifications is shown in the following Figure 11:

There are several methods to remove and add layers to a model, the output of the layer "average_pooling2d_2" is obtained and then added to a flattening vector "myFlatten" contains 512 neurons and two fully connected layers "my_output_layer1" and "my_output_layer2" respectively contains two neurons each.

There are several methods to remove and add layers to a model, we got the output of the layer "average_pooling2d_2" then we added a flattening vector "myFlatten" contains 512 neurons and two fully connected layers "my_output_layer1" and "my_output_layer2" respectively contains each of them two neurons. The instructions are expressed below:

Last_layer=new_model.get_layer('average_pooling2d_2').output x=Flatten(name='myFlatten') (last layer) out=Dense(2,activation="softmax",name='my_output_layerl') (x) out2=Deose(2,activation="softmax",name='my_output_layer2') (out)

My_model=Model(inputs=input_sbape,outputs=out2,name="Scenario2_Model")

The new model is tried several times by changing the number of epochs and the size of the batch. The rate increases each time the number of epochs and the batch_size is levitated. Table 6 shows the different accuracy results obtained. a rate of 86.61% for 100 epochs with 64 batch size is reached.

The graph in (Figure 12) shows that the rate of accuracy during the learning phase is very high but unstable during the testing phase. Therefore, the model could not manage to assign the ideal weights. This interpretation leads us to think about expanding the training layers.

Table 6. Accuracy for different batch size and epochs for scenario 2.a

	Batch_size	Epochs	Batch_size	Epochs	Batch_size	Epochs
	32	**50**	**32**	**100**	**64**	**100**
Accuracy	85,52%		84,15%		86,61%	

Figure 12. Accuracy and loss graph for scenario 2.a

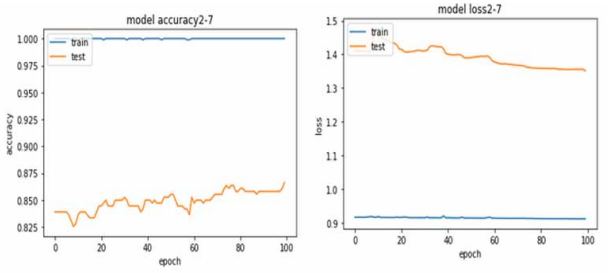
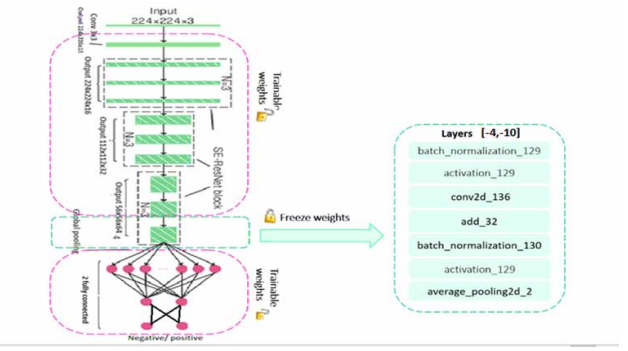


Figure 13. Transfer learning Architecture for scenario 2.b



b. In order to further increase the accuracy rate expressed in Table 6, the layers that will undergo training have been enlarged. For this purpose, training the third module of the third block of the pre-trained model plus the fully connected layers of the scenario 2.a. is determined.

By applying a training set by changing the number of epochs and the batch size, acceptable results are obtained but not satisfactory compared to the results of the previous scenario.

The best results obtained when applying this method are listed in Table 7 with the number of epochs and the batch size.

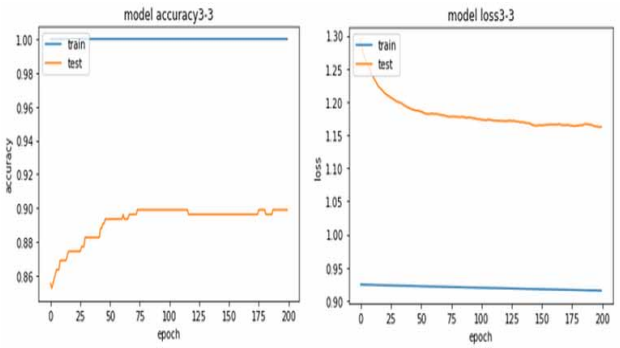
Table 7. Accuracy for different batch_size and epochs for scenario 2.b

	Batch_size	Epochs	Batch_size	Epochs	Batch_size	Epochs
	32	**100**	**64**	**100**	**128**	**200**
Accuracy	89,07%		89,17%		89,89%	

What can be seen in (Figure 14) is the same as in (Figure 12) where the precision is very important during the training phase while during the test phase it is very modest.

