ENSIAS

جامعة محمد الخامس بالرباط
Université Mohammed V de Rabat

MÉMOIRE DE PROJET DE FIN D'ANNÉE

Breast Cancer With Deep Leaning

Réalisé par :

Derrouich Jaouhar

Outtaleb Mohamed

Encadré par

Mr. Tabii Youness

ANNÉE UNIVERSITAIRE 2020 – 2021

Remerciement

Au terme de ce travail, nous tenons à exprimer notre profonde gratitude à notre chère professeur et encadrant Monsieur Tabii Youness, nous le remercions de nous avoir orientés, aidés et conseillés, nous tenons à vous remercier pour les opportunités du développement intéressant que vous nous avez offertes et qui nous ont fait évoluer professionnellement mais aussi personnellement.

Nous souhaitons remercier également tous les professeurs de l'ENSIAS, bien que les membres de l'équipe pour leurs efforts, leurs engagements, et leurs patiences afin d'accomplir ce projet.

Enfin, nous ne pouvons achever ce projet sans exprimer notre reconnaissance à nos chers parents, nos frères, nos sœurs qui ont toujours été là pour nous, à nos amis pour leur sincère amitié et confiance, et à tous ceux qui ont contribué de près ou de loin à l'élaboration de ce projet.

Résumé

Le cancer du sein est la forme de cancer la plus courante chez les femmes, et le Carcinome Canalaire Invasif (CCI) est la forme la plus courante de cancer du sein. Identifier et catégoriser avec précision les sous-types de cancer du sein est une tâche clinique importante, et des méthodes automatisées peuvent être utilisées pour gagner du temps et réduire les erreurs.

Le but de ce script est d'identifier l'CCI lorsqu'il est présent dans des images histopathologiques autrement non étiquetées.

Table des matière

Table des figures

1. Dataset

Le Carcinome Canalaire Invasif (CCI) est l'un des plus courants sous-type de cancer du sein. Afin d'identifier les cancers du sein, les pathologistes se concentrent généralement sur les régions contenant le CCI.

Ici, nous utilisons des images de diapositives de montage du cancer du sein (BCa) spécimens numérisés à 40x.

Les captures d'écran de tissus sont sélectionnées dans Kaggle Breast Cancer dataset de classification. Il contient différents dossiers nommés avec identifiants des patients. Dans chaque dossier, nous avons 2 dossiers d'images avec étiquette 1 ou 0 représentant "les zones de capture d'écran ont un cancer" ou « les zones de capture d'écran n'ont pas de cancer » respectivement. Le dataset est recueilli auprès d'un certain nombre de patients atteints de cancer. Les images de pathologie contenant CCI sont étiquetées comme 1 et celles non contenant CCI sont étiquetés comme0.

```
$ tree --dirsfirst -L 4
.
├── datasets
│   └── orig
│       ├── 10253
│       │   ├── 0
│       │   └── 1
│       ├── 10254
│       │   ├── 0
│       │   └── 1
│       ├── 10255
│       │   ├── 0
│       │   └── 1
...[omitting similar folders]
│       ├── 9381
│       │   ├── 0
│       │   └── 1
│       ├── 9382
│       │   ├── 0
│       │   └── 1
│       ├── 9383
│       │   ├── 0
│       │   └── 1
```

Figure 1 dataset tree

Comme vous pouvez le voir, notre dataset se trouve dans le dossier datasets/orig et est ensuite ventilé par ID de patient. Ces images sont séparées en répertoires bénins (0/) ou malins (1/).

