Acute Ischemic Stroke Lesion Pattern Classification in DWI Using Machine Learning

Ruize Zhang*, Qingyi Zhao*
University of California, Los Angeles, USA

I. Introduction

DFFUSION-weighted magnetic resonance imaging (DW-MRI or DWI) is proved to be the most accurate technique to detect and diagnose acute ischemic stroke[1], which makes it play an essential role in analyzing different patterns of stroke. Previous studies show that patterns of lesions in DWI of acute ischemic stroke contain a significant amount of information on the pathogenic mechanisms underlying the stroke, and even the outcome of the stroke. The lesion patterns generally have six different groups: territorial infarcts, small superficial infarcts, other cortical infarcts, internal border zone infarcts, small deep infarcts, and deep infarcts, as shown in Figure 2[2]. In this project, we aim to develop a model that can accurately classify the lesion pattern in a DWI of a patient with acute ischemic stroke into one of the six categories described above. We tried several different machine learning algorithms, including random forest, SVM, and gradient boosting, among which gradient boosting showed the most promising result in our experiment setting. We also designed a Convolutional Neural Network (CNN) to test if deep learning is also applicable to this task. The result showed a high accuracy but a rather low AUC-ROC score (shown below).

II. Dataset

The dataset we use consists of DWI images of 241 patient of acute ischemic stroke. For each patient, there are 26 slices of the MRI images of his/her brain, and each of the slice has a size of 130 $\times$ 130 pixels. We pre-processed the data images and removed data of 12 patients due to lack of label as ground-truth. So in the end, we have data for 229 patients for training and testing. The data is provided by professor Fabien Scalzo.

III. Experiment

A. Machine Learning Approach

1) Method: To apply the models in Scikit-learn and deep neural network to the data, we first used interpolation to resample the data into the same size, which is 26*130*130. Then for Scikit-learn, we convert the data into 1-d arrays. Since we found that the accuracy for this multi-label classification problem can be very low, we first attempted to reduce the dimensions of the data we fed in and only used the information of certain layers of the original data to train the model, which actually gave us no improvement. We then tried to build six classifiers that classify each class against the rest to divide it into sub-problems. However, after we have trained a few models and tested the result, we then found that while the accuracy is around 80%, the ROC-AUC is very low, around 0.5. After checking the predicted results and the true testing set labels, we found that the classifiers are predicting the 1s (the class we want to classify against the others) as 0s for most of the times and the accuracy is still high since most data in the testing data belongs to other classes. Moreover, the ROC-AUC score changed a lot when we repeated the experiments. In our thoughts, this can be related to both the scarcity of data and the distribution of different classes. There could be cases in which most of the data that belongs to a certain class are in the testing set and only a few are in the training set. There could also be the cases in which the distribution of the other classes we are classifying against is different in training and testing set. Thus we decided to manually generate our new training data and testing data, which uses about 80% of the whole data to train and 20% of the data to test and keeps the proportion of each class approximately the same in both sets. It is worth noting that there are only 6 patients labeled as 'D', which made it hard to be classified. The other class that is difficult to classify according to our result is class 'B', which also has low accuracy and ROC-AUC score.

2) Result: Following is the result we got using Gradient Boosting Classifier, SVM with different weights and kernels, and Random Forest Classifier.

a) Gradient Boosting:

Class(number of estimators)	ROC-AUC	Accuracy
A(5000)	0.6342	0.7708
B(5000)	0.5000	-
C(1500)	0.6066	0.7083
D(5000)	0.7500	0.9792
E(4600)	0.6767	0.8958
F(6300)	0.5833	0.7083

Note: For class B, the ROC-AUC score is always 0.5 no matter how many estimators we use. Therefore, we didn't report accuracy score since it is useless for this class of labels.

b) SVM: The table below records the best SVM score for each class.

*Equal Contributions

Class(SVM kernel)	ROC-AUC	Accuracy
A(sigmoid)	0.6211	0.7500
B(linear)	0.5000	0.875
C(linear)	0.6011	0.7708
D(linear)	0.5000	0.9583
E(sigmoid)	0.5000	0.8958
F(sigmoid)	0.5000	0.7500

c) Random Forest: The best result is class 1, with 140 estimators. We got a ROC-AUC score of 0.55 and accuracy of 0.8125.

Other classes have too low of a ROC-AUC score.

d) Summary: Based on our experiment, GradientBoostingClassifier tends to be the best for our task and yields a relatively high ROC-AUC score for almost all the classes. Given that GradientBoostingTree is robust to overfitting, we can set n-estimator to large numbers and still get ROC-AUC score over 0.6. In contrast, SVM gives relatively high ROC-AUC scores only for class A and C and Random Forest is not suitable for most classes. The improvement of the result after we generate the new training and testing set also showed us who greatly the amount and distribution of the input data can affect the classification result.

