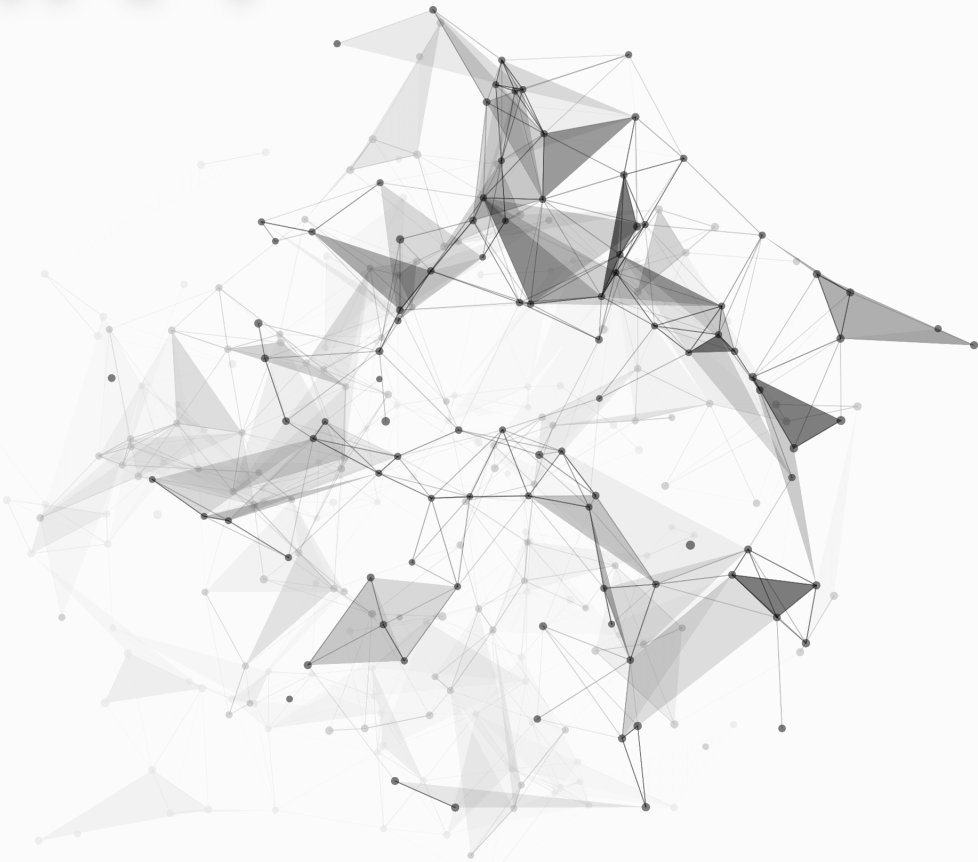
SUMMARY

Group 2

Qianyi Xue, Kejia Liu, Yuchen Ge, Longxin Wang

Breast cancer classification based on BreakHist dataset



01 Data

In women worldwide, breast cancer is the most common cancer and the second leading cause of cancer death (Boyle & Levin, 2008).

Composed of **7909 microscopic images** of breast tumor tissue with **four magnification** factors (40x, 100x, 200x, and 400x). Divided into **two groups: benign and malignant.**

Benign(4): adenosis, fibroadenoma, phyllodes tumor, and tubular adenoma.
Malignant(4): carcinoma, lobular carcinoma, mucinous carcinoma, and papillary carcinoma.

Magnification	Benign	Malignant	Total
40×	625	1370	1995
100×	644	1437	2081
200×	623	1390	2013
400×	588	1232	1820
Total	2480	5429	7909
# Patients	24	58	82

Table 1. Image Distribution by Magnification Factor and Class [2].

