DLMI midterm report
Intracranial Hemorrhage prediction

Gi-Luen Huang
National Taiwan University, Taiwan
Graduate Institute of Communication Engineering
r09942171@ntu.edu.tw

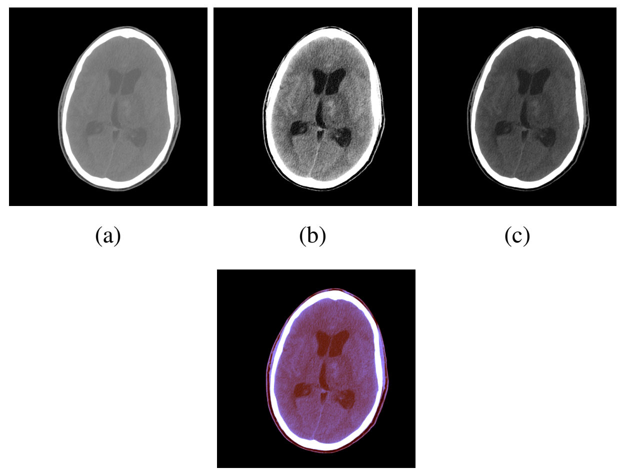
I-Hsiang Chen
National Taiwan University, Taiwan
Graduate Institute of Electrical Engineering
f09921058@ntu.edu.tw

Abstract

In this midterm project, our task is intracranial hemorrhage prediction. Given a Dicom file, we should determine whether the image is a positive sample or not. If it is a positive sample, we should further determine which class of sick he/she obtains. Because the image may not have only one class of sick, our task is a multi-label classification. In our method, We can divide our steps into three: preprocessing, training, and testing. In the preprocessing phase, we apply the three types of windows: brain window, subdural window, and bone window. After that, We concatenate them into an image with 3 channels. In the training phase, we randomly split the training data into train/val with the ratio of 8:2. We utilize the two models, Resnext50 and SEresnet50 with pretrained weight from ImageNet, to train and validate the results. The loss function is weighted BCEloss, to solve the problem of the unbalanced data. In the testing phase, we use the ensemble model with Resnext50 and SEresnet50 to predict the class of sick. Finally, we get the f1_score 0.72167 on the public dataset.

1. Introduction

With the rapid development of modern medicine, many diagnostic instruments have been released, and the most important one is a Computerized Tomography (CT) scan. The CT image is established for the human body by rotating X-rays scanning. Due to the human tissue is made up of many different components which have different densities and penetration coefficients, so the degree of absorption of X-rays at each point is different. In fact, the X-ray will be received by a highly sensitive sensor and converted it into post-digital signals, then computing attenuation coefficient of each point by a highly complex mathematical algorithm. Therefore, the CT imaging process needs to scan from multiple directions and convert the values into 2D slices of body part. This process will repeat times to provide complete diagnostic information. Finally, doctors can use this information to diagnose whether there are hidden diseases in the human body.



(a) (b) (c)

(d)

Figure 1: Example of extracted image. (a) Bone window (b) Brain window (c) Subdural window (d) Concatenated image

Nowadays, the diagnosis for medical imaging data still relies on the doctor's visual observation. The results of the diagnosis are based on the doctor's subjective conclusions and experiences. In addition to this, each CT scan will product over 500 images, which requires large consuming review by the doctor and the accuracy will decrease with long hours of work. The cumbersome inspection process and the large amount of CT data are exactly suitable for deep learning in computer vision. In the recently, there are many methods are training on 3-channels imaging which constructed from fixed window filters. Therefore, we will use the (Deep Convolution Neural Network, DCNN) architecture for the intracranial hemor-rhage prediction task.

On the other hand, the most of pathological datas are health samples, it may cause the serious data imbalance, and there may be multiple diseases at the same time. We