Figure 2 quelques échantillons d'image de pathologie sans cancer CCI

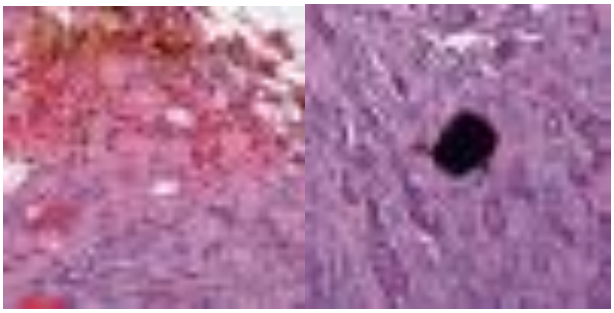

Figure 3 quelques échantillons d'image de pathologie avec cancer CCI

2. Data preparation

Dans ce projet, l'idée centrale est de former le modèle d'apprentissage automatique avec des images étiquetées pour obtenir un diagnostic de cancer pour le patient. Nous avons appliqué plusieurs étapes de prétraitement d'image et de machine algorithme d'apprentissage pour construire, entraîner et tester le modèle.

Pour extraire les informations des images avec précision, nous appliquons également un algorithme de traitement d'image tel que ‘histogram equalization’ pour équilibrer l'intensité des images et s'assurer qu'elles sont cohérentes pour le modèle de formation.

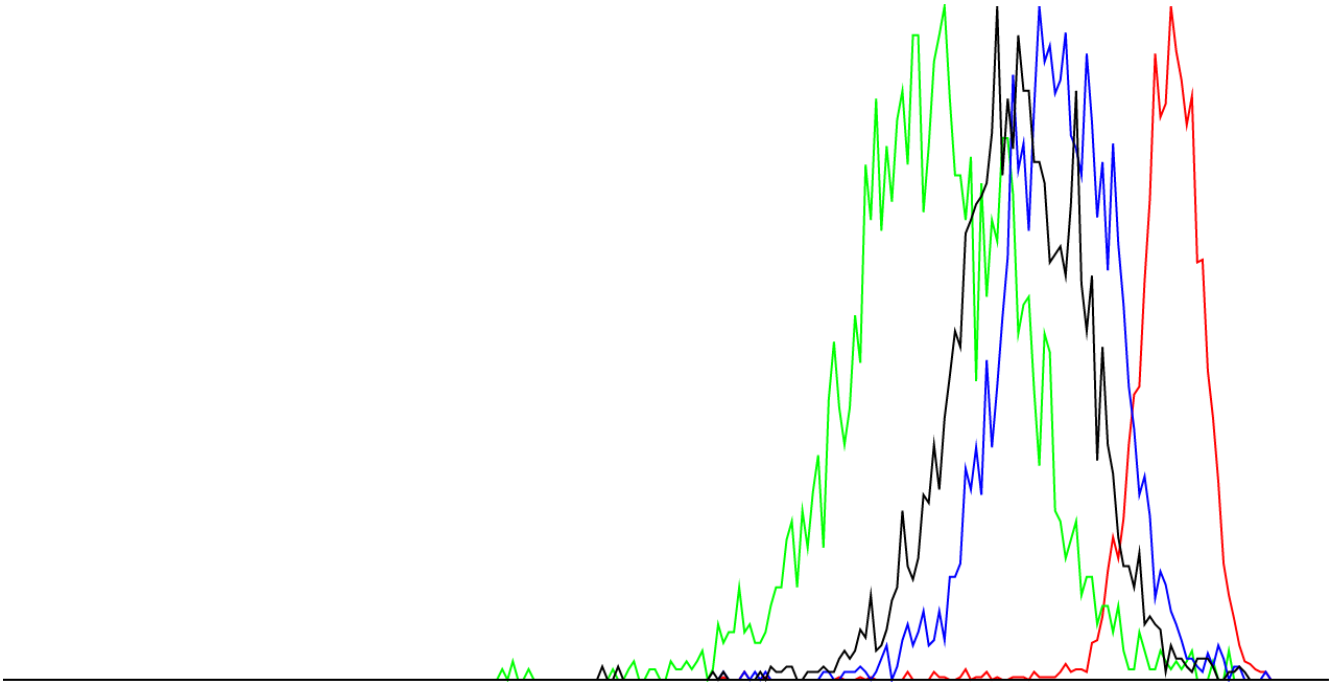

Figure 4 histogramme avant l'algorithme

On constate que l’intensité n’est pas bien équilibrée, mais après l’application de cet algorithme on a les résultats suivants :

