

# Patch-based CNN for Breast Cancer Image Classification

## ABSTRACT

With the development of AI, scientists have been putting on efforts to solve problems in the medical field by using machine learning models. As we know, breast cancer is considered as one of the most common cancers in women. The Convolutional Neural Network (CNN) is a particular deep learning model as a solution that deals with image recognition problems.

**Keywords:** CNN; breast cancer; discriminative patches; Gaussian Smoothing;

## 1. INTRODUCTION

Convolutional Neural Networks (CNNs) are state-of-the-art classifiers that have been widely used for image recognitions. However, CNNs have limitations on high resolution images because of high computational cost. Solutions need to be proposed and implemented in order to deal with such limitations. In this project, we are given x-ray images of women's breast. Each image has very high resolution than normal images.

For the purpose of avoiding risks of losing any important details, dividing the image into several patches is proposed rather than just down-sampling the original image. By applying CNNs models, we consider each patch as an instance and perform multi-instance deep learning with image-level labels.

## 2. DISCUSSION OF THE PROJECT

### 2.1 Design procedure

Our project basically follows the implementation algorithms in the article "Patch-based Convolutional Neural Network for Whole Slide Tissue Image Classification, CVPR 2016".

In general, the design procedure can be separated into two main parts: Discriminative Patches Selection and Image Level Prediction.

In the first part, we follow the following five steps:

- Split training images into patches
- Train CNN using patches
- Output probability maps
- Apply Gaussian smoothing and thresholding to maps
- Extract discriminative patches and restart from step 1 until convergence.

Given sets of high resolution images with their corresponding training, validation and test labels, we first read all images and then map each input image array to their labels. We also split all images into different views(subsets) according to the information of naming type of each image.

For the CNN model, we use a single convolutional layer due to the small size of patches and our laptop computation power. We use the stochastic gradient descent as the optimizer for training.

In the training process, for each image, we split it into multiple patches with given size of height and width and then train all these patches for two epochs. After the first iteration, we use the Gaussian Smoothing to with given threshold filter out the discriminant patches of that particular image. Then we use these discriminant patches as input patches of each image. Rerun the whole process until convergence. For testing the feedbacks, we use visualization to analyze the training feedbacks by comparing the generated image that differentiates discriminant patches to the original image.

## 2.2 System Description

Data description: Data folder contains whole train and test x-ray images, filename includes patient ID, left or right breast and the view of the breast. For example, “20586934 \_6c613a14b8 0a8591 \_MG\_L\_CC \_ANON”, 20586934 is the ID of an image, 6c613a14b8 0a8591 is patient’s ID, which can be used to merge four views of images, R or L represents right or left breast, and ‘cc’ stands for the view.

Train.txt and val.txt are train and validation data set respectively, which includes image ID and corresponding label. The test.txt only contains image ID

Our system helps us to classify the x-ray images into normal and abnormal to detect the cancer. There are four main steps in our system

- Pre-processing, we Split different views of the same patient, they are R\_CC, L\_CC, R\_ML, L\_ML respectively. After that we use sampling method to balance the data set, because the label in train data set is unbalance, which may cause inaccurate results, as a result we balance the 0 and 1 in the data set.
- The second step is to process images, we set the size of each Patches among 50-200. After comparison, we decide the optimal size is 50. when the system detects the tumour, it will be represented by the color white, the rest of part will be displayed with the color black, and the whole image will display it as grayscale. Besides, there are three main functions in this procedure, the first one is to generate patches function, which is used to divide the whole image into small patches, and the second function is to generate image by small patch images, this function is to visualize the original image, we gather patch images together by mapping method.
- The third step is building CNN model, we build 16 layers 2D convolution, activation is 'relu', in addition the pooling size is (2,2), dropout rate is 0.5, and dense is from 16 to 2, in the and adam is sgd(lr=0.0001).
- The last step is to detect tumor, At first we use discriminative patches to train our CNN model for 2 epochs, and the system output all of the patches of an image by

patch-level predictions, because the similar patches have the same label, and then we get the probability image, after that Gaussian smoothing to modify the result.

## 2.3 Algorithm Implementation

We use dictionary to build connection between training id from train.txt and image name, and then use map function to connect each image with corresponding label in the train, validation and test txt files.