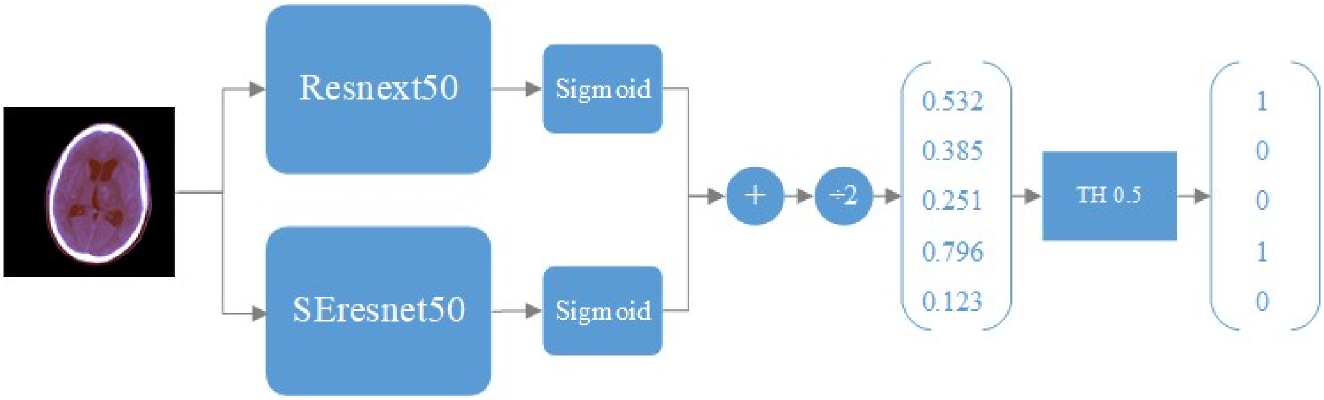


Figure 2: Example of testing procedure.

make a series of argumentations for data preprocessing, and use weighted BCE training for multi-category prediction. By the attention to this, there are many misjudgments and sample offsets in reality. We adopt the dual path projection to simulate the effect of multiple diagnosis, and integrate the results to obtain a more robust result. The major contributions of this work are as follow:

- We introduce a powerful dual-path embedded architecture. Inspired by the re-examination mechanism, we train two independent and powerful feature extraction backbones respectively. In the testing stage, the ensemble mechanism is used to integrate two backbone's predictions, which ensure the final prediction result is certified by both observers. In the final experimental stage, we show that this mechanism has a significant improvement in F1score.

- We adopt weighted BCE loss function to solve the problem of data imbalance, each weight is defined by the proportions of positive and negative for each category. In addition, we also adopt a series of data argumentation to reduce the impact of hard sample in the pre-processing stage. After all, this strategy improve the overall accuracy and make the model more robust.

2. Methods

Our method can be divided into three steps: preprocessing, training, and testing. We'll introduce above three steps below.

2.1. Preprocessing

Dicom file has many information about the patient. We extract the some of information on it, including "SOPInstanceUID", "StudyInstanceUID", "PixelArray", "intercept", and "slope". First, we transform the Dicom format image into a CT image according to "intercept" and "slope". The transformation formula is shown below:

$$x' = x * slope + intercept$$

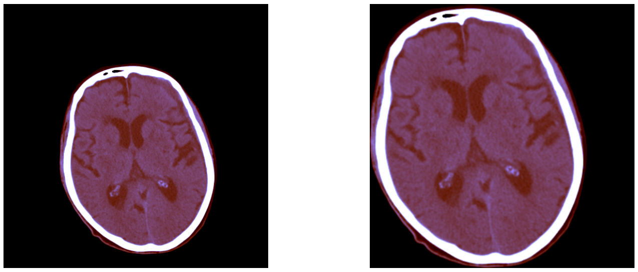
Where x is the dicom format image and x' is the CT image. Second, we apply three windows to extract the information on the CT image: brain window, subdural window, and bone window. Among them, the center and width of the three windows are 40/80, 80/200, and 40/380, respectively. Subsequently, we normalize each extracted window by using min-max normalization. The min-max normalization is shown as below:

$$w' = (w - min(w))/(max(w) - min(w))$$

Where w is the extracted window and w' is the normalized window. We concatenate the three windows as one image and take it as our training image. The example of extracted image is shown in Figure 1. Furthermore, because each Dicom file has a unique "SOPInstanceUID", we use it as the name of the training images. Finally, we randomly separate the training data into train and validation with the ratio of 8:2. Therefore, we have 74,092 training data and 18,524 validation data, respectively.

2.2. Training

Before training the model, we had some preprocessing on the training data. To make the model more robust and see more data, we apply three data augmentation techniques on the data: RandomHorizontalFlip, ShiftScaleRotate, and Transpose. In addition, we find that the image has many black edges, we also use the method of black edge cropping. The example of black edge cropping is shown in Figure 3. Because the number of classes in training data is unbalanced, we apply the weighted Binary Cross-Entropy(BCE) loss function to penalize the larger number of classes. The



(a) (b)

Figure 3: Example of black edge cropping. (a) Before cropped (b) After cropped

weighted BCEloss can be defined as:

$$L = -\frac{1}{N}\sum_{i=1}^{N} y_i \cdot \log\left(p(y_i)\right) + (1 - y_i) \cdot \log\left(1 - p(y_i)\right)$$

$$W_i = p_i / n_i, i = 0, 1, ..., K - 1$$

Where N is the number of data, y_i is the corresponding label, $p(y_i)$ is the prediction of the model's output, W_i is the weight of the i_{th} class, p_i and n_i are the number of positive and negative samples, and K is the number of classes. Furthermore, we set a threshold value, which is a hyperparameter, to decide whether the output value of the model is positive sample or not. If the output value is equal to or larger than the threshold value, we will change the output value to 1, and vice versa. After experiments, the threshold value is 0.5 in our work. Finally, We train the models resnext50 and SEresnext50, respectively.