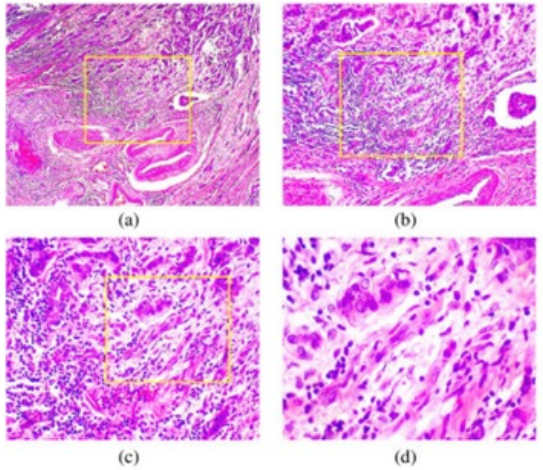


Figure1. Slide of breast malignant tumor (stained with HE) seen in different magnifification factors: (a) 40×, (b) 100×, (c) 200×, and (d) 400×. H [2].

8 Classification issues

01 Model

The overall approach is divided into 2 steps: extracting features from the images and building the classifier.

Extracting Features : **vgg16, vgg19, xception, resnet50, inception, inception_resnet**
Classification: **FC, LR, SVM,etc.**

We constructed a concise architecture, taking a 5-layer dense layer for each model to facilitate rapid convergence, and the final output matrix of $[image\_num, 8]$(8 classes) is obtained as the extracted features.

```
def resnet50_model(load_weights = True):
    base_model = ResNet50(include_top=False, weights='imagenet', input_tensor=None,
input_shape=(image_height, image_width,3), pooling='avg')
    x = base_model.output
    x = Dense(1024, activation='relu')(x)
    x = Dense(256, activation='relu')(x)
    x = Dense(64, activation='relu')(x)
    x = Dense(16, activation='relu')(x)
    x = Dense(8, activation='softmax')(x)
    model = Model(inputs=base_model.input, outputs=x)
    model._name = 'resnet'
    return model
```

Code of resnet50

Regarding the training parameters, we set the upper limit of **epoch to 500**, the **b** according to our computational ability, the initial value of learning rate to **0.0** function to $"categorical\_crossentropy"$, and the strategy of early stopping learning rate for training.

```
    model.compile(loss="categorical_crossentropy", optimizer=Adam(lr=0.00005), metrics=['a
    early_stopping = EarlyStopping(patience=10, verbose=2)
    model_checkpoint = ModelCheckpoint(model_name + "_combine" +".model", save_best_o
verbose=2)
    reduce_lr = ReduceLROnPlateau(factor=0.1, patience=5, verbose=2) #min_lr=0.00001,
```

Key for training

01 Result

We present the final accuracy of classification using fully connected layers.

	Vgg16	Vgg19	xception	resnet50	inception	inception_resnet
40x	0.807	0.786	0.722	0.702	0.725	0.837
100x	0.845	0.777	0.744	0.880	0.735	0.819
200x	0.855	0.784	0.774	0.872	0.784	0.865
400x	0.826	0.803	0.754	**0.883**	0.735	0.841

Table 2. Table of the accuracy of the classification results of different models with FC.

Cyclic Learning Rate Decay

The learning rate is too large may cross the optimal value, and the learning rate is too small cannot converge for a long time. The basic idea of learning rate decay is that the learning rate decays gradually as the training progresses. Adjusting the learning rate periodically may give better results.

