CS230

Intracranial Hemorrhage Classification using CNN

Hyun Joo Lee
Department of Mechanical Engineering
Stanford University
hjlee22@stanford.edu

Abstract

In this study, multi-class classification is conducted to diagnose intracranial hemorrhages and its five subtypes: intraparenchymal, intraventricular, subarachnoid, subdural, epidural. Transfer learning is applied based on ResNet-50 and linear windowing is compared with sigmoid windowing in its performance. Due to the high imbalance in the number of examples available, an undersampling approach was taken to provide a better balanced training dataset. As a result, the combination of sigmoid windowing and combining three windows of interest showed the highest F1 score. Through error analysis, it was shown that the correlation of subtype and diagnosis of intracranial hemorrhage was not clear and next steps look toward changing the architecture into 2 steps, where the first step diagnoses intracranial hemorrhages, and the second step classifies subtype within the positively diagnosed examples.

1 Introduction

Intracranial hemorrhages have fatal consequences depending upon its subtype, location, and size. Radiologists are up to the task of evaluating CT(computed tomography) scans both accurately and promptly, since some instances require immediate attention. A fast and accurate classification using a machine learning algorithm well fitted to aid the current clinical workflow could provide critical assistance. In this project, I use a single slice of a CT scan in DICOM image format as input. Then, I preprocess the grayscale image to Hounsfield Units(HU) and select three different contrasts to highlight the window of interest. Next, I use the pretrained weights from imagenet for ResNet-50 and further tune the weights to output a binary classification for each of the classes, one class for any and 5 classes each representing subtype: intraparenchymal, intraventricular, subarachnoid, subdural, epidural. The image data set is provided by ASNR(American Society of Neuroradiology) via the kaggle competition, which is prelabeled by experts on the subtype of the intracranial hemorrhage. The code for this project is shared in [1]

2 Related work

In recent years, there has been a push towards domain specific classification algorithms. For intracranial hemorrhage classification, since HU units used by CT scans have (-1000,1000) range, whereas grayscale can only express (0,255), finding the correct windowing is crucial for an accurate diagnosis. In a clinical setting, practitioners use a predetermined system window setting, specific for each area of interest, i.e. "brain window", "subdural window". A recent study using deep learning has shown that the traditional window setting can be further optimized and lead to a more accurate diagnosis[2]. An approach using an ensemble of CNNS show that different architectures will focus on different features and therefore a fine-tuned combination would be promising[3], and 3D CNNs have also proven to be more precise in certain cases because it takes advantage of the spatial connectedness of hemorrhages [4]. In particular, by dividing the detection of intracranial hemorrhage and subtype classification into a 2 step process, they were able to detect intracranial hemorrhages in a 30 second

CS230: Deep Learning, Autumn 2019, Stanford University, CA. (LateX template borrowed from NIPS 2017.)

time frame which poses a great possibility to be implemented in a clinical setting [5]. Looking deeper into how deep learning algorithms could be integrated in clinical settings, an algorithm has also been developed to alert medical practitioners and prioritize urgent patients according to their classification[6].

3 Dataset and Features

3.1 Background : Hounsfield Scale

The Hounsfield scale represents the density of tissues by using the attenuation of X-ray, which is proportional to their physical density. In HU, water is 0, air is -1000 , and bone is +1000. To display CT scans, a contrast enhancement method called "windowing" is used. By choosing two parameters, window level(L) and window width(W), viewers are able to see the specified window([L-W/2, L+W/2]) mapped onto the full range of gray scale([0,255]). Anything above the window would be seen as white, and anything below would be seen as black. In particular, commonly used parameters for a brain window is L=40, W=80.

3.2 Dataset

The dataset is from Kaggle RSNA Intracranial Hemorrhage Classification competition round 1[7] and were labeled by experts. The images were in DICOM(Digital Imaging and Communications in Medicine) format, a standard format for handling medical images. This format contains network communications protocol that allows the use across different machines and the transfer of patient data. The total number of examples available were 674258, but there were two major sources of imbalance in the dataset. First, there were significantly less positive examples than negative(healthy) examples. Secondly, there was also an imbalance in the number of positive examples per subtype. In order to ensure efficient training, I equalized each training batch by under-sampling so that each subtype had equally 12.5 percent of positive examples and there were 37.5 percent of negative examples. However, since each example could belong to multiple subtypes, some subtypes had higher than 12.5 percent of examples.