What can be learnt from this graph is that training other layers than the fully connected layer cannot give us satisfactory results.

Figure 14. Accuracy and loss graph for scenario 2.b



4.4 Scenario 3

The observation obtained in scenario 2 has oriented us to work only with the fully connected block. so the objective is to freeze all the other blocks and train only the fully connected layers with updates of this block.

To do this, new fully connected layer formed by two neurons to the model proposed in scenario 2 is added. So the new model is composed of 3 SE-ResNet blocks of a flaten layer and 3 dense layers each one is formed with two neurons.

The training will be started by freezing the three SE-ResNet blocks and training only the fully connected layers. As previously it started with the training by playing with the number of epochs and the batch size, the results obtained are mentioned in table 8.

Comparing the results with the results of scenario 2.b and compared to the results of scenario 2.a a very perceptible improvement but not satisfactory is obvious.

Comparing graph (Figure 17) with the other precision and loss graphs above, a very satisfactory improvement in the test phase is shown. This remarkable improvement shows the accuracy of the

path and the robustness of the strategy that was selected to improve the accuracy of the model and reduce the loss.

Figure 15. Transfer learning Architecture for scenario 3

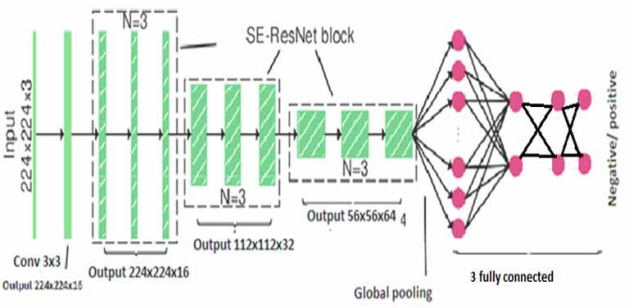
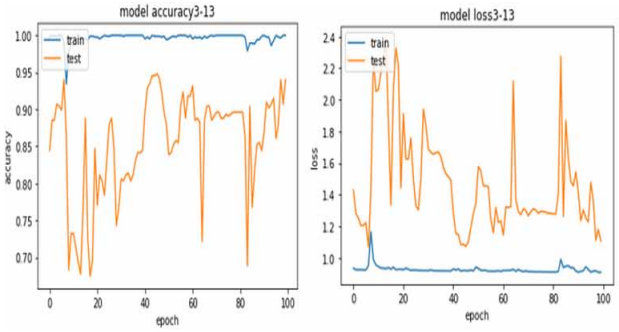


Table 8. Accuracy for different batch_size and epochs for scenario 3

	Batch_size	Epochs	Batch_size	Epochs	Batch_size	Epochs	Batch_size	Epochs
	32	**50**	**32**	**100**	**64**	**100**	**128**	**200**
Accuracy	89,07%		89,17%		89,89%		90.43%	

Figure 16. Accuracy and loss graph for scenario 3



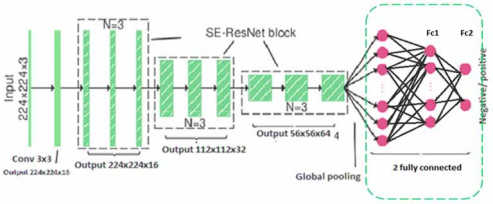
4.5 Scenario 4

The steps followed in this scenario are the result of what has been followed in the previous scenarios, as it is noticed and concluded that the improvement in accuracy and the reduction in losses are mainly due to the fully connected layer.