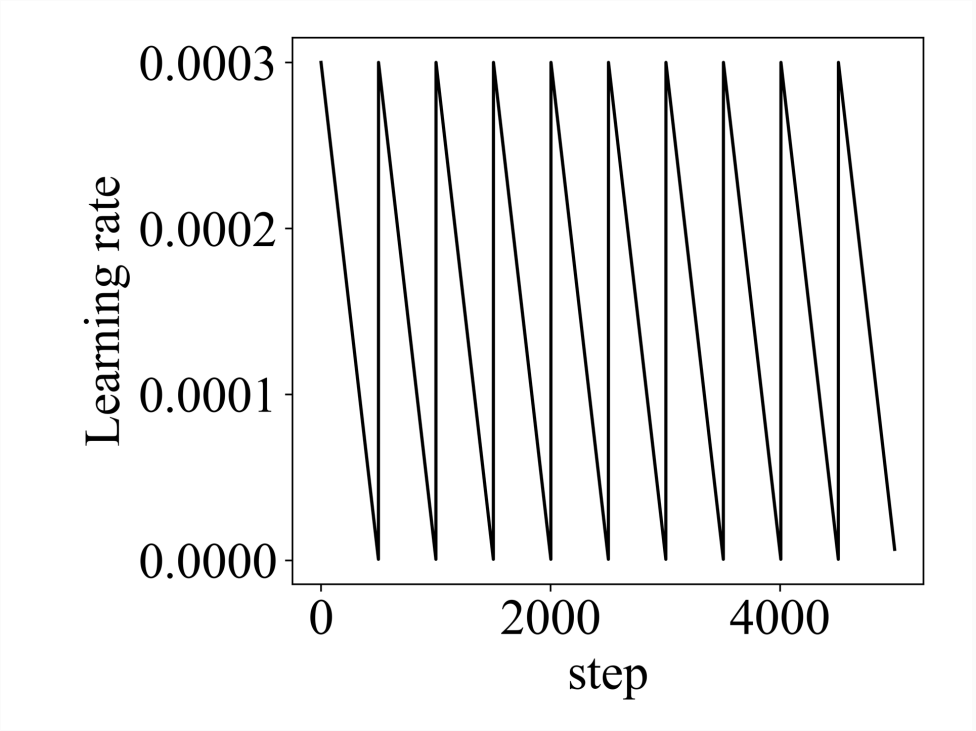
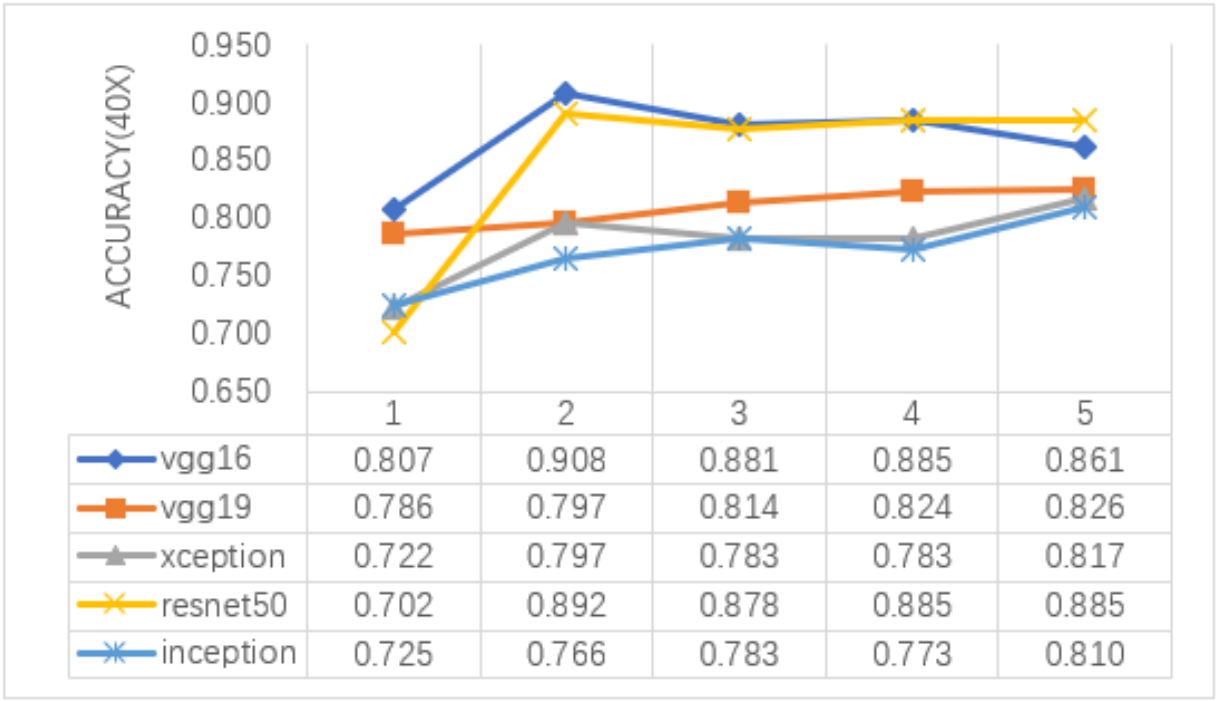


Figure: Simulation training using CLR.

