Matrix Methods for Data Analysis and Machine Learning Final Project
Zeta Surgical

Joshua Bigman Jonathan Goldstein Sean O'Hara Conner Lusk Jesse Segel
David Zhuang

December 11, 2021

Abstract

Intracranial hemorrhaging (ICH), or acute bleeding inside of the skull or brain, is a medical emergency which must be scanned for and acted upon quickly, as they can lead to seizures, strokes, or even death. ICH's are found using computed tomography (CT) scans, which provide a number of cross-sectional images of the brain. These images can be analyzed by radiologists to locate an ICH and identify the specific type of hemorrhage. However, because of the need for both high efficiency and high accuracy when locating and identifying an ICH, we hypothesize that machine-learning algorithms could perform at a level equal to or greater than that of a human radiologist. The best results garnered were a result of transfer learning, with a training accuracy of 0.923 and a validation accuracy of 0.832.

1 Introduction

Zeta Surgical is a biotechnology company developing computer vision and artificial intelligence tools to improve surgery [1]. The company's first product will guide neurosurgeons during the treatment of neurological emergencies. To train a device capable of navigating the brain during surgery, Zeta has compiled 150,000 images of brain hemorrhages. The goal of this project is to train and compare successively more complex models to identify the type of brain hemorrhage shown in an image. The types of hemorrhage are limited to five different possibilities: Intraparenchymal, Intraventricular, Subarachnoid, Subdural, and Epidural.

Specifics in the differences of these types of hemorrhage are noticeable both visually and through deep learning. Intraparenchymal hemorrhages are generally a result of high blood pressure, trauma, or a tumor and are located within the brain. They are spawned arterial or venous. Intraparenchymal hemorrhages are typically rounded and present with acute headaches, and nausea.

Figure 1: Intraparenchymal Hemorrhage

Intraventricular hemorrhages can be associated with intraparenchymal or subarachnoid hemorrhages and are located within the ventricle. They can be both arterial or venous and the generally conform to the shape of the ventricle. This type of hemorrhage presents with similar symptoms as an intraparenchymal hemorrhage, acute headaches, and nausea.

Figure 2: Intraventricular Hemorrhage

Subarachnoid hemorrhages are located between the arachnoid and the pia mater. They are most usually the result of a ruptured aneurysm, malformation, or trauma. These are predominantly arterial and appear as fissures along the sulci. They present with an acute and severe/worsening headache.

Figure 3: Subarachnoid Hemorrhage

Subdural hemorrhages are located between the dura and the arachnoid and are almost always the result of trauma. These spawn through bridging veins in a crescent shape. They present with worsening headaches.

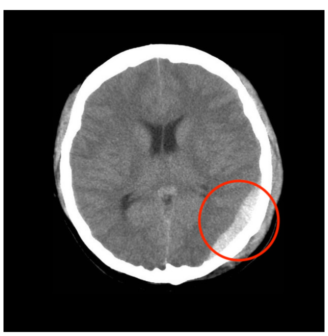

Figure 4: Subdural Hemorrhage

Lastly, Epidural hemorrhages are located between the dura and the skull. They are generally brought about due to trauma after some form of cranial surgery. They spawn arterially, are lentiform in shape, and present as acute skull fractures and altered mental states.

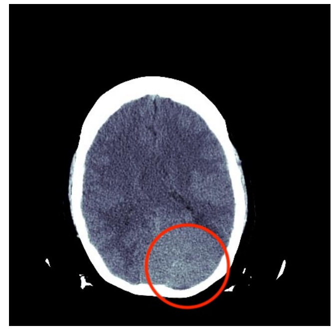

Figure 5: Epidural Hemorrhage

The dataset consists of over 100,000 512x512 computed tomography (CT) scan images provided by Zeta Surgical of patients with or without an intracranial hemorrhage. The CT scans can be rendered using varying windowing techniques, allowing for multiple images for each patient. There are four image types in the dataset: brain-bone window, brain window, subdural window, and max contrast window. Using different deep learning methods we attempt to create models that can predict the location, size, and type of hemorrhage within a specific brain scan window.

2 Related work

RADnet Analyzed CT scans of the brain in order to detect brain hemorrhages, and was trained using 185 scans for training, 67 for validation, and 77 for testing [2]. It utilizes a 40-layer DenseNet architecture with three additional auxiliary tasks which allow it to focus primarily on the hemorrhagic region, improving both efficiency and performance. RADnet's accuracy and precision is comparable to that of a senior radiologist.

