MSDS 458
A4
Jason Adam
August 29, 2019

Project Problem Statement

I consider this project to be more of a personal challenge than a problem statement. I chose to work on a dataset of Computed Tomography (CT) scans of the human brain in an attempt to build a classification model as to whether the scan contained an intracranial hemorrhage or not. The main challenge is to build a Convolutional Neural Network (CNN) that can be trained on 3D arrays since CT scans are made up of a series of slices.

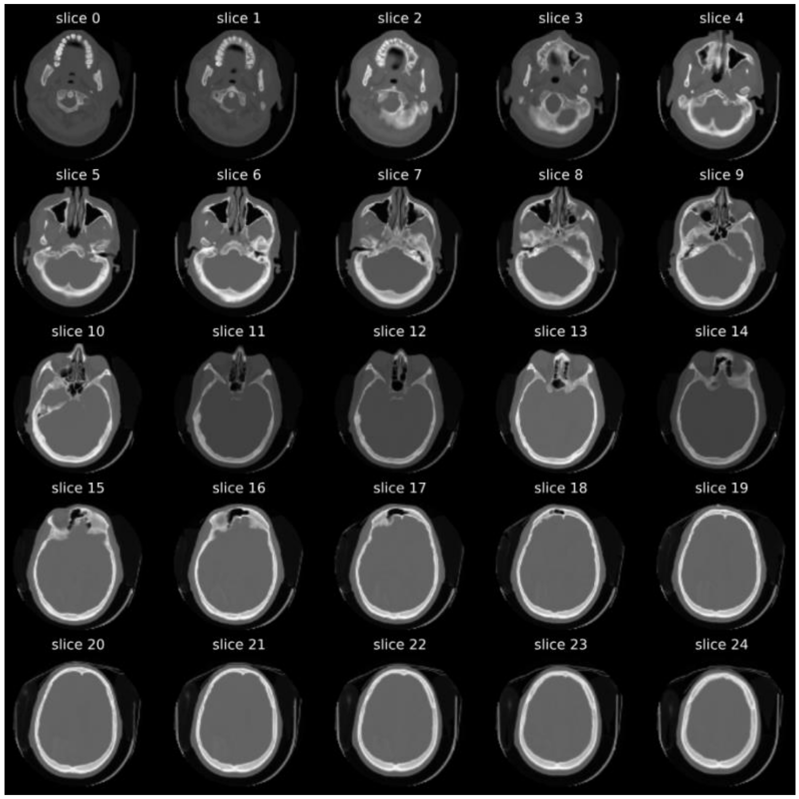
Data Set

The final dataset for this project totaled 120 sets of scans (i.e. 120 patients). Each patient's scan consisted of 32 scans. This was the most common dimension, so I decided to use it to avoid overly complicated pre-processing. As a brief reminder, the image slices are in the DICOM format (Digital Imaging and Communications in Medicine). This format contains a lot of metadata about the patient, slice, etc. as well as the numerical array of the image itself. Each scan came with an annotation of its problems as determined by a panel of 3 radiologists (Chilamkurthy, et al., 2018).

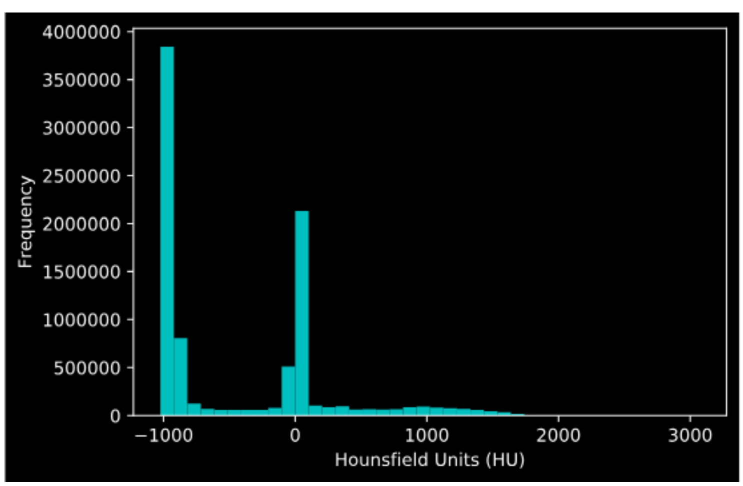
I decided to use the category of intracranial hemorrhage (ICH) as it contained roughly a 50-50 split in the dataset overall. After selecting all patients with 32 slice scans, the split was roughly 69% with the ICH and 31% without (table below):

ICH Flag	Patient Count
0	37
1	83

This set the baseline for my experimentation at an accuracy of 69%. The scans could be visualized as follows (sample patient):



The array values of the scan were capture in Hounsfield units. The Hounsfield scale is a standard numerical scale that applies to medical-grade CT scans. Different values represent different tissue types.