B. Deep Learning Approach

1) Method: Our interpretation is that (1). the dataset is too small. There are only 229 images in total, and our train-test split is 179:50. Moreover, each image in the dataset has 439400 pixels due to its 3d representation. (2). the dataset is too unbalanced. We have 6 classes of labels in the data, but their distribution is really sparse and unbalanced, with number of images in each class (A to F) being 50, 32, 57, 6, 24, 60. Therefore, it is obvious that class D has way less images than other classes, and class B and E also have relatively low number of samples than the other three. As a result, it is reasonable to conclude that the model can not get enough training in three of the six classes. In the extreme case, all 6 images of class D may be in the test set, yielding an accuracy of 0% for class D. (3). Some classes of stroke pattern are similar. For example, class C and class F in figure 1. have lesion located at areas relatively close, and the size of lesions in those two classes are of similar scale. After we flatten the image into 1d array representation, it is rather difficult to train the network to recognize the difference in these two classes given the small number of training images available.

To solve this problem, we applied the same approach to process data as described above for other methods. We first labelled class A as 0 and all other classes as 1, and train the CNN in this binary classification setting. Then repeat the same process for the other 5 classes. In this way, the CNN achieved on average of 83% accuracy.

2) CNN Architecture: The input of our CNN is images of size $130 \times 130 \times 26$, with 1 color channel. We used 3 convolutional layers with filters of size of 32, 2, and 2 respectively. Each convolutional layer is followed by a max pooling layer of size 2 by 2 by 2. In the end, we used 3 fully-connected layers which reduced the dimension from 25088 to 2 for binary class classification. We trained with one image at each time(batch-size = 1) due to the constraints of small size of the dataset and computational power of our machine. The visualization of the structure is shown in figure 1.

3) Results: We first tested the CNN on class A, and got 83% accuracy averaged over 8 runs. However, the ROC-AUC score is 0.5 for all the runs. It turned out that the model just predicted all the test images to have label 1, so that the errors in prediction would only appear for the original class A with label 0. Since the number of a single class is always less than the sum of all other classes, this prediction will generate high accuracy, but as a result, the ROC-AUC score of it will always be 0.5, which means that the prediction cannot be generalized for other datasets even though the accuracy is high.

As for the reason of its poor performance, we still suspect that the scarcity and unbalanced grouping of the data play the most significant role.

IV. Conclusion

The problem for our project is that out dataset is too small for a deep neural network to learn the actual classification patterns. On the contrary, even though this is an 3D image classification task, traditional machine learning algorithms such as gradient boosting outperforms deep CNN.

In the future, we will first try to gather more data for training using techniques such as data augmentation, which is a method to enlarge the dataset by doing linear transformation to images in the current dataset. By using more training samples, we have a faith in CNN to perform better in this multi-class classification task.

Moreover, the promising results we got from traditional machine learning methods showed a strong probability that machine learning can actually be applied to do pre-processing for DWI image of acute ischemic strokes.

Acknowledgment

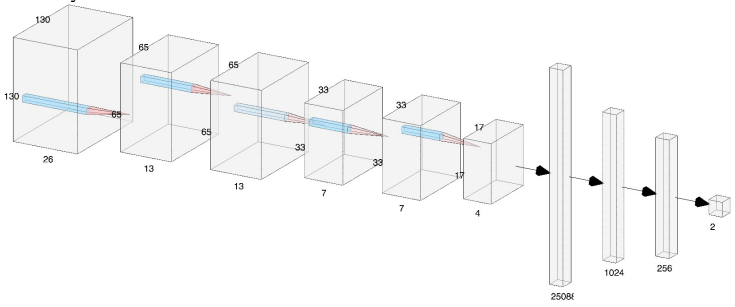
The authors would like to thank Professor. Fabien Scalzo at University of California, Los Angeles for the brilliant ideas and data.

References

[1] Van Everdingen KJ, van der Grond J, Kappelle LJ, Ramos LM, Mali WP. *Diffusion-weighted magnetic resonance imaging in acute stroke.* Stroke. 1998; 29:17831790.

[2] Bang OY, Lee PH, Heo KG, Joo US, Yoon SR, Kim SY. *Specific DWI lesion patterns predict prognosis after acute ischaemic stroke within the MCA territory.* J Neurol Neurosurg Psychiatry 2005;76(9): 12221228.

Fig. 1. The structure of CNN we designed. The input is of size 130 x 130 x 26, and we used 3 sets of conv-maxPooling layers followed by three fully connected layer which reduced the dimension to 2 in the end.



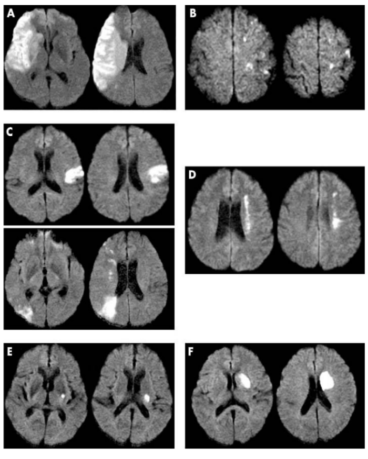


Figure 2. Diffusion-weighted imaging lesion patterns: (A) territorial infarcts, (B) small superficial infarcts, (C) other cortical infarcts, (D) internal border zone infarcts, (E) small deep infarcts, and (F) other deep infarcts. [Bang. et al]