2.3. Testing

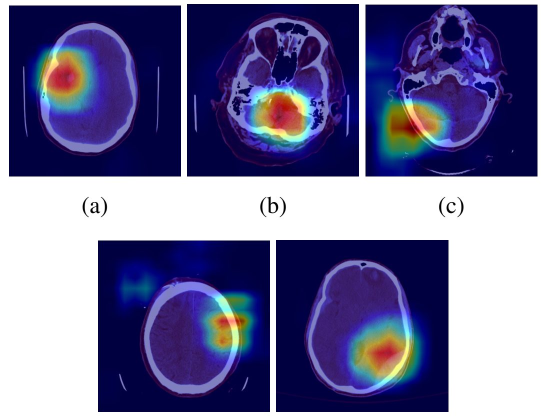
First, we apply the same preprocessing method (described in section 2.1) on the testing data. Because the ensemble model can usually get better performance than only one model, we apply the ensemble model to predict the results of testing images. The ensemble model includes the two models, resnext50 and SEresnet50. We give the output of these two models the same weight and set the threshold value 0.5 the same as the training part. The testing procedure is shown in Figure 2. The ensemble model's output can be formulated as follow:

$$y = \frac{1}{2}\sum_{i=0}^{1} Sigmoid(M_i(x))$$

Where y is the output of the ensemble model, M_i denotes the i_{th} model within the ensemble mode, and x is the input image.

3. Experiments

We try two different methods of separating training data. One separates the training data randomly, and another separates the training data according to the patient's ID. The results are shown in Table 1 and Table 2. We also use the Grad_cam[1] to demonstrate the visualization of each positive sample. The example can be seen in Figure 4.



(a) (b) (c)

(d) (e)

Figure 4: Example of the Grad_cam visualization. (a) Intraparenchymal (b) Intraventricular (c) Subdural (d) Subarachnoid (e) Subarachnoid

3.1. Dataset

The dataset contains labeled data and unlabeled data, which are 92,616 and 55,532 images, respectively. There are 5 classes: subarachnoid, epidural, intraventricular, intraparenchymal, and subdural. The positive samples are 4196, 390, 3260, 4616, and 6208, respectively. Therefore, there is a critical problem we need to solve – data imbalance.

3.2. Implementation details.

Our method is implemented via Pytorch and is run on a server with two NVIDIA GeForce RTX 3090 GPUs and two NVIDIA Tesla V100 GPUs holding a graphics memory capacity of 24GB and 32GB for each one, respectively. We first resize the image of size $(w, h) = (512, 512)$. We adopt the Adam optimizer with a learning rate of 5 x 10^{-5} and exponential decay rates (β_1, β_2) = (0.9, 0.999) for the moment estimation. The batch_size and epoch are 48 and 100, respectively. We also apply the step learning decay scheduler, decaying the learning with 0.412 every 50 epochs.

4. Discussion

This subsection will discuss the effectiveness of this work. In order to effectively analysis the actual effectiveness of the proposed method, we use the F1 score metrics for the evaluation. F1 score considers both precision and recall, which can effectively reflect the problems of misjudgment and omission. As the result from Table 1, we can

Model(f1_score)	Validation	Public
Resnet18	0.82	0.66583
Resnext50	0.84	0.69819
SEresnet50	0.84	0.69075
Resnext50 and SEresnet50	**0.86**	**0.72167**

Table 1: The results of randomly separating the training data.

Model(f1_score)	Validation	Public
Resnet18	0.80	0.65730
Resnext50	0.83	0.70002
SEresnet50	0.82	0.69023
Resnext50 and SEresnet50	**0.84**	**0.71544**

Table 2: The results of separating the training data according to the patient's ID.

find that resnet50 and serenest50 have almost the same upper limit capacity. However, when we use our dual-path embedded mechanism, the F1 score is increased by 3% in the public dataset. As we mentioned at the beginning, there are varying symptoms at different patients, and the milder ones are often be neglected. However, it can effectively improve the patient's recall and increase the overall accuracy by the feature of dual-path.

On the other hand, in order to observe the effectiveness of the model. We observe the attention feature through Grad-cam visualization in Figure 4., and we can also find that have indeed paid attention to the corresponding pathological feature location in each different case. In addition, there are different scale and angle of each patient's lesion, such as the angle variance in Figure 4(b). and the scale variance in Figure 4(d). Therefore, effective data argumentation has a great impact on the robustness of the model, and we can indeed effectively adapt to a variety of different patient information.

5. Conclusion

To conclude, this work presents a powerful dual-path embedded architecture. We leverage two strong feature extractor backbone, and integrate two predictions to make the model be more robust. To solve the data imbalance problem, we adopt a series of data pre-process and use weighted BCE Loss for training, and we achieve the highest F1-score in the public dataset. In the future, we can explore the capabilities of our model on different tasks.

References

[1] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017.