3 Data Cleaning

The data given for the masks for image segmentation were strings with coordinates of the outlined segmentation. If there were multiple hemorrhages, the image was split into multiple numpy arrays, before starting to fill the hemorrhages. Our algorithm for filling the images, then outputed a (512,512) image. Since the segmentation model was based off of densenet, before training the data was resized to (512,512,3). When transfer learning, the data was reshaped into (512,512,1). All of the data then was split by the sklearn split test train library into 80% for training and 20% for testing/validation.

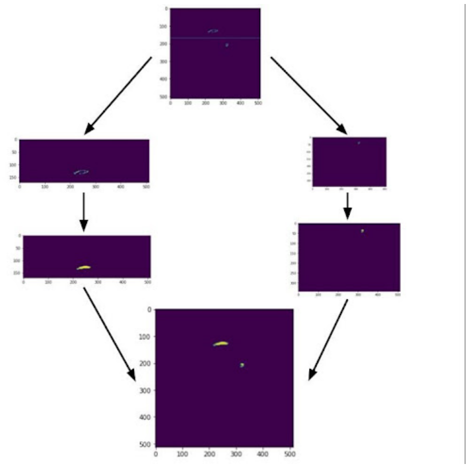

Figure 6: Data Masking

4 Learning Method Description

One of the most significant modeling challenges within our research has been the sheer size of the data. With more than 100,000 512x512 images and a total storage size exceeding 9GB, it is infeasible to simply load the data into memory.

One approach we adopted to get around this issue was randomly sampling images from the dataset to include in training and testing, stratifying by the hemorrhage class. This approach allowed for a massive reduction in data size

Another problem with the data is that it consists of several different scan types, and each scan type is likely to be most effective in diagnosing a certain subset of hemorrhages. For example, the subdural window CT images very clearly show the patient's dura region, but contain much less detail in the patient's brain area; these images are likely very effective in diagnosing ICHs along the dura (epidural and subdural) but much less so for the remaining classifications.

4.1 Parameter Optimization

Through testing, the model's hyperparameters were attempted to be optimized to decrease overfitting and training time. The two most important hyperparameters that were optimized were learning rate and momentum since the learning rate is regarded as the most important hyperparameter to tune and the momentum is dependent on the learning rate [3]. These two parameters were increased as much as possible before causing instabilities while training.

Momentum helped increase the speed of our model since it accelerates SGD in the relevant direction while dampening the isolation [4]. This helps find the local/global minimum at a much quicker rate since it decreases the movement in the wrong direction. Our learning rate was adaptive in order to attempt to curve overfitting. The learning rate and number of epochs were negatively correlated which helped curve the effect of overfitting through our observations. The final method that was used to combat overfitting was implementing dropout layers after each dense layer. Dropout layers temporarily remove nodes from the network, along with all its incoming and outgoing connections. This decreases the number of features and provides a way of approximately combining exponentially many different neural network architectures efficiently [5].

4.2 SciKit Learn Logistic Methodology

The sci-kit learn library was used to perform a logistic regression. The data was split into two equal parts for testing and training. Additionally, the images were downsampled to allow for quicker iterations.

4.3 NN and CNN Methodology

The neural network contained two dense layers, 700 nodes with a dropout rate of 0.3 and 300 nodes with a dropout rate of 0.1, along with a softmax layer for prediction. In total this model contained 183,713,907 total parameters, all of which being trainable.

The convolutional neural network contained two convolution layers, two max pooling layers, followed by a dropout layer of 0.3, a dense layer with 40 nodes, and a softmax layer for prediction. The total parameters of this model was 20,011,175 and was trained with Root Mean Squared Propagation.

4.4 Image Segmentation Methodology

The image segmentation model was built using the UNET framework which works by using an encoder-decoder scheme. This method reduces spatial dimensions before increasing them back up and concatenating the corresponding encoding and decoding layer to produce the segmentation [6]. The input to this model was a (512,512,3) image and output was a (512,512,1) image. Since our model was only trained on correct and majority labels, it was only trained on 16,000 images.