So, what is going to be done in this scenario is to add a second fully connected layer to the pre-trained model by playing with the number of neurons. Now the fully connected module consists of a flatten layer and 2 dense layers fc1 and fc2, one of which is an output layer (Figure 18).

the idea of this scenario is to play with the number of neurons of the fc1 layer in order to improve the accuracy rate.

Figure 17. Transfer learning Architecture for scenario 4



3 cases to treat and experiment it according to the change in the number of neurons are chosen. These cases were as follows:

- Case 01: the two fully connected layers: “fc1” made up of 256 and “fc2” made up of 2 neurons.
- Case 02: the two fully connected layers: “fc1” made up of 128 and “fc2” made up of 2 neurons.
- Case 03: the two fully connected layers: “fc1” made up of 64 and “fc2” made up of 2 neurons.

3 models are trained with batch_size =32 and epoch=100, as these are the most responsive values that gave us a good accuracy rate in the previous scenario s.

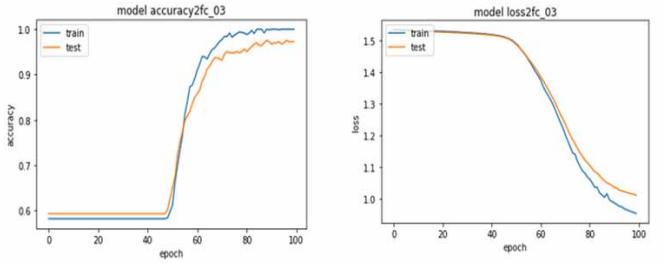
After a number of training iterations,the accuracy rates are obtained as mentioned in Table 9.

Table 9. Accuracy by changing the number of neurons in the fully connected layers.

	Taux d’apprentissage		
	Case 01	**Case 02**	**Case 03**
Accuracy	95.90%	97.27%	95.35%

According to this table, the best accuracy rate is obtained in the model of case 02 which gave an accuracy rate of 97.27% with 100 epochs and batche_size 32. (Figure 19) shows the precision and loss graphs of the model of case 02. The accuracy was stable on the value 59.29% until epoch 49 when the value increased to 60.11% and it continued to increase until the accuracy of 97.27%. The loss graph confirms the accuracy rate. The other thing that confirms the quality of this model is the very similarity that appears between the training graph and the test graph in both accuracy and loss.

Figure 18. Accuracy and loss graph of the case 02 model of scenario 4.



5 DISCUSSION AND VALIDATION

In this work, the pre-trained BHCNet model is used, which is characterized by a very high accuracy rate and a very low number of parameters despite the number of layers used. The work consists of freezing, training the layers of the BHCNet model, also removing and adding layers in order to classify the dataset into positive E-cadherin and negative E-cadherin. The parameters of BHCNet have been initialized in pre-trained weights optimized for the BreakHis dataset with an accuracy rate of almost 98%. Initially the model is used as it is, where it gives an accuracy rate of 88.52%. To better adapt the model to the dataset used in transfer learning.

5.1 Discussion of Different Scenarios

To evaluate our Deep-learning cyberattack detection model, we opted for the typical metric used in deep learning as performance measures, namely: accuracy.

- *Accuracy (ACC):* indicates the ratio of correct detection over total traffic trace:

$$ACC = \frac{1}{M}\sum_{i=1}^{M}\frac{TP_i + TN_i}{TP_i + TN_i + FP_i + FN_i}$$

The first scenario is to freeze all 3 SE-ResNet blocks in order to keep the same weight of these 3 blocks by driving only the fully connected layer and an accuracy of 40.71% is attained, which is very low compared to the initial model rate.

So " scenario 2.a" with the same principle of freezing the weights of the three blocks is attempted, and replacing the last two layers of the model (dense_2, flatten_2) by another layer of flatten and two other fully connected drivable layers. The rate obtained from this scenario is equal to 86.61% which is considered a remarkable increase in rate.

For more improvement, the " scenario 2.b" with the principle role to train more layers is conducted, more precisely to train the 3rd module of the 3rd block by freezing the other modules. The rate obtained is (89.89%) after further modification in the fitting options.

scenario 3 consists in adding a third layer and applying the principle of scenario 2.b to determine whether the addition of several fully connected layers makes the transfer model more efficient or not. The result is that the accuracy rate to be increased up to 93.99%.

