Capstone Project Proposal

UCSD Machine Learning Engineering Bootcamp, October 2021 Cohort

Keenan Hom

# Identified Problem

Internal hemorrhaging within the brain is a major cause of strokes in the US, which is among the top causes of death nationwide. When a patient shows signs indicating that they might have a stroke in the future, radiologists must scrutinize intracranial scans to detect both the presence and location of a potential hemorrhage. As common sense would have it, the process is time-consuming, which adds further risk to the patient, and very prone to human error. Many preliminary studies show that using a machine learning algorithm to review scans for the presence of cancer, bleeding, and other malicious presences not only performs diagnoses significantly faster than humans, but also at a higher accuracy.

My chosen dataset contains many scans that may show one of four subtypes of intracranial hemorrhaging, or none at all. Sage & Badura (2020) produced a model trained on CT scans that varied from 75% to 98% depending on the subtype. Lee et. Al produced a novel type of learning algorithm that doesn’t use backpropagation for training, reaching sensitivity/specificity rates of around 82% on average, and as high as 91 percent for one subtype.

Such an application of machine learning that produces more accurate results in a shorter time than humans could lead to early action taken to prevent strokes, saving many lives, as well as provide insights on how radiologists can improve their own diagnosis accuracy.

# Data

My dataset comes in the form of DICOM images from a [Kaggle competition](https://www.kaggle.com/c/rsna-intracranial-hemorrhage-detection/overview/description) organized by the Radiological Society of North America (RSNA). The training dataset contains about 753 thousand images, and the test dataset contains about 121 thousand. Each DICOM image contains a 512x512 pixel array of integers, which may range from -1024 with no upper bound, so each image should likely be normalized by its highest magnitude pixel. Each DICOM image is labeled by the type of hemorrhage present (one of five subtypes), if any. At first glance, the data appears to be unbalanced in favor of data with no hemorrhaging.

# Tentative Approach

This is a supervised classification problem, with six categories (an image may be one of five subtypes or have no hemorrhage). I certainly need to use a combination of deep learning techniques in order to produce a satisfactory model. As a baseline, I would like to aim for 75% sensitivity/specificity on all six categories. I will focus most of my time on model development, instead of deliverability and presentability.

## Tentative model Design

The main predictor of my model will most likely be a Convolutional Neural Network (CNN) with several compressed artificial neural network layers at the end, an architecture best suited for image classification. Some heavy feature extraction may be necessary before training the CNN, for which I may use unsupervised methods such as Autoencoders to reduce dimensionality, and other techniques to detect blobs and ridges in the scans (such patterns are often a sign of hemorrhaging). I might also find improved results with the use of random forests.

# Deliverability

I will not focus that much on my deployment method, so I will likely store my model in a library or web service API instead of an app with a UI. The focus will be on results rather than packaging.

# Required resources

I do not have a good idea of the extent of my model architecture yet, however I’m certain that training a CNN on 753 thousand images will have a heavy computational cost. I would like to have around 256 to 512 GB of memory which would allow me to store the entire dataset, however I can otherwise split the dataset and only store certain sections at a time. However, I will still need a large amount of memory to store and update the CNN’s parameters. Being image data, I won’t be able to use Pandas or PySpark for vectorized operations. My CPU processing power should emphasize multicore processes, and GPU should also be capable of handling vast parallel workloads (NVIDIA workstation or server GPUs would be best). I’m thinking of using a library like TensorFlow that can accomplish this.

Works Cited

Lee, Ji Young, et al. “Detection and Classification of Intracranial Haemorrhage on CT Images Using a Novel Deep-Learning Algorithm.” *Scientific Reports*, vol. 10, no. 1, 2020, https://doi.org/10.1038/s41598-020-77441-z.

Sage, Agata, and Pawel Badura. “Intracranial Hemorrhage Detection in Head CT Using Double-Branch Convolutional Neural Network, Support Vector Machine, and Random Forest.” *Applied Sciences*, vol. 10, no. 21, 2020, p. 7577., https://doi.org/10.3390/app10217577.