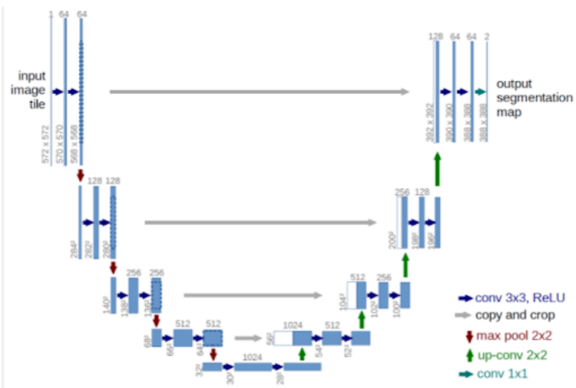

Figure 7: U-Net Architecture [6]

4.5 Transfer Learning Methodology

Another approach that was tested was using a pre-trained convolutional neural network to diagnose the presence of intracranial hemorrhages. We chose to use the architecture and weights from the ImageNet model, which contains 5 convolutional layers and upwards of 18 million parameters. The final layer of the model was re-trained to fit this specific application, i.e. predicting one of seven possible hemorrhage classes.

One challenge of the transfer learning approach is that the pre-built model expects a single 3-channel RGB image as an input. This prevented use of all four scan types simultaneously, and necessitated some arbitrary choice as to the type of scan to use within the model (we chose brain window scans).

One notable challenge of the transfer learning approach was the time required to train the model. A training sample of 79,670 images over 7 epochs took over 22 hours to train! The computational complexity of training made it difficult to optimize hyperparameters and limited the number of epochs that could be run.

5 Results

5.1 SciKit Learn Logistic Regression Accuracy

As an initial test, 40 iterations were used which yielded an accuracy of 0.369. A confusion matrix was generated and can be seen below.

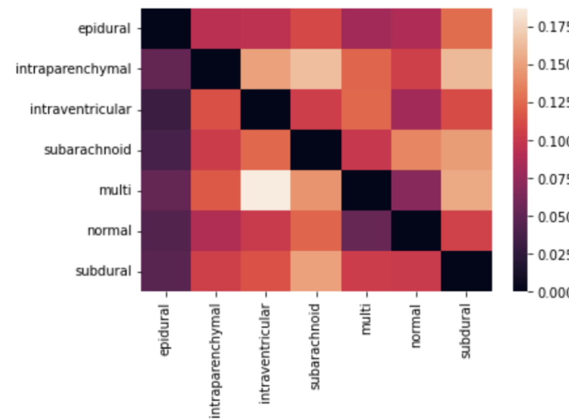

Figure 8: Logistic Regression Confusion Matrix

5.2 Trained Neural Network Accuracy

The tensorflow library, along with the keras API were used for the creation and training of our neural network. Our model was trained using stochastic gradient descent and through 150 epochs with an accuracy of 0.754 and a validation accuracy of 0.563. A model accuracy diagram can be seen below.

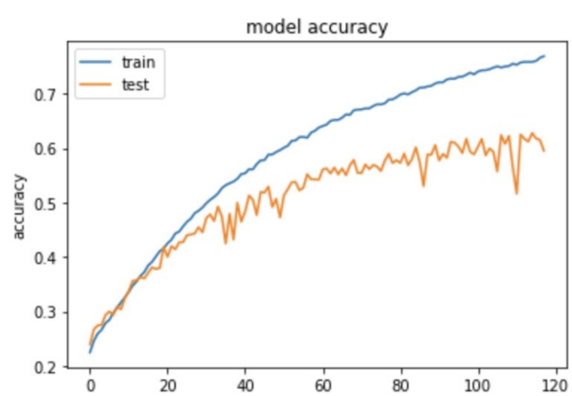

Figure 9: NN Train/Test Accuracy

5.3 Trained Convolutional Neural Network Accuracy

The tensorflow library, along with the keras API were used for the creation and training of our convolutional neural network. This model was trained for 8 epochs and finished with an accuracy of 0.813 and a validation accuracy of 0.624. A model accuracy diagram can be seen below.