In scenario 4, first all modules from all blocks of the pre-trained model are frozen. Then, an addition of a (dense) layer to the fully connected module is made before the output layer with 128 neurons. The result obtained is very satisfactory with an accuracy of 97.27%.

The following table 10 summarizes the best results of each scenario:

The results obtained show that the choice of the epoch number and the batch_size as well as the depth of the network have a great influence to have better results. It is noticed that the best precision rate is the one of scenario 4. It is due to the increase of the number of neurons of the fully connected layer.

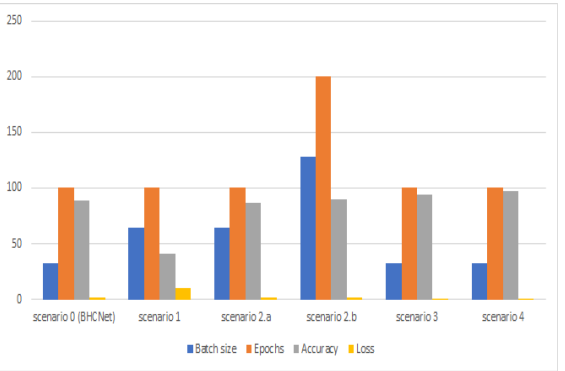
It is also perceived that the number of parameters in the first three scenarios is equal to 14832770 (the same number of parameters of the pre-trained model) parameters because nothing has been modified at the level of the pre-trained model. The minimum number of parameters is obtained in scenario 4 with 14837766 parameters thanks to the choice used in the layers and the number of neurons in the fully connected layers.

Table 10. Les meilleurs taux de précision obtenus de chaque méthode

scenario	Batch size	Epochs	Accuracy	Loss	total parameters	Description
scenario 0 (BHCNet)	32	100	88.52%	1.48	14832770	Toutes les couches sont entrainées
scenario 1	64	100	40.71%	10.023	14832770	Entrainer juste la couche FC
scenario 2.a	64	100	86.61%	1.3511	14832770	Entrainer les deux couches FC
scenario 2.b	128	200	89.89%	1.1624	14832776	Fixer les couches entre [-4,-10]
scenario 3	32	100	93.99%	1.1142	14832782	Ajout de 3eme FC et fixer les couches entre [-4,-10]
scenario 4	32	100	97.27%	1.0101	14837766	Changer le nombre de neurones couche fc1 (128)

Then, the rate closest to the rate obtained from the pre-trained model (with the BreakHis database) is that of scenario 4 with a satisfactory number of parameters and close to the basic model.

Figure 19. Comparison of different scenarios



The graph above shows that the number of epochs and the batch size does not directly influence the accuracy for the convergence of the results, while the choice of the layers to freeze and the layers to train in addition to the added dense layers. All this shows the value brought by transfer learning to create models from pre-trained models while keeping almost the same precision.

A better performance is better estimated if a good number of microscopic images of the anapathis slides stained by E-cadherin antibodies is collected.

5.2 Evaluating Model

In order to validate the model, first the changes in accuracy and loss model during the training process were visualized, as this gave us important information to evaluate what can be done to improve

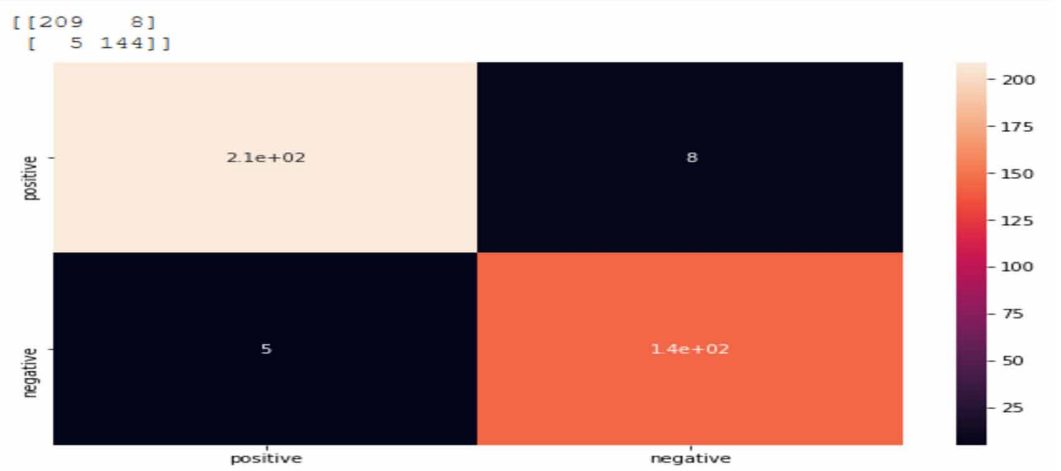
accuracy. To better evaluate the proposed model, first "confusion matrix" is calculated, then making a prediction on new images with the help of a medical expert followed.