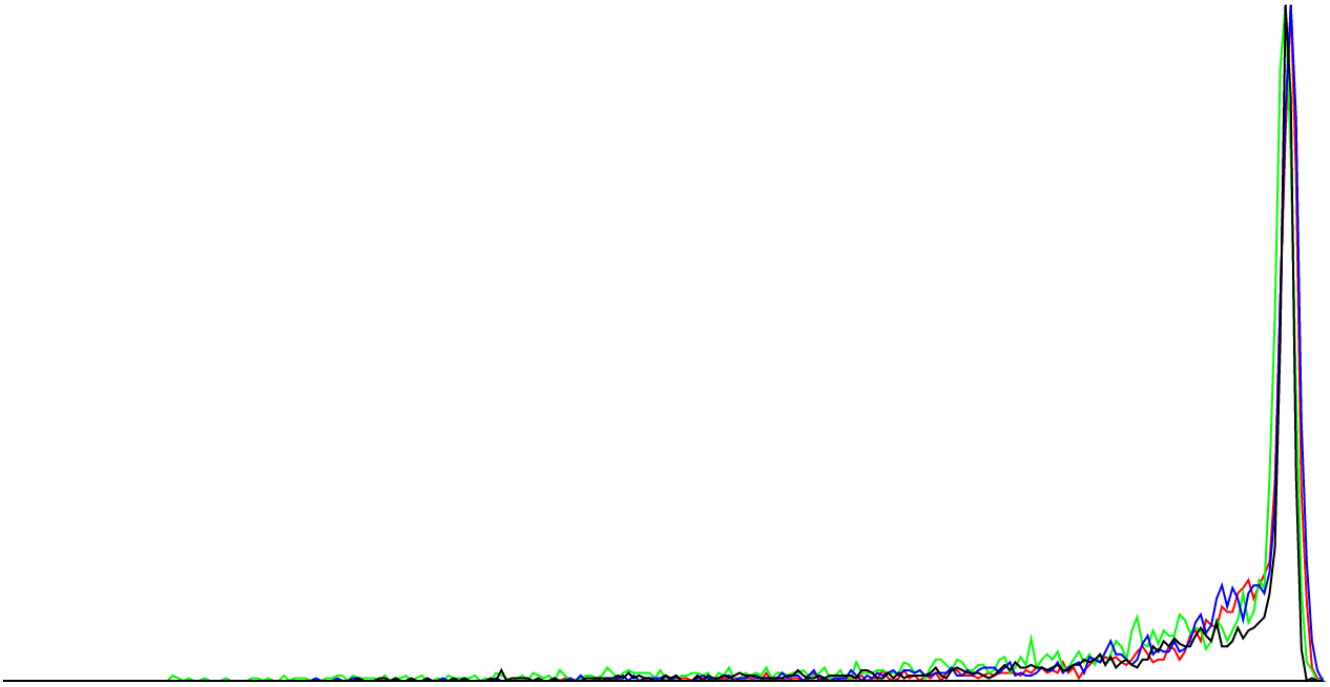

Figure 5 histogramme après l'algorithme

Ceci nous a permet de bien améliorer notre model, on va discuter profondément ces améliorations dans la partie des résultats obtenue.

3. Data training

Convolutional Neural Network (CNN) est utilisé pour extraire des caractéristiques à partir des images de pathologie et d'effectuer une classification des images. CNN a une grande précision et des paramètres moins libre, ce qui est excellent pour les ensembles les datasets à grande échelle. Nous avons construit CancerNet comme notre modèle, et nous utilisons CNN, l'activation RELU et la couche de pooling comme les 3 couches de notre réseau. Après avoir construit le modèle, nous adaptons le modèle sur notre ensemble de formation. Ensuite, nous exécutons le modèle ajusté sur les tests et un ensemble de validation pour générer des résultats de prédiction et de précision.