- For generating patches, we first assign the size of patch and then split the original into smaller patches with the size we've specified. Then we use loops to scan and cut out each patch and we create a list to store each array of smaller patches that have been cut out.
- For training the model, we use a simple CNN model with the stochastic gradient descent optimizer. We use the Gaussian Smoothing to filter out the discriminant patches for each image.
- For the data pre-processing, we use the image generator function with vertical and horizontal flips to be TRUE in order to avoid over-fitting.
- For the training process, we first split each image into patches and assign the label of the original image to labels of all smaller patches and train patches of each image of for the first iteration. After the first iteration, we use the Gaussian Smoothing to get the probability of the hidden variable which indicates whether the instance  $x$  is discriminative or not. We set a threshold to get the discriminative patches. Then, we use the discriminant patches of each image as inputs for the rest iterations of training. For each image, we train it until convergence and then train the next image. In the second part, when a new image comes, we first split the image into multiple patches and feed them into the CNN model that we have already trained. It will output probabilities for each patch. We count the number of predicted classes and set a threshold. If the percentage of predicted class 1 is above the threshold, we predict the image to be abnormal. To test the performance, we use the visualizing helper function to trace and analyze the discriminant area and compare to the original image.

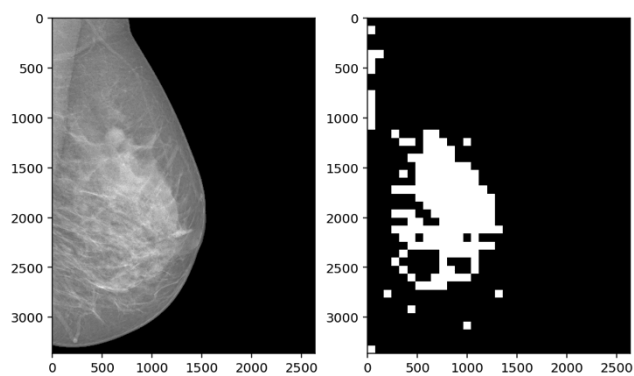
Finally, we've trained four models with one for each view type of images to make predictions on the validation set and the test set. Each model is used to train a group of images of particular view and we've set difference parameters for the whole process such as patch size and threshold of the discriminant patch.

## 3. TEST

### 3.1 Test Result

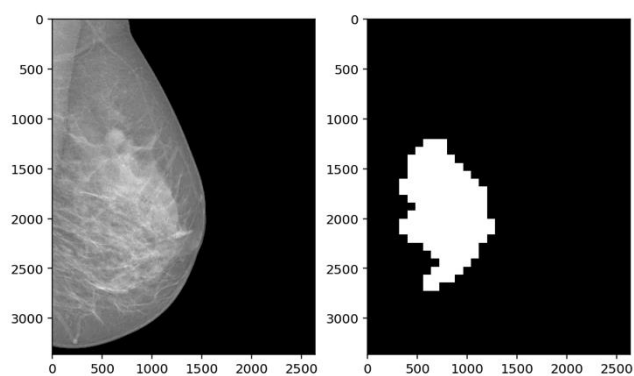
Below are the feedbacks from training of a particular image for the first three iterations. We can see that it almost converges for the first two iterations. It clearly shows the location of the potential tumour.

ID:22614568, Label:1



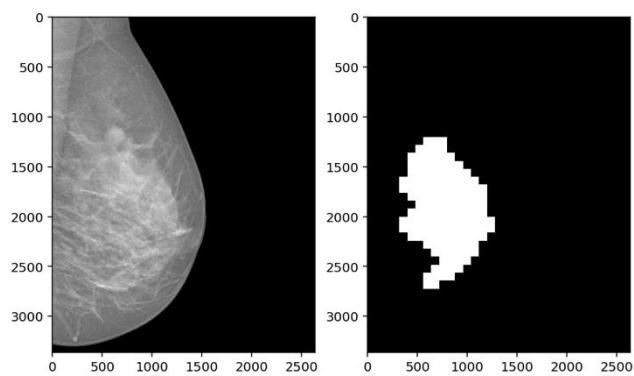
**Figure 1.** Training feedbacks for the first iteration

ID:22614568, Label:1



**Figure 2.** Training feedbacks for the second iteration

ID:22614568, Label:1



**Figure 3.** Training feedbacks for the third iteration

### **3.2 Evaluation Metrics**

Given a picture, if one of the discriminative patch has tumour, the output image will be labelled as 1, and if aggregation of small patch doesn't have tumour, result will be labelled as 0.

## **4. CONCLUSION**

In conclusion, we design and implement a patch-based CNN model on the classification of breast cancer in women. We've also implemented an Expectation Maximization(EM) based method introduced from the article that is able to identify discriminate patches of a given image with high resolution and use these patches as part of inputs during training process. In general, we have gained some practical experience on the deep multi-instance learning with high resolution images in this project. In the future, we will optimize our CNN networks and training iterations so that it can fit well on pathology datasets that have even larger scale.

## **REFERENCE**

Hou, L., Samaras, D., Kurc, T. M., Gao, Y., Davis, J. E., & Saltz, J. H. (2016). Patch-based convolutional neural network for whole slide tissue image classification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 2424-2433).