5.2.1 Evaluation with the Confusion Matrix

The confusion matrix is an illustrative way to evaluate the accuracy of the model, as it compares the "true" class and the "predicted" class for all images in the test set.

The best model, among the proposed scenarios, achieved an average accuracy of 97.27%. The result of the confusion matrix for this best architecture can be found in (Figure 21).

Figure 20. Confusion matrix for best scenario



- Every observation in the testing set is represented in exactly one box
- It's a 2x2 matrix because there are 2 response classes

Interpretation

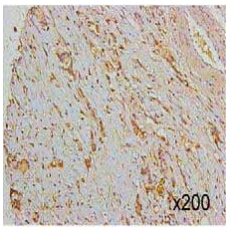
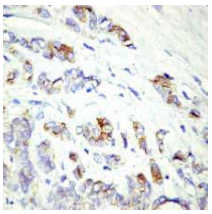
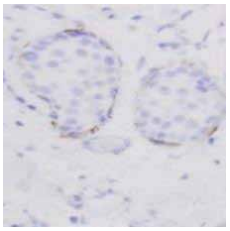
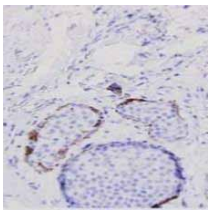
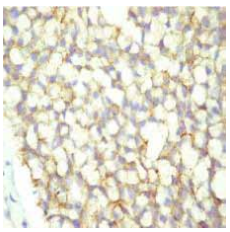
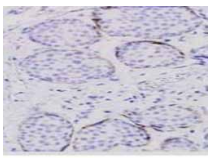
True Positives (TP): 209 images sensitive to immunohistochemistry are correctly predicted.
True Negatives (TN): correct prediction that the images are not sensitive to immuno are 8.
False Positives (FP): wrong prediction that images that are immuno-sensitive are 5.
False Negatives (FN): erroneous prediction that images that are not immune-sensitive are 144.

5.2.2 Evaluation Against the View of Pathologists

To give more power to the proposed model, a comparison between the prediction results of the model and the opinions of pathology experts is made. So the images are loaded and resized to the size used during training and testing of 224 × 224 pixels. Then the model was loaded and the prediction was made by calling the "Predict" function.

Below, Table 11 illustrates some examples of anatomical pathology slides tested by the proposed model in relation to the view of pathologist

Table 11. Comparison between the result of the model and the expert's opinion

Microscopic slide	Predict	Expert	Microscopic slide	Predict	Expert
	Positive (91.76%)	Positive		Negative (82.38%)	Negative
	Negative (88.68%)	Negative		Negative (88.61%)	Negative
	Positive (91.88%)	Positive		Negative (84.74%)	Negative

6 CONCLUSION

Immunohistochemistry is a technic widely used to test the sensitivity of tumor cells such as breast cancer to hormone therapies. This study requires the use of antibodies. The interpretation of the influence of these antibodies on the tissue is rendered by a pathologist. This interpretation takes time and suffers from poor reproducibility even in the hands of expert pathologists.

It is within this framework that the work is carried out, which aims to classify histopathological images colored with the E-cadherin antibody to help pathologists in their work to specify the sensitivity of tissues to hormone therapies, in order to facilitate the work of oncologists for the use of the most appropriate treatment.