Figure 6 depthwise separable convolution

Dans la phase du CNN l'image devient abstraite en une carte de caractéristiques, également appelée carte d'activation. Après dans la phase du ReLU (Rectified Linear Unit) Les fonctions d'activation introduisent une non-linéarité dans le modèle, ce qui lui permet d'apprendre des mappages fonctionnels complexes entre les entrées et les variables de réponse, dernièrement dans la phase de Pooling le résultat de l'utilisé de créer des cartes de caractéristiques sous-échantillonnées ou regroupées qui est une version résumée des caractéristiques détectées dans l'entrée. Ils sont utiles car les petits changements dans l'emplacement de l'entité dans l'entrée détectée par la couche convolutive se traduiront par une carte d'entités regroupées avec l'entité au même emplacement. On a choisi d'utiliser depthwise separable convolution parce qu'il est plus efficace dans notre cas et aussi il utilise moins de calcul et de mémoire.

```
27          model = Sequential()
28          model.add(SeparableConv2D(32, (3, 3), padding="same", input_shape=inputShape))
29          model.add(Activation("relu"))
30          model.add(BatchNormalization(axis=chanDim))
31          model.add(MaxPooling2D(pool_size=(2, 2)))
32          model.add(Dropout(0.25))  # (CONV => RELU => POOL) * 2
33          model.add(SeparableConv2D(64, (3, 3), padding="same"))
34          model.add(Activation("relu"))
35          model.add(BatchNormalization(axis=chanDim))
36          model.add(SeparableConv2D(64, (3, 3), padding="same"))
37          model.add(Activation("relu"))
38          model.add(BatchNormalization(axis=chanDim))
39          model.add(MaxPooling2D(pool_size=(2, 2)))
40          model.add(Dropout(0.25))  # (CONV => RELU => POOL) * 3
41          model.add(SeparableConv2D(128, (3, 3), padding="same"))
42          model.add(Activation("relu"))
43          model.add(BatchNormalization(axis=chanDim))
44          model.add(SeparableConv2D(128, (3, 3), padding="same"))
45          model.add(Activation("relu"))
46          model.add(BatchNormalization(axis=chanDim))
47          model.add(SeparableConv2D(128, (3, 3), padding="same"))
48          model.add(Activation("relu"))
49          model.add(BatchNormalization(axis=chanDim))
50          model.add(MaxPooling2D(pool_size=(2, 2)))
51          model.add(Dropout(0.25))  # first (and only) set of FC => RELU layers
52          model.add(Flatten())
53          model.add(Dense(256))
54          model.add(Activation("relu"))
55          model.add(BatchNormalization())
56          model.add(Dropout(0.5))   # softmax classifier
57          model.add(Dense(classes))
58          model.add(Activation("softmax"))
```

Figure 7 Model architecture

4. Résultats :

4.1. La première méthode :

Dans un premier temps nous avons entrainer notre modèle ‘’CANCERNET ‘’ sur les images du training set originaux sans aucune modification par conséquent on a obtenu les résultats suivants :

```
2021-06-18 01:42:59.659264: W tensorflow/python/util/util.cc:348] Sets are
[INFO] evaluating network...
              precision    recall  f1-score   support

           0       0.94      0.48      0.64     71295
           1       0.41      0.92      0.57     28448

    accuracy                           0.61     99743
   macro avg       0.67      0.70      0.60     99743
weighted avg       0.79      0.61      0.62     99743

[[34277 37018]
 [ 2347 26101]]
acc: 0.6053
sensitivity: 0.4808
specificity: 0.9175

Process finished with exit code 0
```

Figure 8 First result

Confusion Matrix :