Type	Number of examples	Number of training examples per batch
Positive	97103(0.14)	80
Negative	577157(0.86)	48
Epidural	2761	16
Intraparenchymal	32564	16
Intraventricular	23766	16
Subarachnoid	32122	16
Subdural	42496	16
Total	674260(1)	128

Table 1: Data set

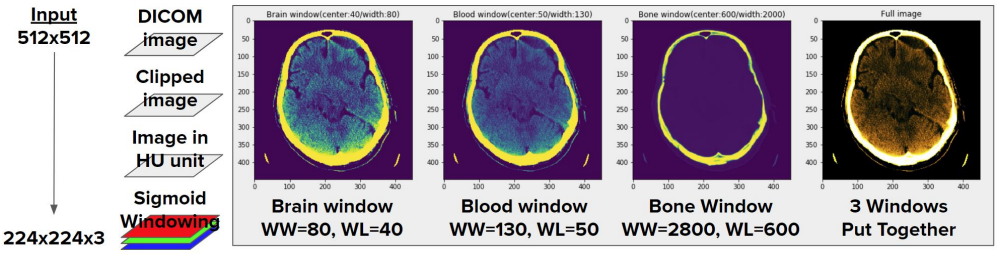

Figure 1: Overview of Preprocessing

Most of the images were (512,512), which is the most common image size for CT scans. Then, images was either padded or clipped resulting in image size of (448,448,1). The first step in preprocessing the data was to change the pixel values so that it is represented HU(Hounsfield Units). Then, I applied windowing for three windows: brain window(WW=80, WL=40), blood window(WW=130,WL=50), bone window(WW=2800,WL=600). By using sigmoid windowing, it was possible to also capture the data outside of the window instead of setting it all to zero.

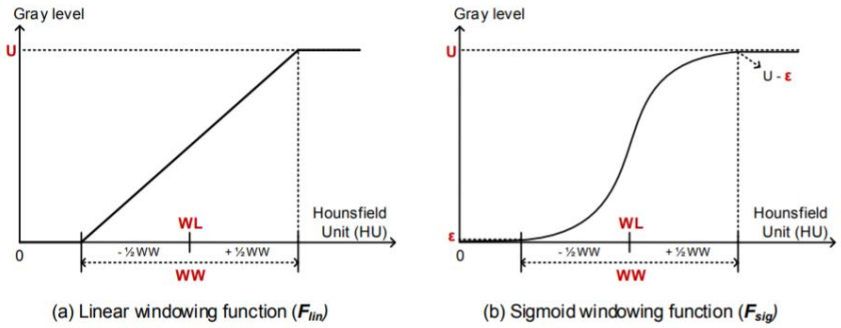

Figure 2: Windowing Methods: (a) Linear windowing (b) Sigmoid windowing: The constant U is the upper limit of windowing functions, and ϵ is the margin between the upper/lower limits and window end/start gray levels which determine the slope at the center. Figure from [2]

$$F_{sig}(x) = \frac{U}{1 + e^{-(Wx+b)}}, where W = \frac{2}{WW} log(\frac{U}{\epsilon} - 1), b = \frac{-2WL}{WW} log(\frac{U}{\epsilon} - 1) \quad (1)$$

Each of the resulting images served as separate channels, so that the output of the preprocessing step resulted in an image with size (448,448,3). Experimenting with the best size for efficient training, a max pooling step was optionally included and downsized the image to (224,224,3).

4 Methods

I add a global average pooling layer and a fully connnected layer on top of ResNet-50 to get a classification output of (m, 6). I use the pretrained weights from imagenet for ResNet-50 and further tune the weights to output a binary classification for each of the classes. Since each image could be multiple subtypes, binary cross-entropy loss function was used.

$$L(i) = -(y^{(i)} \log \hat{y}^{(i)} + (1 - y^{(i)}) \log (1 - \hat{y}^{(i)})) \quad (2)$$

where:
$\hat{y}^{(i)}$ = predicted label value for i th example
$y^{(i)}$ = true label value for i th example

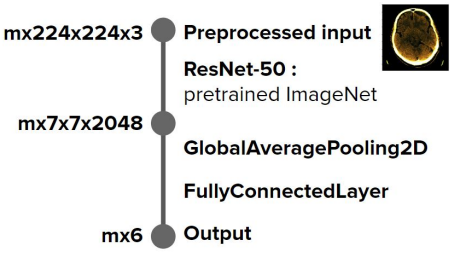

Figure 3: Model overview

5 Experiments/Results/Discussion

Table 2 shows the precision, recall, F1 scores evaluated on the test sets with each windowing applied in the data preprocessing phase. All models were trained with mini-batch size of 128 and for 10 epochs to fit machine capabilities and time duration for testing. When using 10 steps per epoch, the weights overfitted to the training set after 3 epochs and the validation loss did not go lower than 0.4. Then, the steps per epoch was raised to 40-80, which was found to have lowest validation loss at 10 epochs. Another path I took was to increase the image size from (224,224,3) to (448,448,3) while still keeping the steps per epoch as 10. Both methods had similar effects in that it provided more data.