02 Tricks

Cyclic Learning Rate Decay

Comparison of cyclical learning rate decay and general learning rate decay



	1	2	3	4	5
vgg16	0.807	0.908	0.881	0.885	0.861
vgg19	0.786	0.797	0.814	0.824	0.826
xception	0.722	0.797	0.783	0.783	0.817
resnet50	0.702	0.892	0.878	0.885	0.885
inception	0.725	0.766	0.783	0.773	0.810

Figure: Results of 5 training sessions with different model cycles at 40x.

Magnification Model	Vgg16	Vgg19	Xception	Resnet50	Inception	Inception_resnet
40x	0.786	0.647	0.680	0.702	0.675	0.722
100x	0.825	0.680	0.680	0.777	0.793	0.848
200x	0.794	0.753	0.686	0.743	0.699	0.757
400x	0.731	0.761	0.659	0.773	0.701	0.731

Table 4. Test accuracy without cyclical learning rates decay

There is a slow increase.

Local optimal solutions for small data sets

We selected **two** of the eight types of breast cancer cells, benign selected for fibroadenoma and malignant selected for ductal_carcinoma.

These two breast cancers were selected because they represent a larger portion of the entire dataset and they are the representatives of the benign and malignant tumors correspondingly.

Magnification Model	Vgg16	Vgg19	Xception	Resnet50	Inception	Inception_resnet
40x	0.946	0.886	0.783	0.952	0.994	0.970
100x	0.920	0.931	0.695	0.966	0.971	0.977
200x	0.906	0.953	0.947	0.959	0.982	0.959
400x	**0.986**	0.986	0.959	0.973	0.986	0.973

Table 6. Test accuracies of two breast cancer cells types (fibroadenoma and ductal_carcinoma).

From Table 6, we can see that the accuracy of the dichotomous classification task has improved significantly, and the average accuracy can reach more than 90% compared to the 8 classifications. In practical applications, we can jointly use while classifier and 8 classifiers to get more accurate judgments.

Increasing the number of dense layers

Research by Krizhevsky et al. (2017) demonstrated that **increasing the depth of the network** is advantageous to extract more effective features and **enhance the performance of the network.**

magnification\model	Vgg16 + 5 FC layers	Vgg16 + 6 FC layers	Vgg16 + 7 FC layers
40x	0.874	0.845	0.918
100x	0.888	0.921	0.870
200x	0.881	0.867	0.876
400x	0.907	0.918	0.907

Table 3. Test accuracies for three networks with VGG16 as the base model (batch size = 32).

- More layers will help extract more features, which can be done up to a certain extent.
- More layers = more parameters -> overfitting or need more data

Replacing softmax with logistic regression

Since the softmax classifier is more suitable for datasets with more inter-class mutual exclusivity and less overlap, in the case of low similarity of features that cannot be model extracted, we guess that using LR classifier will give better results.