A quick recap of the histogram reveals that large majority of the scan is air (-1000), and the remaining spike is white and grey matter (just above 0). The final spike around 1000 represents the skull (Wikipedia, 2019).

Problems that Arose

This dataset proved incredibly difficult. Admittedly, I might have been overly ambitious with this problem. There are plenty of image datasets on Kaggle with lots of code already written for building the model, visualizing the features, etc., but I wanted to challenge myself from a pre-processing perspective and from an architecture perspective as there is scant resources out there for a problem of this type. My main pain points came from two distinct areas. The first was the pre-processing component. The file structure was sloppy at best and there was no common naming convention which lead to quite a few dynamic functions. Additionally, the files were in a format (DICOM) that did not fit into pre-baked TensorFlow and Keras pre-processing functions. The second, and admittedly the most frustrating, was the lack of compute power on my local MacBook. I tried several iterations of 3D CNN architecture, but my computer struggled to perform the computations.

Interpretation

I don't feel as if I've gotten as good a grasp on the interpretation as I would like. I struggled to find ways to interpret the outputs of 3-dimension conv nets. Below is a table of my experimentation configuration, and I will follow it up with discussion:

Architecture	Array Size	Training Size	Accuracy	Loss
- Conv3D - MaxPooling3D - Flatten - Dense - Dense - Dense (output)	8x128x128	50	69%	2.73
- Conv3D - MaxPooling3D - Conv3D - MaxPooling3D - Flatten - Dense - Dense - Dense (output)	32x256x256	120	69%	4.91

For each model, I used a sigmoid activation function on the final output layer since I was performing a binary classification. I initially started with fewer samples that were scaled down further from the original. I was unable to achieve greater than the baseline percentage in the data, so I processed a second batch of samples that had all the slices but were scaled to 50% of the pixel size. I first ran this on the same architecture as the first model but got the same results. I ran this second dataset on the model with an additional 3D convolutional layer, but it still returned the same values. Additionally, these 3D neural networks take a really long time to train on a CPU, so it was incredibly hard to iterate.

It feels like the model is having a difficult time picking up on nuanced features in the arrays as the location of an ICH is very small amongst the 3-dimensional representation of the brain. When I thought back to the recognition of the letters, they had fairly distinct characteristics as well as common characteristics. This allowed the model to build up a representation of the letters effectively. The brain CT scans would require a lot more epochs to pick up on the nuanced features in the data. Each scan is very similar and shares a majority of features with every other scan. Also, the original study used a much higher number of scans/slices (numbering greater than 10k patients) whereas I was attempting to isolate very nuanced features on only 120 samples. Overall, I feel like the model would learn better with more samples and perhaps a longer set of epochs.

Conclusion

Overall, I enjoyed this project despite its difficulties. I would have liked to get to iterate more on the models themselves, but I felt somewhat limited by my computing power. I will admit that I considered abandoning this project in favor of something “easier”, but I decided to stick with it because I feel like medical image classification is such an important field. I get a sense of excitement working on problems like this with real world implications. I can’t say I feel the same way about classifying cats vs dogs. I believe that as neural networks become easier (using that term loosely here) to interpret, they will be embedded into medical systems to augment the skills of a radiologist. I don’t foresee them ever replacing medical professionals, but rather a silent partner that could potentially give really good advice and direction.

Future State

I’m planning to pick this project back up in the future. I want to look at different architectures in the literature as a beginning roadmap. Additionally, I want to provision some cloud resources with GPU so that I can iterate through models a little more quickly. There were some really awesome visualizations of feature maps in the Deep Learning book by Chollet, but they were based on 2D images. I’d like to attempt to overlay a heatmap on the 3D graphic of a brain scan to see where the model is picking up an ICH classification. This is all assuming I can get the model to perform better than baseline. I think I will also attempt a data augmentation step which will increase the number of samples I can train on. Overall, I've really enjoyed this project even though it had me pulling my hair out at times and questioning by bull-headed desire to tackle a difficult project.

References

Chilamkurthy, S., Ghosh, R., Tanamala, S., Biviji, M., Campeau, N. G., Venugopal, V. K., ... & Warier, P. (2018). Deep learning algorithms for detection of critical findings in head CT scans: a retrospective study. *The Lancet*, *392*(10162), 2388-2396.

Wikipedia contributors. (2019, August 9). Hounsfield scale. In Wikipedia, The Free Encyclopedia. Retrieved 19:31, August 18, 2019, from https://en.wikipedia.org/wiki/Hounsfield_scale