Computer-Aided Brain Abnormality Detection Using CAT Images of Brain

Nima Hamidi
Statistics Department
Stanford University
hamidi@stanford.edu

Ahmadreza Momeni
Department of Electrical Engineering
Stanford University
amomenis@stanford.edu

Abstract

The abundance of medical imaging data as well as recent advances of deep learning models have enabled researchers to build models that can perform better than radiologists in diagnosing diseases. Disease detection using MR images and CT scans have attracted a lot of attention. In this project, we explored different algorithms to predict *white matter loss* and *hemorrhage* using 3D CT scan images.

1 Introduction

CT scan images are one of the most popular tools for diagnosis in urgent situations such as head injury, or patients with stroke symptoms. One of the reasons that this tool is widely used in emergency rooms is its low acquisition time [5]. Also, the usage of the aforementioned tool for the emergency rooms has been growing in the past few decades [3].

CT scan images can be used to detect many kind of abnormalities, among which intracranial hemorrhage and brain stroke are two of the most important ones. Both these issues are extremely time-sensitive and failing to detect them on time can be life-threatening.

One other detectable abnormality using CT scans is chronic white matter loss. White matter disease is the wearing away of tissue in the largest and deepest parts of the brain. This tissue contains millions of nerve fibers, or axons, that connect other parts of the brain and spinal cord and signal nerves to talk to one another.

While these abnormalities are found only on a small fraction of CT scans, automating abnormality detection in head CT scan would significantly decrease the time to diagnose and facilitate treatment. This would in turn decrease morbidity and mortality consequent to stroke and head injury. In addition, such an automation would lower the need for experienced radiologists to interpret head CT scans, and finally may lead in less erroneous interpretation.

In this project, we try to develop a deep learning algorithm to detect the mentioned abnormalities in head CT scans: hemorrhage, and chronic white matter loss.

2 Related work

Recently, there has been huge advancements in automating diagnosis based on medical imaging. Rajpurkar et al. [4] have built an algorithm to detect pneumonia using chest X-rays. They have shown that the performance of the proposed algorithm exceeds the performance of practicing radiologists.

Wang et al. [6] propose an algorithm to segment brain tumors using manually segmented brain MR images. They approach this problem using a cascaded model to detect different proportions of tumors sequentially. Their model consist of three networks called WNet, TNet, and ENet to segment the area segmented by the previous network further more. The model is trained using MR images as well as masks obtained by manual segmentations.

Chilamkurthy et al. [1], in contrast, have developed models to predict several issues at the same time. They use larger datasets Qure25K and CQ500 to train the models. These datasets provide slice-level labels and so, traditional two dimensional models can be used for the network architecture. Two deep learning model as well as one random forest model are used to make the predictions. The authors have stacked three different windows(brain, bone, and subdural) of CT scan images to get one RGB image to feed the models. Windowing is a thresholding technique widely used to let the radiologists focus on different types of tissues of the brain.

In this project, we design algorithms that can predict white matter loss and hemorrhage using 3D CT scans. We thus have two binary labels for the entire scan, in contrast to the above-mentioned papers that have more elaborated labels.

3 Dataset and Features

3.1 Data

We use a dataset of 970 head CT scan images from 874 unique patients to train our model. This dataset was collected in January 2016 at Stanford Hospital. We randomly split the data into training set, validation set, and test set and we make sure that there is no patient overlap between these sets. The resulting sets include 675, 192, and 96 images respectively.

Each CT scan consists of a number of *slices*. Each slice is a 512×512 single-channel image. Each pixel is given in Hounsfield Unit(HU). The number of slices vary from image to image, but it is around 50-60 for most scans.

For each scan, we have a vector of 17 binary labels generated by a radiologist. After exploring the data, we noticed that most of the labels are not useful as there are less than 20 positive images per label. Moreover, we observed that labels with more positive examples are highly correlated with each other, since there is a causal relation between them. For instance, bleeding can cause other problems in the brain like herniation and so, we need to aggregate these labels to get one informative label. Hence, we ended up using two of the labels as our responses. As mentioned in previous sections, these two labels are hemorrhage and chronic white matter loss.

3.2 Preprocessing

As we will explain in the next section, we are using a pre-trained model which is trained over ImageNet. So, we have to feed the network with images of the same shape as the images in ImageNet. Therefore, we rescale all slices to 224×224 pixels.

The next challenge is that ImageNet images have exactly 3 channels, but our images have different number of slices. The number of slices is determined by the *thickness* of imaging process. Thickness is the physical distance between two consecutive slices. Since, it is challenging to work with variable-length inputs, we decided to standardize the thickness by choosing a fixed number of slices from each scan. Initially, we started with 20 slices and then, we replicated each image three times to get an RGB image. We then used the same network to extract features from these 20 images. This approach failed in the training phase due to memory constraints, even after decreasing the batch size to 4. So, we decided to pick 18 slices equidistantly spread from each other and then arrange them into groups of 3 images to get 6 RGB images and the network was then fed by these 6 images, instead. Figure 1 presents an example of one of these 6 images described above[1].

