MSBD 6000B Deeping Learning Project 3

CHANG, YAJIE 20459996
XU, CHAOYI 20461901
WANG, WEIXIAO 20476516
QIU, FENG 20398324

1. Data description

For this project, we are aiming to classify the x-ray images into normal and abnormal to help detect the breast cancer. There are in total 411 high resolution medical images in our dataset.

2. Data pre-processing

2.1 patch division

Because of the high resolution for each image, it's hard to store such big feature mappings in the GPU. It's also not allowed to resize the images, for it will lose some details. Therefore, we divided the whole slide issue image into multiple patches. And we predict the label of a whole slide image based on patch-level predictions.

2.2 black patch removing

In order to get discriminative patches with higher probability values, we removed those patches unrelated to our training process and prediction process, which is discriminative patch extraction. For each of the breast x-ray image, there are plenty of fully black patches. We used Isblack() function to remove the fully black patches.

3. Experiment deployment

3.1. Predict the label for each patch.

We trained a CNN model to predict the label for each patch with the assumption that each patch has the same label as its image. The layers are listed here as table 1.

Layer	Filter size, stride, padding	Output W*H*N
Input	-	100 * 100 * 1
Conv	3 * 3, 1, “SAME”	100 * 100 * 8
Max-pool	2 * 2, 2, “SAME”	50 * 50 * 8
ReLU + LRN	-	50 * 50 * 8
Drop (0.5)	-	50 * 50 * 8
Conv	3 * 3, 1, “SAME”	50 * 50 * 16
Max-pool	-	25 * 25 * 16
ReLU + LRN	2 * 2, 2, “SAME”	25 * 25 * 16
Drop (0.5)		25 * 25 * 16
Conv	3 * 3, 1, “SAME”	25 * 25 * 8
Max-pool	-	13 * 13 * 8
ReLU + LRN	2 * 2, 2, “SAME”	13 * 13 * 8
Drop (0.5)		13 * 13 * 8
Flatten	-	1352
FC	-	16
ReLU		16
FC	-	2
SoftMax	-	2

We then do Gaussian smoothing. When the Gaussian filter smoothes the pixels in the neighborhood of the image, the pixels in different positions in the neighborhood are given different weights, and the image is smoothed while preserving the overall gray level distribution of the image.

Gaussian filter is a linear filter, which can effectively suppress the noises, smooth the images. Its principle is similar to the principle an averaging filter, taking the mean of the pixels in the filter window as the output.

3.2. Select the discriminative patches

We do two steps in this section.
For the first section, we only look one image at a time. We set a probability p1=30% to discriminate non-black patches. We need to make sure that one image has more than 30% patches among all the non-black patches can be used in the next step.

For the second section, we glance at the whole dataset. We set a threshold to divide the all the data into two clusters. If the percentage of patches selected higher than the threshold, we consider the label of this image as 1.

Finally, we take the minimum values of this two parameters.

3.3. Predict the label for each image

We have to predict the image label based on its patches' label. We choose Support Vector Machine to predict the image-level label. Because it only has two classes, and we only have 0 and 1 as outputs. For this step, we count the total number of label 0’s patches and label 1’s patches for each image as the input of our SVM model. The output of the SVM model will be the real label for corresponding image.

4. Analysis

There may be some way to improve the result.

Firstly, we find that there is about 3/4 of train data comes have label 0. Firstly, we do not focus on it for we thinking this percentage can be accepted. However, after CNN we find almost all the data regarded as class 1, and we have to set a threshold, to make the patch divided into two classes. That may be because the data is not balance and the difference between different patch is not big, so some of they have much more possibility to classified to the bigger class. Maybe oversampling is a good idea.

Another problem is that the model may be too simple, add more filters may get a better result. In project 2, we have hundreds of filters, and give much better predict(0.66/ 5 class)

But the calculate ability can be a limitation, for we have too many patch, and it is hard for we to solve this problem. For example, we meet at least 3 times computer crash. Making the data bigger without down-sampling can cause more problem.

Besides, we can do some pre-process before training model. One possible way is separate structure and texture before train the model. Many people use this way to solve the problem directly, but the problem is how to define texture in high resolution picture. We think some patch-based way can be a good choice (for example, Bilateral Texture Filtering, Hojin Cho, etc). For patch-based way can be fast if we have GPU and in this way we can regard several pixels as an integer, define texture as high frequency change of color. But, this method may also delete some information which may be useful for computer.