Predicted Class / Actual Class	Cancer	No Cancer
Cancer	34277	37018
No Cancer	2347	26101

Figure 9 first Confusion Matrix

Après l’obtention de la matrice de confusion. On l’utilise pour le calcul des indices suivant : ‘’Accuracy’’, ‘’Sensitivity’’, ‘’Specificity’’

Accuracy	Sensitivity	Specificity
0.6053	0.4808	0.9175

Conclusions :

1-la précision de ce modèle est très faible (contradiction avec le tuto du site car la data set a été changé), ceci confirme qu'on n'a pas bien prédit le cancer

2-la sensitivité est de même très faible ce qui signifie qu'on a du mal à détecter les personnes qui ont vraiment du cancer

3-par contre la spécificité est très grande ce qui montre que dans la plupart des cas on a bien prédit les personnes qui n'ont pas de cancer

Figure 10 first result plot

Ce graphe nous montre les niveaux de de ''accuracy '' et ''loss'' pendant la phase de l'entrainement et de validation. Ce dernier valide nos conclusions. Comme il est clair du taux de l'indice ''loss'' pendant la phase de validation est très élevé (très proche de 1) ce qui signifie que le modèle fait beaucoup d'erreur dans la détection du cancer.

4.2 La deuxième méthode :

Cette fois-ci et avant la phase de l'entrainement de notre modèle sur les images, on a ajouté une

phase de préparation de ces derniers dans laquelle on va appliquer un algorithme de traitement d'image qui est « égalisation de l'histogramme » pour mieux extraire les informations depuis les images puis on a procédé comme dans la première méthode. Les résultats obtenus sont les suivants :

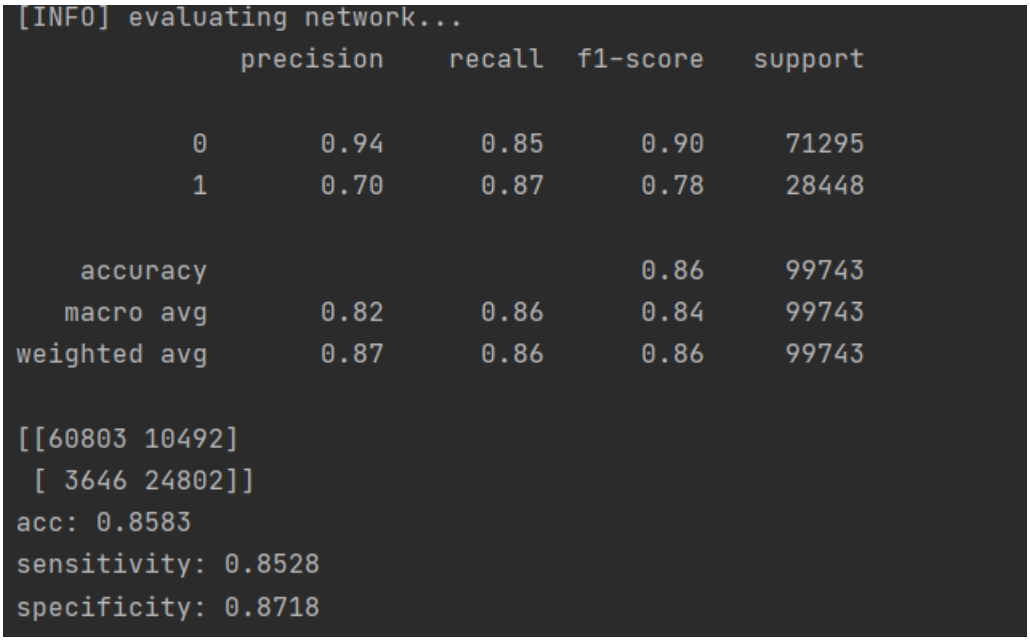

Figure 11 final result

Confusion Matrix :