We also applied a standard thresholding method to the images to focus on soft-tissues. Due to the presence of different tissues in brain images, the pixel values of soft-tissue parts of the brain have far less variance than the pixel values of the entire image. This means that, by naively mapping these values into $[0, 255]$, different parts of the soft-tissue area(which is of great importance for us) will become indistinguishable from each other. So, we had to apply the following thresholding function

[1] Note that, due to confidentiality, we had to use a publicly available CT scan instead of one of the images in our dataset.

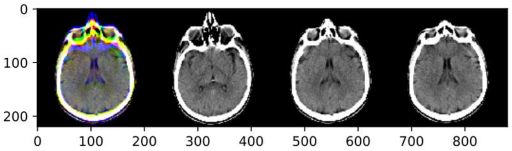


Figure 1: The image in the left is obtained by concatenating the three images on the right.

to all pixels to eliminate this issue:

$$f(x) = \begin{cases} 0 & x \leq 0, \\ \frac{255 \times x}{80} & 0 \leq x \leq 80, \\ 255 & x \geq 80. \end{cases}$$

Finally, due to the insufficient number of examples in our dataset, we utilize data augmentation techniques. In particular, random shift, random rotation, and random horizontal flip are performed during the *training*. We only allow small changes to ensure that the entire head stays in the image and the label remains unchanged.

4 Methods

In this section we will explain the architecture of our network. We followed the ideas proposed in [4] and we used a pre-trained model as the core of our network to extract low-level general-purpose features of the images. We used a 121-layer DenseNet[2] initialized from a model trained on ImageNet. We discarded the fully-connected layers on top of the DenseNet network as they seem more task-specific and irrelevant to our project. We then feed this network with each of the 6 slice groups mentioned in the previous section. Then, the corresponding 6 output tensors are passed to an average pooling to get one vector of the same size. The justification for this architecture is that the model should be invariant under small shifts along z-axis. By doing so, we have ensured this property (similar to the philosophy of convolutional neural networks) and also decreases the size of the parameters in the following layers.

We then added a fully-connected layer followed by dropout and batch normalization and another fully-connected layer with sigmoid activation. Figure 2 illustrates our network schematically. The output of the network is a vector of two numbers lying between 0 and 1 for the diseases.

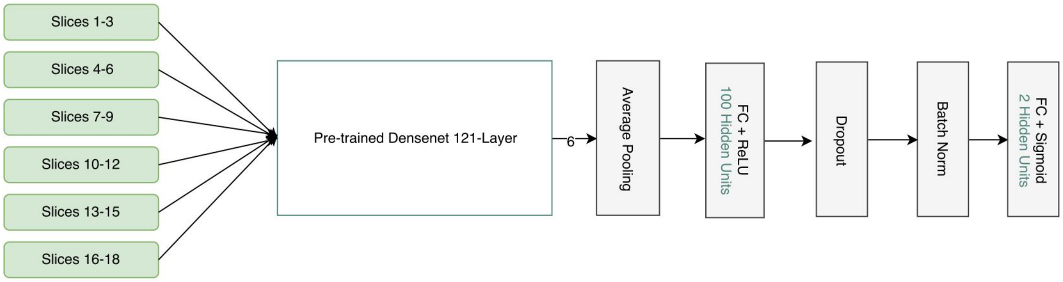


Figure 2: Architecture of the network used to train the model.

We used *weighted* cross-entropy as the loss function. This is due to the imbalance of positive and negative labels in the dataset. More specifically, the loss corresponding to example (X, y) for each of the two labels is

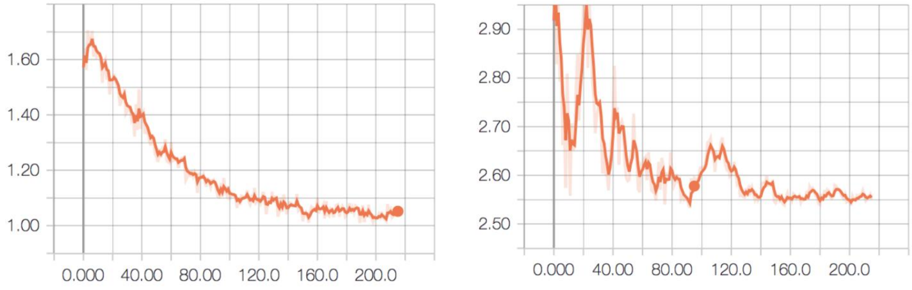
$$L(X, y) = -w_+ \cdot y \log p\left(Y = 1 \mid X\right) - w_- \cdot (1 - y) \log p\left(Y = 0 \mid X\right),$$

where w_+, and w_- are two weights, the ratio of which is equal to that of the number of negative, and positive examples in the dataset.

The network is then trained using Adam optimizer with learning rate 0.0001 and $\beta_1 = 0.9$ and $\beta_2 = 0.999$. We noticed that the network rapidly overfits to the dataset. Therefore, we froze the pre-trained model and added L2 regularization with $\lambda = 0.01$. This significantly improved the generalization of the model. We also used batches of size 8 to train the network. We didn't have much options for the batch size due to the memory limit issue. We trained the model for 200 epochs.

