

HOHOHO: intracranial Hemorrhage detection enhanced by asymmetric loss with CNN-LSTM

Group 23

Ming-Yang Ho, Bin-Ray Wu, Hsin-Yu Ho, and Si-Yang Jiang

Introduction

- Cerebral hemorrhage, bleeding that occurs around or within the brains, is a serious health problem requiring rapid and often intensive medical treatment.
- The cerebral hemorrhage can be divided into 5 categories: Intracerebral hemorrhage (ICH), Intraventricular hemorrhage (IVH), Subarachnoid hemorrhage (SAH), Subdural hemorrhage (SDH), Epidural hemorrhage (EDH).
- While the diagnosis requires an urgent procedure, the process is complicated and often time-consuming. Herein, this problem is attempted to be solved by learning-based methods.

Methodology

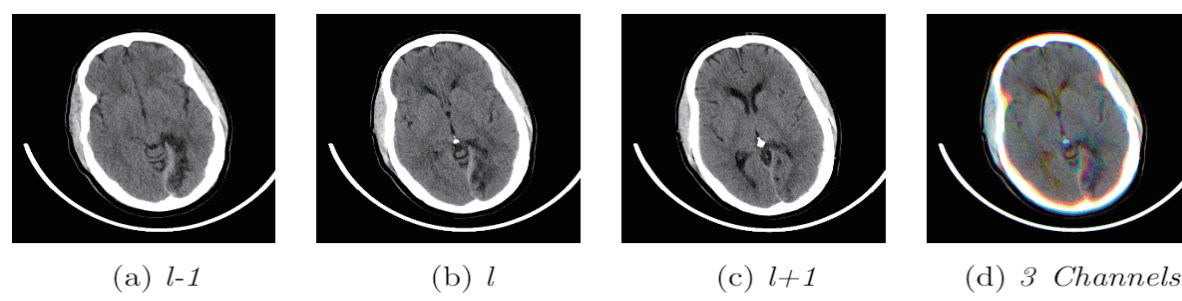


Figure 1. Example of stacking CT image

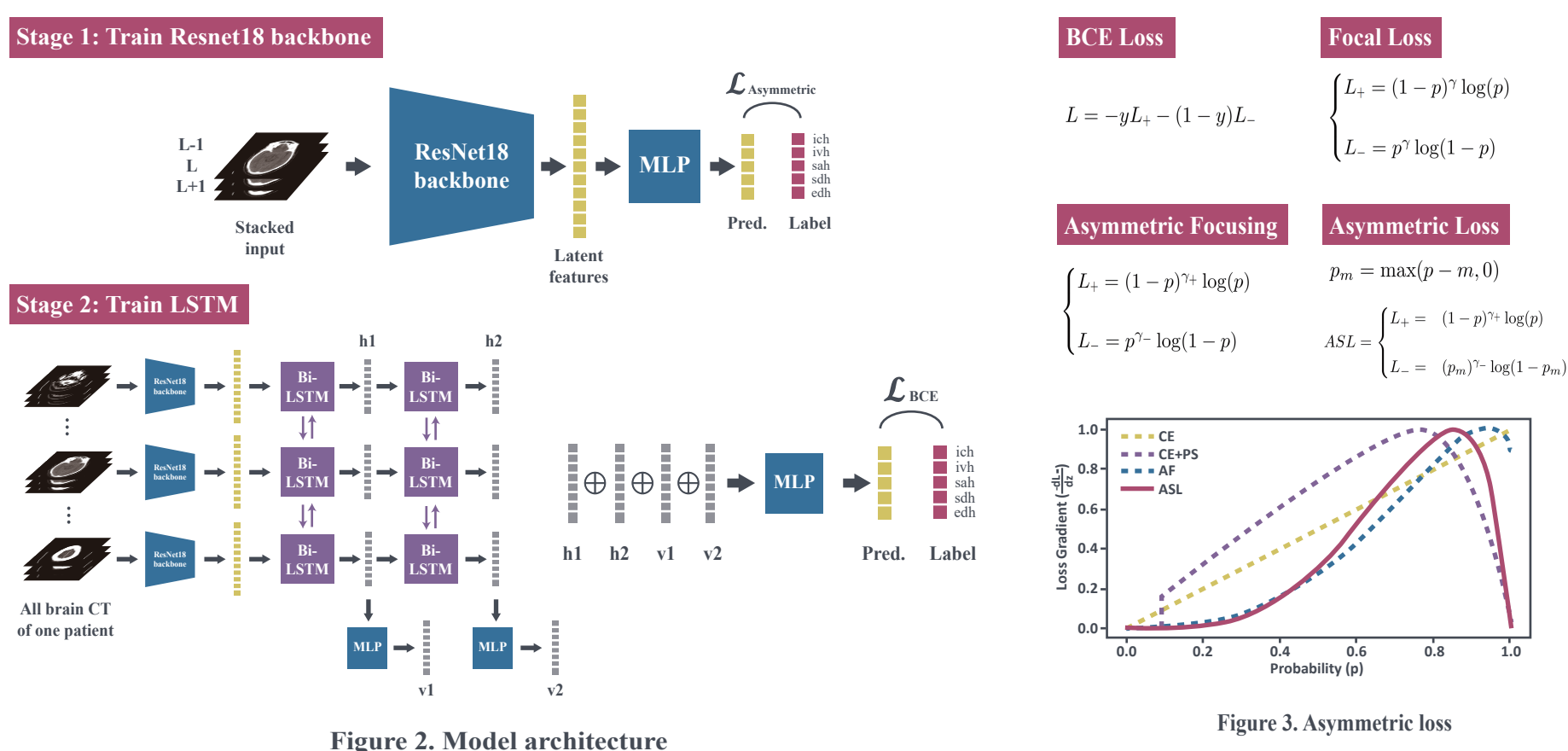


Figure 3. Asymmetric loss

A. Full Training Dataset

- One CT image was firstly stacked with two aside ones in the preprocessing step to extract more information owing to the property of sequential CT scanning (Fig. 1).
- Limited augmentation strategies, rotation with little color adjustment, were utilized to avoid interfering with the intrinsic CT data distribution.
- ResNet-18 was utilized as the features extraction backbone trained with asymmetric loss, which is an improved focal loss designed for unbalanced positive and negative samples training. A tunable parameter gamma can control the model to focus on positive samples. Furthermore, there was a clip design, which would prune the class if the confidence of the class was extremely excessive (Fig. 2 and 3).
- Finally, a stacked LSTM architecture trained with BCE loss utilizing 512-dim embedding from ResNet-18 was leveraged to further enhance the overall performance, which thoroughly considers a patient at the same time (Fig. 2).

B. Small Training Dataset

- Due to the label deficiency in the small dataset, a self-supervised method, SimCLR, which enables distinguishing the positive and negative samples without labels via appropriate data augmentation, was utilized to overcome this problem. Afterward, the pre-trained weight from SimCLR was fine-tuned by the labels from the small dataset (Fig. 4).