Predicted Class / Actual Class	Cancer	No Cancer
Cancer	60803	10492
No Cancer	3646	24802

Figure 12 final Confusion Matrix

Après l'obtention de la matrice de confusion. On l'utilise pour le calcul des indices suivant : ''Accuracy'', ''Sensitivity'', ''Specificity''

Accuracy	Sensitivity	Specificity
0.8583	0.8528	0.8718

Conclusions :

1-la précision de ce modèle est très bonne donc on a bien prédit les cas positive ainsi que les cas négatifs du cancer

2-la sensitivité est de même très satisfaisante ce qui signifie qu'on a bien détecter les personnes

qui ont vraiment du cancer (les cas positives)

3- la spécificité est grande ce qui montre qu'on a bien prédit aussi les personnes qui n'ont pas de cancer (les cas négatives)

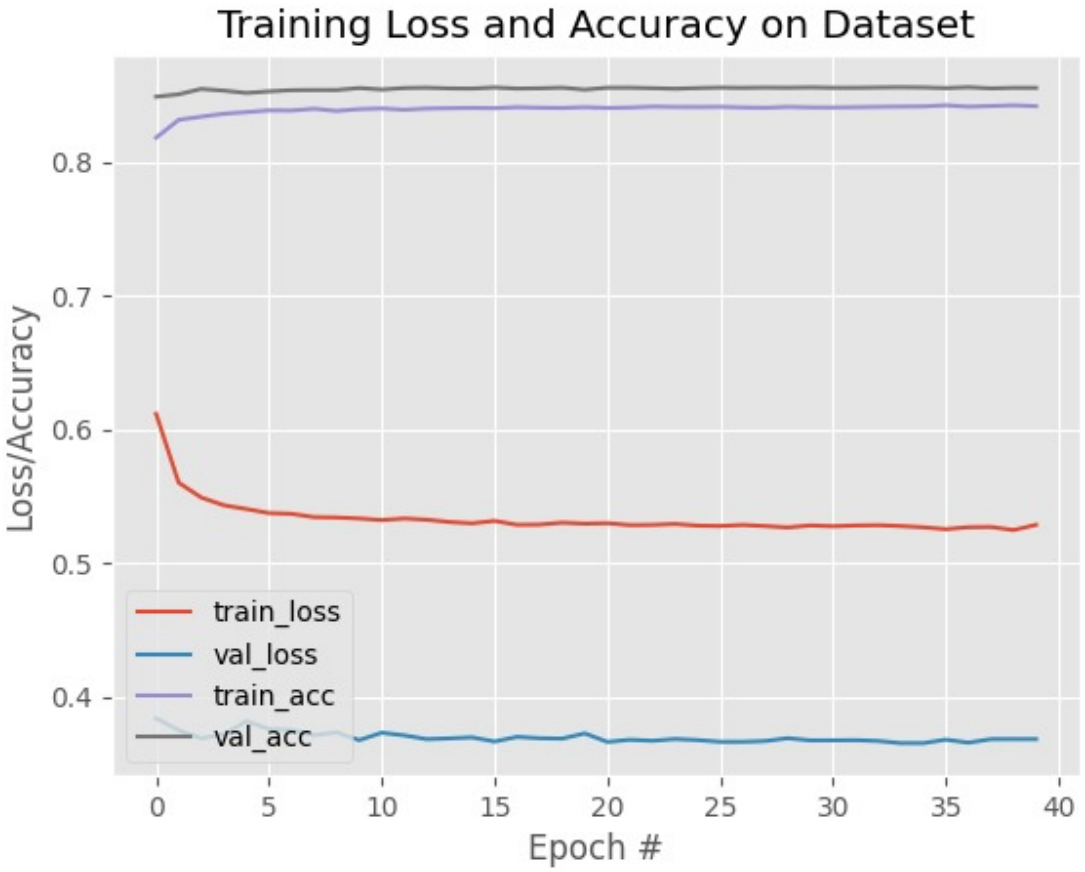

Figure 13 final result plot

Dans cette fois-ci on remarque que le graphe confirme toujours nos conclusions. Premièrement le niveau de ''accuracy'' est très grand dans les deux phases d'entrainement et de validation ainsi que le niveau du « loss » est très faible généralement dans les deux phases ce qui signifie que notre modèle marche bien cette fois-ci.