To carry out the classification work, deep learning is used and convolutional neural networks and transfer learning are selected as learning techniques that consequently show a great success in the detection of mitotic cells from histopathological images.

Obtaining the classification results requires a large amount of data (images) the thing that got us into trouble. To solve this problem of lack of learning data, the technique of "data augmentation" and transfer learning is selected so as to create a classification model dedicated to anapathic slides of breast cancer tissues treated with E-cadherin antibodies. The model is created from a model achieving better prediction and parameter performance, pre-trained on a large dataset of real histopathological images (BreakHis). The new model is in turn re-trained on the dataset of histopathological images stained by the Ecadherin antibody.

The results obtained are very satisfying both on the accuracy rate where a rate of 97.27% is attained with a very reduced number of parameters and very close to the basic SE-Small-ResNet model.

The perspective of the work is to be able to interpret the sensitivity of cancerous tissues to hormone therapies using basic anapath slides without the use of antibodies, by applying other methods to increase the data. The limitation of this work is that the model and dataset concern only one organ, which is the breast and also only one antibody, which is E-cadherin. Future work could include the study of cancer within other organs that can be treated with both; hormone therapy and targeted therapy. We give much attention to other antibodies, each and their own intended organ. Future goals are to save time that will help patients to start their treatments as soon as possible and also allows the state to avoid wasting money on antibody purchases.

FUNDING AGENCY

Publisher has waived the Open Access publishing fee.

REFERENCES

Adrian, R. (2019, May). *Transfer Learning with Keras and Deep Learning*. Machine Learning Mastery.

Alex, K., Ilya, S., & Geoffrey, E. (2012, January). ImageNet Classification with Deep Convolutional Neural Networks. *Advances in Neural Information Processing Systems*.

Ben Cheikh, B., Bor-Angelier, C., & Racoceanu, , D. (2017, March). A model of tumor architecture and spatial interactions with tumor microenvironment in breast carcinoma. *SPIE Medical Imaging Conference*.

Benzheng, W., Zhongyi, H., Xueying, H., & Yilong, Y. (2017, April). Deep learning model based breast cancer histopathological image classification. *IEEE 2nd International Conference on Cloud Computing and Big Data Analysis (ICCCBDA).* doi:10.1109/ICCCBDA.2017.7951937

Carneiro, G., & Vasconcelos, N. (2005). Formulating semantic image annotation as a supervised learning problem. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. doi:10.1109/CVPR.2005.164

Chih-Chung, C., & Chih-Jen, L. (2011, May). LIBSVM- A Library for Support Vector Machines. *ACM Transactions on Intelligent Systems and Technology*.

Christian, S., Vincent, V., Sergey, I., Jon, S., & Zbigniew, W. (2016, June). Rethinking the Inception Architecture for Computer Vision. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Duc, M.V., Ngoc, Q.N., & Sang, W.L. (2019, May). Classification of breast cancer histology images using incremental boosting convolution networks. In *Information Sciences*. Elsevier.

Fabio, A. S., Luiz, S. O., Caroline, P., & Laurent, H. (2016, July). A Dataset for Breast Cancer Histopathological Image Classification. *IEEE Transactions on Biomedical Engineering*, *63*(7). PMID:26540668

Fabio, A. S., Luiz, S. O., Paulo, R. C., Caroline, P., & Laurent, H. (2017, October). Deep features for breast cancer histopathological image classification. *IEEE International Conference on Systems, Man and Cybernetics*.

Fergus, R., Perona, P., & Zisserman, A. (2003). Object class recognition by unsupervised scale-invariant learning. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. doi:10.1109/CVPR.2003.1211479

Gabriella, C., Christopher, R. D., Lixin, F., Jutta, W., & Cédric, B. (2004). Visual Categorization with Bags of Keypoints. In *Workshop on Statistical Learning in Computer Vision*. ECCV.

Geert, L., Thijs, K., Babak, E.B., Arnaud, A.A.S., Francesco, C., & Mohsen, G. (2017, July). A Survey on Deep Learning in Medical Image Analysis. In *Medical image analysis*. Elsevier.