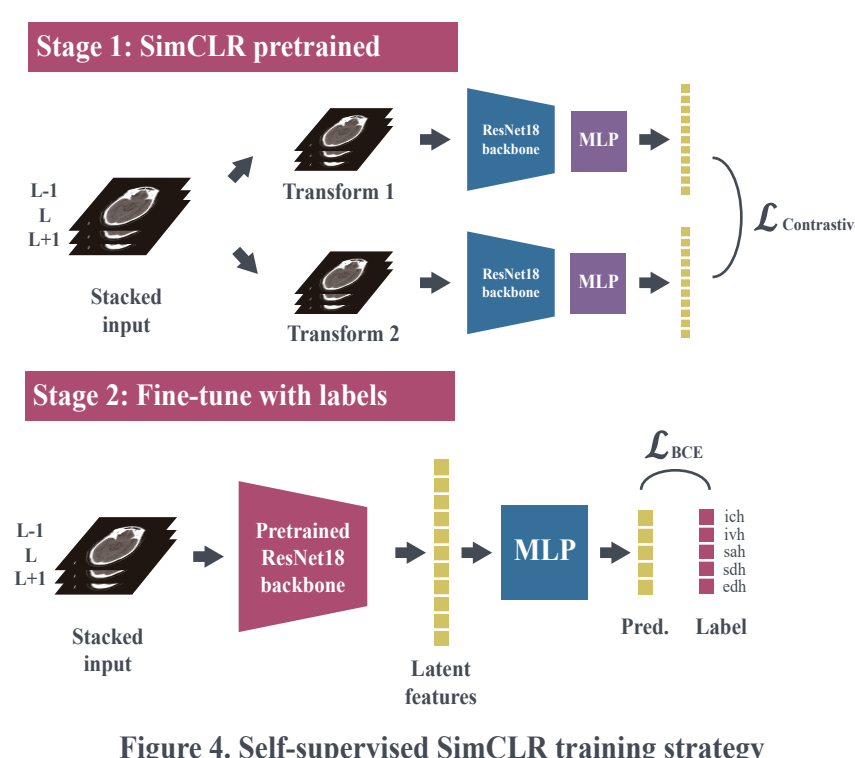


Figure 4. Self-supervised SimCLR training strategy

Result

	Adam		SGD		SGD No nesterov	
Batch size	64	40	64	40	40	40
Color jitter	0.5	0.1	0.1	0.2	0.1	0.1
Rotation	20	20	20	30	30	30
Gray	Train f2	0.8496	0.7273	0.8612	0.8140	0.7480
	Val f2	0.6892	0.7045	0.7064	0.7143	0.7155
Stacked	Train f2	0.7208	0.7986	0.8140	0.8408	0.7726
	Val f2	0.7011	0.7000	0.7143	0.7307	0.7153

Table 1. Comparison results of vanilla ResNet-18 under different hyperparameters setting

	Adam with asymmetric loss					
Color jitter	0	0.1	0.2	0.1	0.1	0.1
Rotation	20	20	20	30	40	50
Train f2	0.843	0.842	0.829	0.865	0.843	0.854
	Val f2	0.741	0.743	0.729	0.781	0.773
Stacked	Train f2	0.743	0.750	0.735	0.736	0.749
	Val f2	0.743	0.750	0.735	0.736	0.749

Table 2. Comparison results of ResNet-18 with asymmetric loss under different hyperparameters setting

	ResNet 18 (rotation 20)		ResNet 18 (rotation 40)	
Batch size	4	64	64	128
LSTM units	64	64	64	64
Train f2	0.8907	0.8943	0.8957	0.8988
	Val f2	0.7714	0.7714	0.7917
Stacked	Train f2	0.7792	0.7787	0.7764
	Val f2	0.7792	0.7787	0.7764

Table 3. Comparison results of stage 2 LSTM

Attempt	ResNet-34 With Adam BCE Loss	ResNet-34 With SGD BCE Loss	ResNet-50 With Adam BCE Loss	ResNet-50 With SGD BCE Loss	DenseNet121 With Adam BCE Loss	DenseNet121 With Adam BCE Loss
Train f2	0.870	0.979	0.687	0.832	0.818	0.796
	Val f2	0.637	0.647	0.637	0.712	0.731
Stacked	Train f2	0.717	0.708	0.698	0.716	0.735
	Val f2	0.717	0.708	0.698	0.716	0.735

Table 4. Other attempts with vanilla ResNet and DenseNet

Attempt	ResNet-18 Asymmetric Loss Related with 0.5	ResNet-18 Asymmetric Loss Related with 1	ResNet-18 Asymmetric Loss Filtered data	ResNet-18 Asymmetric Loss Magic normalization	CNN-LSTM END2END
Train f2	0.812	-	0.881	0.848	0.824
	Val f2	0.757	0.769	0.753	0.759
Stacked	Train f2	0.700	0.744	0.727	0.743
	Val f2	0.700	0.744	0.727	0.743