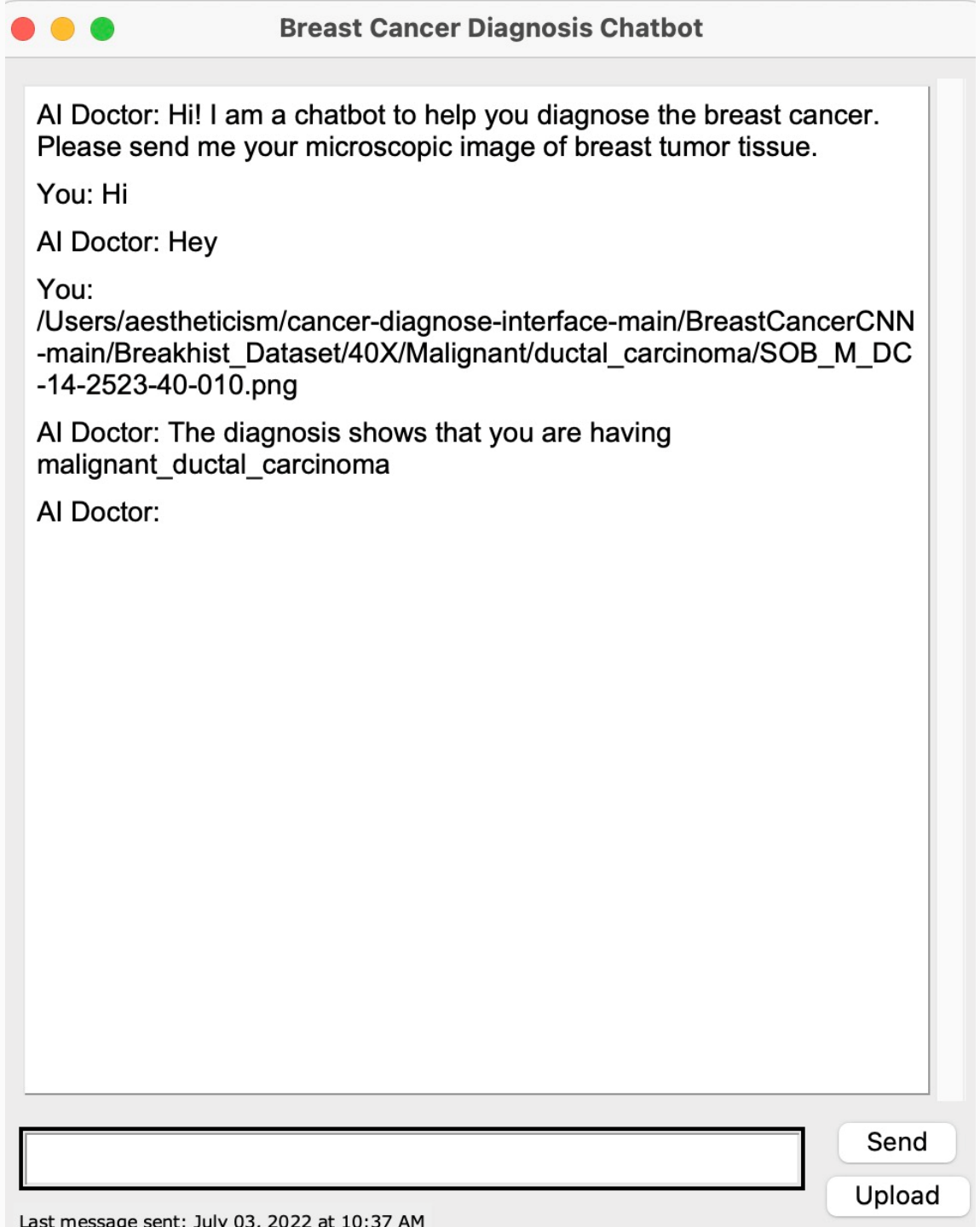
	VGG16	VGG19	Xception	Resnet50	Inception	Inception_res
40x	0.807	0.786	0.722	0.702	0.725	0.837
100x	0.845	0.777	0.744	0.880	0.735	0.819
200x	0.855	0.784	0.774	0.872	0.784	0.865
400x	0.826	0.803	0.754	**0.883**	0.735	0.841

Table 2. Table of the accuracy of the classification results of different models with FC.

	VGG16	VGG19	Xception	Resnet50	Inception	Inception_res
40x	0.834 ↑	0.834 ↑	0.790 ↑	0.834 ↑	0.817 ↑	0.871 ↑
100x	0.828 ↓	0.822 ↑	0.738 ↓	0.828 ↓	0.770 ↑	0.882 ↑
200x	0.885 ↑	0.841 ↑	0.757↓	0.821 ↓	0.807↑	0.873 ↑
400x	0.845 ↑	0.826 ↑	0.727↓	0.848 ↓	0.754 ↑	0.923 ↑

Table 7. Table of the accuracy of the classification results of different models with LR.
(↑ is higher than the original and ↓ is lower than the original)

03 Interface



Generate the model.h5

↓

Save the trained models' architectures in trained.py

↓

Transfer parameters into train.py through model.h5

↓

Model works and predicts

↓

Display

03 Interface

Specific implementation:

1. Imported Tkinter package to implement the chatbot.
2. The class "ChatInterface(Frame)" is individualized and built on tkinter package by implementing a chat box.
3. Imported the nltk package to tokenize the 'breast cancer.txt'.
4. Added the upload button to help upload the image.
5. The chatbot can talk wisely thanks to the nltk package.