5 Test :

Pour le test de notre modèle nous avons utilisé un dossier « Own_Test » qui contient trois images de tests qui sont positive (1) et trois autre négative (0) comme le montre les captures d'ecran

suivants :

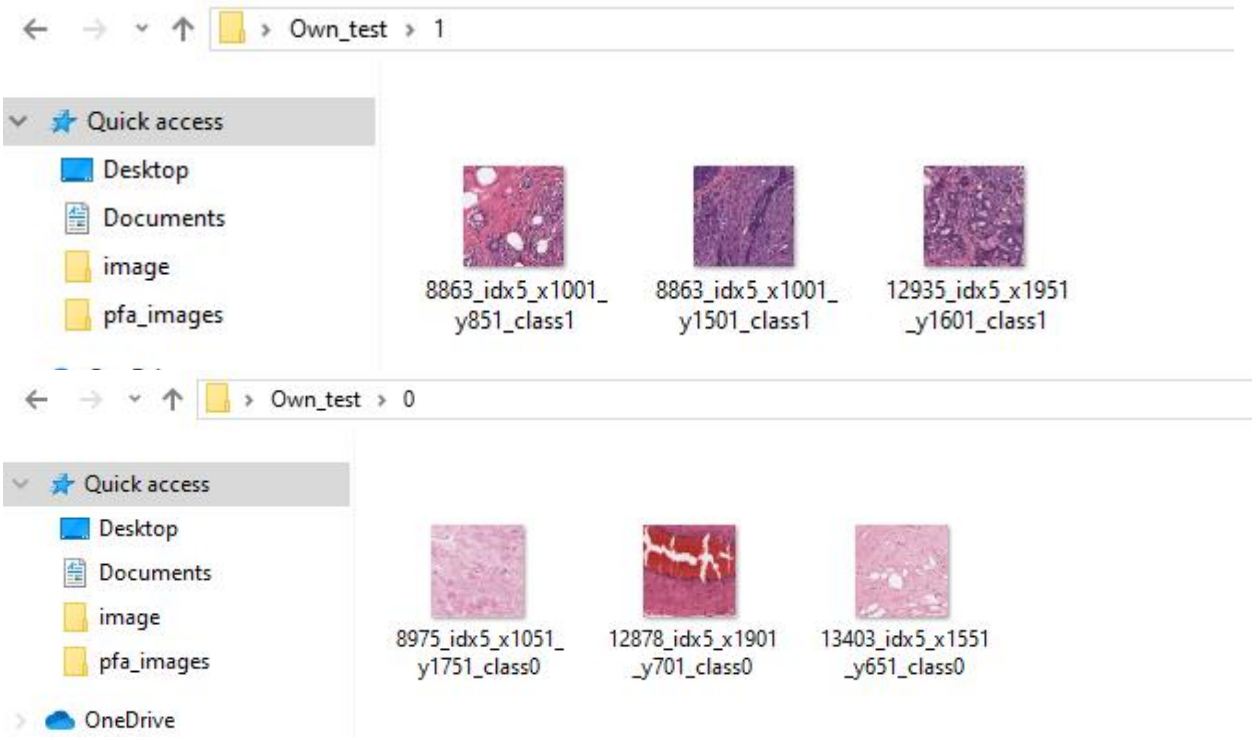

Figure 14 our own test

Et après l'exécution du script du test nous avons obtenu les résultats suivants :

```
PS C:\Users\hpcode\Desktop> python script.py
2021-06-23 14:36:25.569834: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'cudart64_110.dll
'; dlerror: cudart64_110.dll not found
2021-06-23 14:36:25.589834: I tensorflow/stream_executor/cuda/cudart_stub.cc:29] Ignore above cudart dlerror if you do not have a GPU set up
 on your machine.
2021-06-23 14:36:30.301219: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'nvcuda.dll'; dle
rror: nvcuda.dll not found
2021-06-23 14:36:30.313615: W tensorflow/stream_executor/cuda/cuda_driver.cc:326] failed call to cuInit: UNKNOWN ERROR (303)
2021-06-23 14:36:30.334482: I tensorflow/stream_executor/cuda/cuda_diagnostics.cc:169] retrieving CUDA diagnostic information for host: DESK
TOP-RHCE60L
2021-06-23 14:36:30.345551: I tensorflow/stream_executor/cuda/cuda_diagnostics.cc:176] hostname: DESKTOP-RHCE60L
2021-06-23 14:36:30.357496: I tensorflow/core/platform/cpu_feature_guard.cc:142] This TensorFlow binary is optimized with oneAPI Deep Neural
 Network Library (oneDNN) to use the following CPU instructions in performance-critical operations:  AVX AVX2
To enable them in other operations, rebuild TensorFlow with the appropriate compiler flags.
Found 6 images belonging to 2 classes.
2021-06-23 14:36:35.987696: I tensorflow/compiler/mlir/mlir_graph_optimization_pass.cc:176] None of the MLIR Optimization Passes are enabled
 (registered 2)
[[8.8270456e-01 1.1729540e-01]
 [9.9985147e-01 1.4856737e-04]
 [9.9996984e-01 3.0178406e-05]
 [1.6319030e-04 9.9983680e-01]
 [9.1276108e-04 9.9908721e-01]
 [6.8347968e-02 9.3165201e-01]]
finished
PS C:\Users\hpcode\Desktop>
```

Figure 15 the result of our own test