Table 5. Other attempts with multifarious tricks

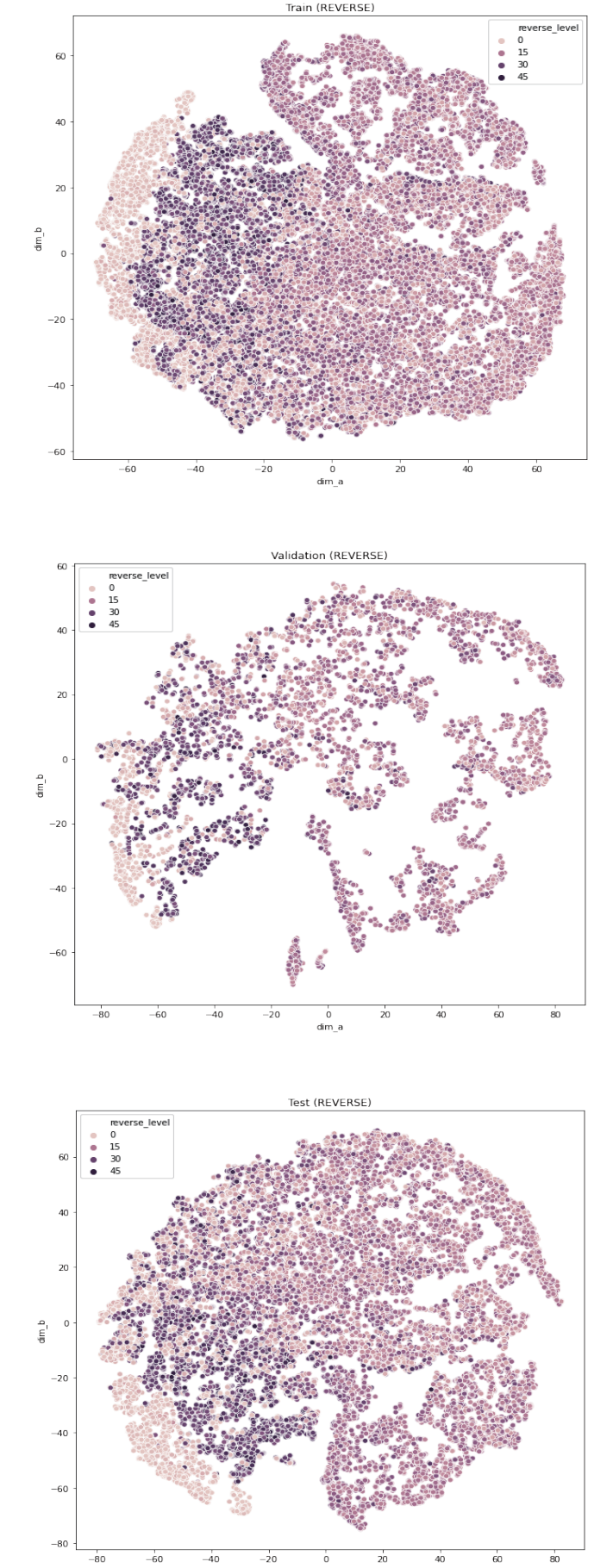


Figure 5. Embedding visualization by t-SNE

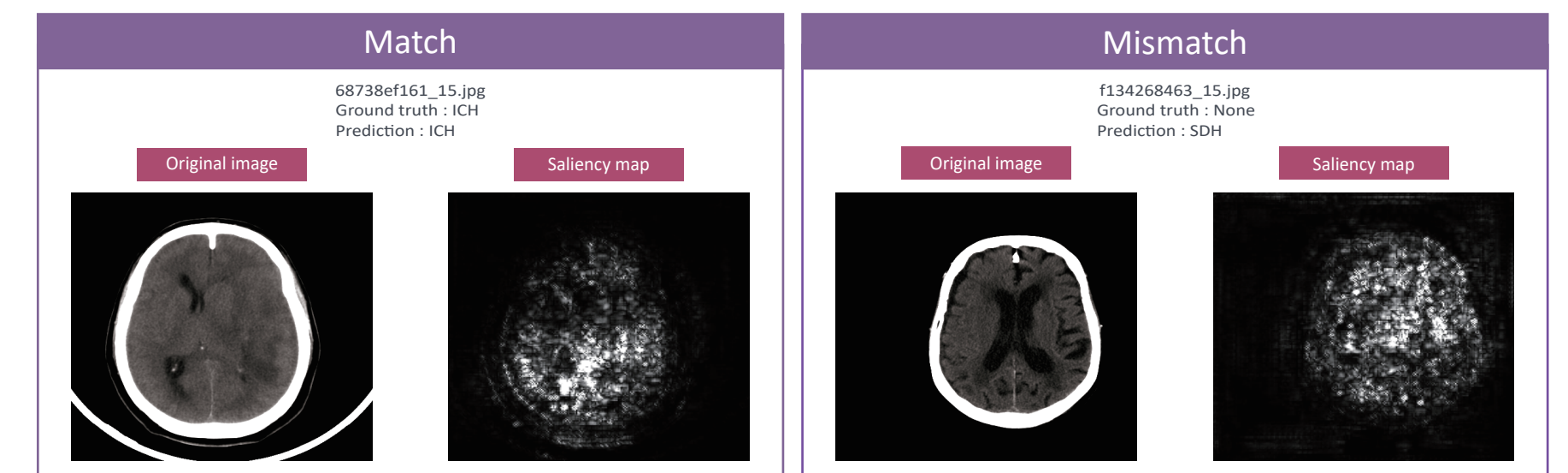


Figure 6. Original image v.s. saliency map

- Multitudinous optimizers and augmentation parameters were firstly investigated with vanilla ResNet-18 trained with BCE loss, which demonstrated the detrimental effect of either large batch size or color adjustment (Table 1).
- Asymmetric loss was an effective strategy to further improve model performance (Table 2).
- Two-stage training with stacked LSTM enabled approximate 2% f2 enhancement (Table 3).
- Multifarious strategies were also leveraged but the endeavor was in vain (Table 4 and 5).
- The embedding itself did contain level information (Fig. 5).
- An explainable saliency map was utilized to elucidate what did the model learn (Fig. 6).
- The best results were shown in (Table 6).