Finally, we are able to gathering together the codes of training inception_resnet_model and the interface, following the topological structure of data.

Figure 4. The code framework diagram of the interface part.

04 Discussion

The accuracy we obtained so far is between **80%-90%**, and we still have a lot of room for improvement in training and model building for this dataset, and the previous methods and accuracy rates using this dataset are listed in the next page. In the future, we can build classification models using more powerful classifiers such as SVM, RF and XGBOOST. We can also make a deeper use of CLR in the tuning of learning rates. We should make greater use of models and datasets for more effective training.

04 Achievements in this dataset (Literature Review)

Author(s)	Methodology	Dataset	Accuracy (in %)
Spanhol et al.(2015)	• Feature Extraction: using LBP, CLBP, LPQ, GLCM, PFTAS and ORB. • Classification: 1-NN, QDA, SVM, and RF	BreaKHis	80%-85%
Spanhol et al.(2016)	• Pre-trained CNN Models: LeNet and AlexNet CNN	BreaKHis	80.8%-85.6%.
Bayramoglu et al.(2016)	• Proposed two CNN models: • Single task CNN: for predicting the malignancy. • Multi-task CNN: for predicting malignancy and image magnification levels	BreaKHis	80.6%-83.3%.
Spanhol et al.(2017)	• Extracted deep features (DeCAF): using pretrained CaffeNet.	BreaKHis	83.6%-84.8%.
Sudharshan et al(2019)	• Proposed MIL based approach: APR, KNN, DD, SVM, non-parametric algorithm and MIL-CNN	BreaKHis	83.4%-92.1%.
Gour et al.(2020)	• Developed a deep residual convolutional neural network (ResHist) for breast cancer diagnosis. • Proposed a data augmentation technique.	BreaKHis	84.34%-92.52%
Dabeer et al(2019)	• 7009 images from BreakHis database is used. Images captured are distributed into 4 magnification levels.	BreaKHis	99.86%
Bardou et al.(2018)	• 25% of the training data is used for cross validation. • A 5 CNN layer topology with 3*3 filters and 2 fully connected layers is employed to classify the dataset. • RELU layer is also applied.	BreaKHis	96.15% and 98.33% for the binary classification. 83.31% and 88.23% for multi-class classification.

05

Reference

Boyle, P., & Levin, B. (2008). *World cancer report 2008*. IARC Press.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. https://doi.org/10.1109/cvpr.2016.90

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). ImageNet classification with deep convolutional Neural Networks. *Communications of the ACM*, *60*(6), 84–90. https://doi.org/10.1145/3065386

Satiate, A. (2019). *BreakHist-Dataset-Image-Classification*. GitHub. Retrieved July 1, 2022, from https://github.com/Anki0909/BreakHist-Dataset-Image-Classification

Simonyan, K., & Zisserman, A. (2014, December 19). *Very deep convolutional networks for large-scale image recognition*. arXiv.org. Retrieved July 2, 2022, from https://arxiv.org/abs/1409.1556v4

Spanhol, F. A., Oliveira, L. S., Petitjean, C., & Heutte, L. (2016). A dataset for breast cancer histopathological image classification. *IEEE Transactions on Biomedical Engineering*, *63*(7), 1455–1462. https://doi.org/10.1109/tbme.2015.2496264

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., & Rabinovich, A. (2015). Going deeper with convolutions. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. https://doi.org/10.1109/cvpr.2015.7298594

x12hengyu. (2022). *Skin-Cancer-Classification-Chatbot*. GitHub. Retrieved July 1, 2022, from https://github.com/x12hengyu/Skin-Cancer-Classification-Chatbot

Thank You