	Any				Epidural				Intraparenchymal			
	P	R	F1	num	P	R	F1	num	P	R	F1	num
W1	0.14	0.28	0.19	587	0	0	0	19	0.05	0.03	0.04	198
W2	0.15	0.41	0.22	616	0	0	0	16	0.05	0.16	0.08	200
W3	0.15	0.38	0.22	585	0	0	0	21	0.06	0.08	0.07	193
W4	0.8	0.96	0.88	80	0.47	0.47	0.47	19	0.51	0.65	0.57	31
	Intraventricualr				Subarachnoid				Subdural			
	P	R	F1	num	P	R	F1	num	P	R	F1	num
W1	0.06	0.09	0.07	150	0.03	0	0.01	203	0	0	0	242
W2	0.06	0.15	0.08	158	0.06	0.04	0.05	212	0.06	0.01	0.02	281
W3	0.04	0.15	0.07	130	0.07	0.05	0.06	185	0.12	0.01	0.03	268
W4	0.83	0.17	0.29	29	0.56	0.44	0.49	32	0.73	0.31	0.44	35

Table 2: Performance summary(1): W1=Brain Window with linear windowing(WL=40,WW=80), W2=Subdural Window with linear windowing(WL=50, WW=130), W3=soft tissue window with linear windowing(WL=100, WW=220), W4=Bone window (40,80), Subdural window (50,130), Bone window (600,2800) put together in RGB format with sigmoid windowing, P=precision, R=recall

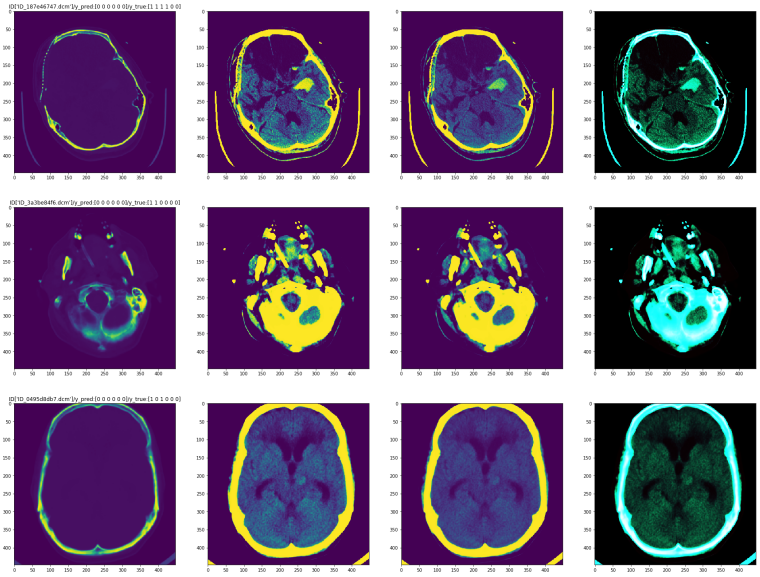

Figure 4: Error analysis: These were the first three images that were mislabeled in the test set. It was shown that the architecture did not make a clear relation that there would be a true subtype if and only when it was an intracranial hemorrhage.

6 Conclusion/Future Work

The highest performing algorithm was chosen that had the lowest validation loss, which was the one using sigmoid windowing with 3 different windows. There were many studies using traditional clinical window setting values, and with linear windowing. Since using different windowing techniques resulted in significantly different results, the next step I would take would be to find the best window setting values for sigmoid windowing. Then, test the most successful previous models with the different window settings. In addition, I would make the classification into a 2 step process to first detect intracranial hemorrhages and then to classify the subtypes.

7 Contributions

This work is done solely by the author. The author would like to thank Jon Farrell Braatz II from Stanford University for mentoring throughout this project.

References

[1] Project github repository.

[2] Hyunkwang Lee, Myeongchan Kim, and Synho Do. Practical window setting optimization for medical image deep learning. *arXiv preprint arXiv:1812.00572*, 2018.

[3] A. Kumar, J. Kim, D. Lyndon, M. Fulham, and D. Feng. An ensemble of fine-tuned convolutional neural networks for medical image classification. *IEEE Journal of Biomedical and Health Informatics*, 21(1):31–40, Jan 2017.

[4] Kamal Jnawali, Mohammad R. Arbabshirani, Navalgund Rao, and Alpen A. Patel. Deep 3D convolution neural network for CT brain hemorrhage classification. In , volume 10575 of *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, page 105751C, Feb 2018.

[5] Hai Ye, Feng Gao, Youbing Yin, Danfeng Guo, Pengfei Zhao, Yi Lu, Xin Wang, Junjie Bai, Kunlin Cao, Qi Song, Heye Zhang, Wei Chen, Xuejun Guo, and Jun Xia. Precise diagnosis of intracranial hemorrhage and subtypes using a three-dimensional joint convolutional and recurrent neural network. *European Radiology*, 29(11):6191–6201, Nov 2019.

[6] Mohammad Arbabshirani, Brandon Fornwalt, Gino Mongelluzzo, Jonathan Suever, Brandon Geise, Aalpen Patel, and Gregory Moore. Advanced machine learning in action: identification of intracranial hemorrhage on computed tomography scans of the head with clinical workflow integration. *npj Digital Medicine*, 1, 12 2018.

[7] Rsna intracranial hemorrhage detection.