Observations :

Pour les 3 premiers ligne fait référence aux images dans le dossier 1 c-à-d les pour les personnes qui ont du cancer et on remarque que les probabilités que ces personnes ont du cancer sont très élevés 88.27 %, 99.85%,99.99%

Par contre les 3 derniers lignes fait référence aux images du dossier 0 qui n'ont pas de cancer. On remarque que les probabilités que ces personnes ont du cancer sont très faible

0.016%,0.091% ,6.83%

Ce test valide bien le fait que notre modèle marche très bien dans la détection du cancer du sein.

Conclusion :

Un modèle rectifié a été proposé par nous pour l'entrainement d'un grand nombre d'images pathologiques étiquetées, puis leur teste et leur validation sur l'autre ensemble d'images pour prédire le cancer du sein. La solution produit une précision d'entraînement qui est très bonne. Nous pouvons également utiliser notre modèle en lui donnant en entré une image d'une histologie de cancer du sein et il pourra prédire la probabilité que cette personne est atteinte d'un cancer du sein ou non, ainsi que la précision des résultats.

Notre solution améliore l'application de l'apprentissage automatique dans le domaine médical. Et démontre comment CNN peut être appliqué en l'imagerie médicale. Elle valide les résultats de certaines recherches qui sont en cours mais améliore également notre compréhension dans ce domaine.

Nous pouvons améliorer notre modèle dans le futur pour l'adapter à d'autres ensembles de données différents, tels que des images pour d'autres parties du corps ou pour des images montrant d'autres types de cancers ou d'autres types de maladies, Avec l'amélioration de la précision bien sûr.

Bibliographie:

- ✓ Le tutoriel utilisé pour l'implémentation de notre modèle de prédiction Breast cancer classification with Keras and Deep Learning - PyImageSearch
- ✓ Le lien du data set utilisé Breast Histopathology Images | Kaggle
- ✓ La documentation de l'algorithme utilisé dans le pré-traitement des images OpenCV: Histograms - 2: Histogram Equalization