Gong, C., Lei, G., Tianyun, Z., Junwei, H., Huihui, L., & Jun, F. (2013, January). Automatic landslide detection from remote-sensing imagery using a scene classification method based on BoVW and pLSA. *International Journal of Remote Sensing*.

Hrushikesh, G., Kari, S. P. K., Debdoot, S., Jyotirmoy, C., Manjunatha, M., & Ajoy, K. R. (2017, August). High-Magnification Multi-views Based Classification of Breast Fine Needle Aspiration Cytology Cell Samples Using Fusion of Decisions from Deep Convolutional Networks. *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW 17)*.

Joseph, R., Santosh, D., Ross, G., & Farhadi, A. (2016, June). You Only Look Once: Unified. *Real-Time Object Detection. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Kaiming, H., Xiangyu, Z., Shaoqing, R., & Jian, S. (2016, June). Deep Residual Learning for Image Recognition. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Karen, S., & Andrew, Z. (2015). Very Deep Convolutional Networks for Large-Scale Image Recognition. *International Conference on Learning Representations*.

Leo, B. (2001, May). *Machine Learning*. Kluwer Academic Publishers.

Saining, X., Ross, G., Piotr, D., Zhuowen, T., & Kaiming, H. (2017, November). Aggregated Residual Transformations for Deep Neural Networks. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Shaoqing, R., Kaiming, H., Ross, G., & Jian, S. (2015, June). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Shazia, A., Mohammad, P., Sherine, S., Sharon, N. M., & Anne, M. (2017, September). Transitioning between Convolutional and Fully Connected Layers in Neural Networks. *International Workshop on Deep Learning in Medical Image Analysis International Workshop on Multimodal Learning for Clinical Decision Support.*

Weiming, Z., Henry, W. F. Y., Zhenghao, C., & Seid Miad, Z. (2017, November). Using Transfer Learning with Convolutional Neural Networks to Diagnose Breast Cancer from Histopathological Images. *The 24th International Conference On Neural Information Processing*.

Xianli, Z, Yinbin, Z., Buyue, Q., Xiaotong, L., Xiaoyu, L., & Xudong, W. (2019, April). Classifying Breast Cancer Histopathological Images Using a Robust Artificial Neural Network Architecture. *International Work-Conference on Bioinformatics and Biomedical Engineering, IWBBIO'19*.

Yen-Wei, C., & Lakhmi, C.J. (2020, January). *Deep Learning in Healthcare: Paradigms and Applications*. Intelligent Systems Reference Library.

Yun, J., Li, C., Hai, Z., & Xiao, X. (2019, March). Breast cancer histopathological image classification using convolutional neural networks with small SE-ResNet module. *PLoS One*. PMID:30925170

Yuval, N., Tao, W., Adam, C., Alessandro, B., Bo, W., & Andrew, Y. (2011). Reading Digits in Natural Images with Unsupervised Feature Learning. *Computer Science*.

Zhongyi, H., Benzheng, W., Yuanjie, Z., Yilong, Y., Kejian, L., & Shuo, L. (2017, June). Breast Cancer Multi-classification from Histopathological Images with Structured Deep Learning Model. *Scientific Reports*. PMID:28646155

ENDNOTE

[1] https://www.iarc.who.int

Mohamed Gasmi received the B.S. and M.S. degrees in Computer Science from the University of Tebessa, Algeria, in 2009 and 2011, respectively. Received the PhD degrees from the University of M'sila in 2017. His research interests are web semantic, description logic, ontologies, and deep learning. Temporary teacher at Tebessa University for the academic year 2017/2018. Currently, Professor Researcher Holder at the Faculty of Science and Technology at the University of Tebessa since December 2018.

Issam Bendib is an associate professor in the department of Mathematics and Computer Science at Larbi Tébessi University and a member of the Laboratory of Mathematics, Informatics and Systems (LAMIS). He has received his Engineering degree in computer sciences from University of Badji Mokhtar-Annaba, Algeria in 1998. He received his PhD from the University of Badji Mokhtar-Annaba in 2018. His research interests are related to speech, image and pattern recognition, semantic indexing, deep learning, and biometrics systems. He has published research papers in national and international journals, conference proceedings.