5 Experiments/Results/Discussion

We run our model on a server with two GeForce GTX 1080 Ti GPU's. We have plotted the training loss and the validation loss as a function of the number of epochs. The learning rate decays by a factor of 0.99 after every epoch. Figure 3 illustrated these plots.



(a) Training loss

(b) Validation loss

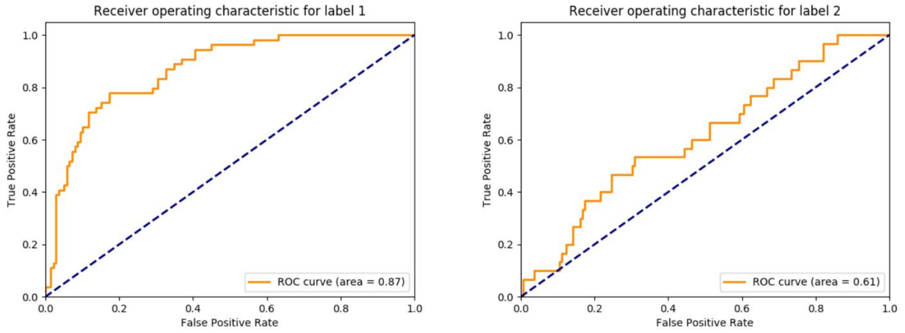
Figure 3: Training and validation loss as functions of number of epochs.

Table 1 shows the accuracies that were obtained over the training and validation dataset:

Labels	**Train acc.**	**Val acc.**
Chronic white matter	0.803	0.781
Hemorrhage	0.794	0.839

Table 1: Accuracies of the trained model.

Finally, Figure 4 shows the Receiver Operating Characteristic(ROC) plots along with the area under curve for each label over the validation dataset. Due to imbalance of positive and negative labels in our dataset, comparing accuracies is not sufficient. We thus plot ROC curves so that the performance of the algorithm can be observed for different thresholds.



(a) Hemorrhage

(b) Chronic white matter loss

Figure 4: ROC curves for hemorrhage and chronic white matter disease.

6 Conclusion/Future Work

As for future work, we have decided to do the following steps:

- **Collecting more data:** We will work closely with the collaborating research group in the medical school to label more *positive* head CT scans for each label. For this purpose, we will help them to pull the scans that seem to have the abnormalities from their database by running a script that reads the report of the head CT scans and infer they are possibly positive or not, and then we verify the labels with an experienced radiologist in the group.
- **Slice-level classification:** In order to cope with the difficulties related to the 3D nature of the scans and in order to increase the number of examples in the dataset we will perform the labeling process at slice-level instead of scan-level. As a result, in the future, we will design a network that gets only one slice as the input and detects any abnormalities only for that specific slice. In order to come up with diagnosis for the whole scan, we are considering using methods such as random forest to aggregate the output of scans.
- **Localization problem:** One of the importance questions in the realm of deep learning models is the interpretebility of the models. When it comes to medical imaging this issue becomes much more important as the models deal with human subjects. So, we will also relabel the dataset so that we can solve a localization problem which its output is more informative than a simple classification model.

7 Contributions

Nima has been responsible for implementing data preprocessing and designing a pipeline for feeding data into the network. Ahmadreza has been in charge of implementing different networks and experimenting. We have both put thought and time in brainstorming different ideas for tackling the problem and have had long discussion sessions with the collaborating research group in the medical school.

References

[1] Chilamkurthy, S., Ghosh, R., Tanamala, S., Biviji, M., Campeau, N. G., Venugopal, V. K., Mahajan, V., Rao, P., and Warier, P. (2018). Development and validation of deep learning algorithms for detection of critical findings in head ct scans. *arXiv preprint arXiv:1803.05854*.

[2] Huang, G., Liu, Z., Weinberger, K. Q., and van der Maaten, L. (2017). Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, volume 1, page 3.

[3] Larson, D. B., Johnson, L. W., Schnell, B. M., Salisbury, S. R., and Forman, H. P. (2011). National trends in ct use in the emergency department: 1995–2007. *Radiology*, 258(1):164–173.

[4] Rajpurkar, P., Irvin, J., Zhu, K., Yang, B., Mehta, H., Duan, T., Ding, D., Bagul, A., Langlotz, C., Shpanskaya, K., et al. (2017). Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning. *arXiv preprint arXiv:1711.05225*.

[5] Ringl, H., Schernthaner, R. E., Schueller, G., Balassy, C., Kienzl, D., Botosaneanu, A., Weber, M., Czerny, C., Hajdu, S., Mang, T., et al. (2010). The skull unfolded: a cranial ct visualization algorithm for fast and easy detection of skull fractures. *Radiology*, 255(2):553–562.

[6] Wang, G., Li, W., Ourselin, S., and Vercauteren, T. (2017). Automatic brain tumor segmentation using cascaded anisotropic convolutional neural networks. *arXiv preprint arXiv:1709.00382*.