Dataset	Full	Small
Best	Train f2	
	Val f2	
	Test f2	

Table 6. Best results

Discussion and Conclusions

- Stacked CT images, asymmetric loss, and a further LSTM model were actually competent strategies to tackle imbalanced sequential CT data.
- Strange patterns did occur in the sequential labels, which were potentially erroneously mislabeled and somehow deteriorated the model performance.
- However, the model could still rectify these serious mistakes, which definitely manifests the benefits of using deep learning techniques in clinical scenarios (Fig. 7).

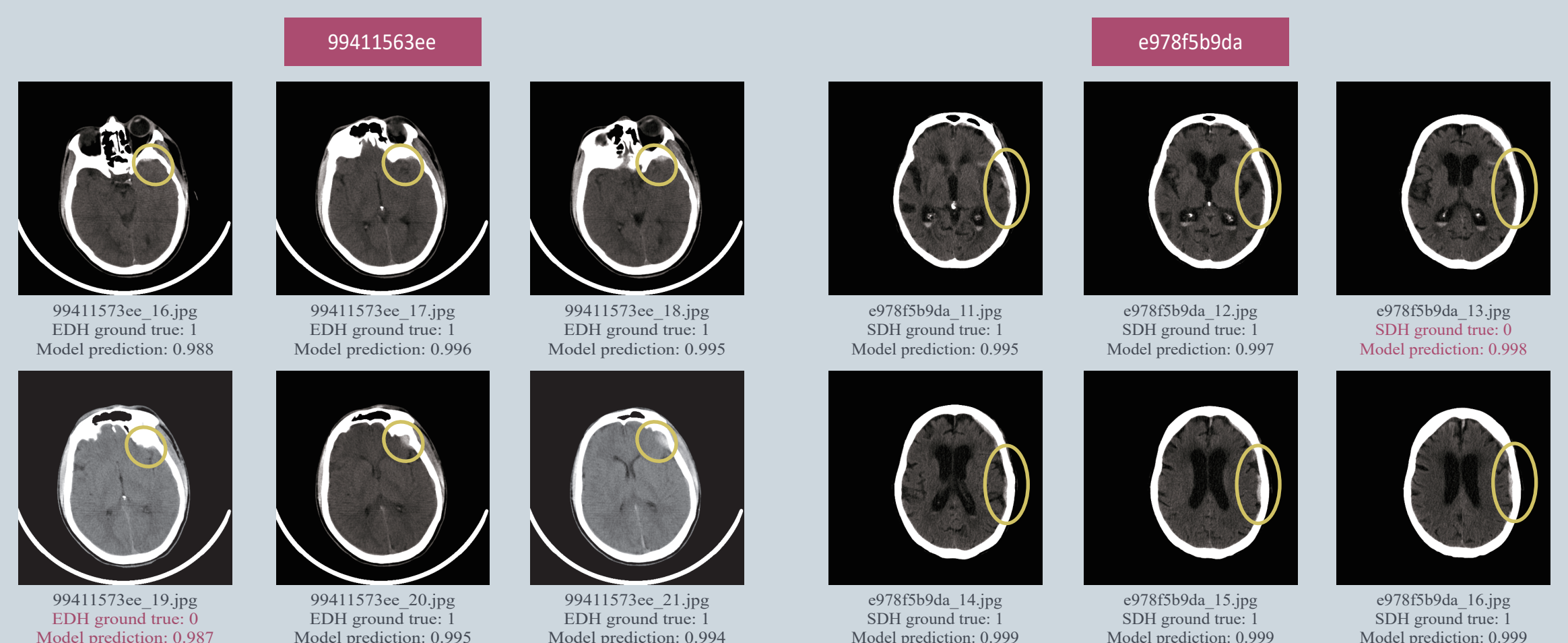


Figure 7. Example of potentially mislabeled CT image