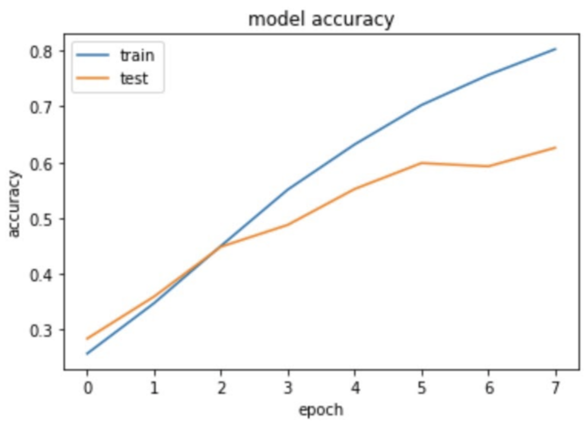

Figure 10: CNN Train/Test Accuracy

5.4 Image Segmentation Accuracy

This model saw the worst result out of any model barring the Logisitic Regression, which can be attributed to the least amount of images and little fine tuning. One improvement to our model besides fine tuning could be using data augmentation to increase the testing data. Our final result was obtained after 70 epochs and is attributed to 55% accuracy using intersection of union.

5.5 Transfer Learning Accuracy

Despite the extensive time to train, the transfer learning model was the best-performing classifier we tested, predicting the hemorrhage type correctly 92.2% of the time in training and 79.3% of the time in testing. The superior performance of the model likely stems from the robustness of the Densenet model, which was trained on millions of images; the large training sample also likely plays a role.

Although the transfer learning model performed exceptionally well, the 13% gap between training and testing error could indicate that some overfitting is present. This could signal that the model was perhaps trained for too long, or it could indicate that the model hyperparameters or architecture may need to be adjusted. Another interesting tidbit from the confusion matrix (see below) is that the most common error of the model was classifying subdural hemorrhages as epidural. Considering that both of these hemorrhage types occur around the dura, this sort of error speaks to the challenge of delineating ICHs with similar appearance.

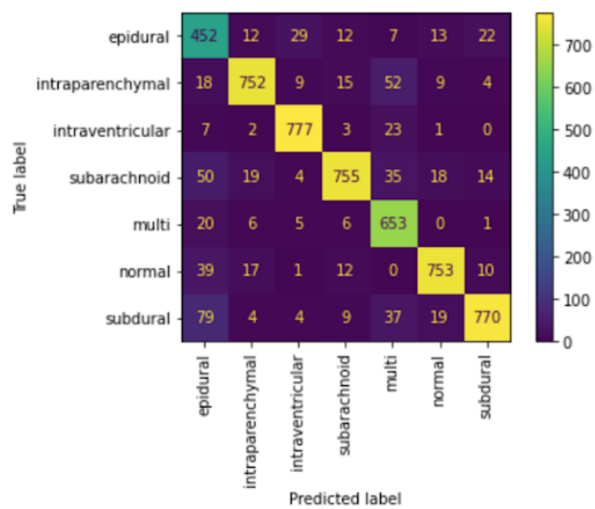

Figure 11: Transfer Learning Confusion Matrix

6 Conclusions

Using different deep learning methods we created models that can predict the location, size, and type of hemorrhage within a specific brain scan window. The accuracy of these models varied greatly given the type of learning. The worst of which presented is the logistic regression model which had an accuracy of 0.369 (36.9%) over 40 epochs. This was done using the sci-kit learning library.

The next two methods of a Neural Network, and a Convolutional Neural Network both yielded semi-accurate results. A basic Neural Network had an accuracy of 0.754 (75.4%) and a validation accuracy of 0.563 (56.3%). The Convolutional Neural Network had an accuracy of 0.813 (81.3%) and a validation accuracy of 0.624 (62.4%). The large discrepancies between accuracy and validation accuracy is a result of overfitting.

This problem is apparent with all of the complex models and will need further development to accurately predict hemorrhage type and location.

Image Segmentation had an accuracy of 0.550 (55%) which is lower than both types of Neural Networks attempted. This is a result of limited amount of images, that could be processed by the system. With fine tuning and data augmentation we believe this method could have a large increase in accuracy.

Lastly, Transfer Learning yielded the best results out of every method attempted. Despite the lengthy run time of 22 hours, the Densenet model garnered significant results. With an accuracy of 0.922 (92.2%) and a validation accuracy of 0.793 (79.3%), the Transfer Learning method surpassed our expectations. There is still apparent overfitting of this model, with a gap between testing and validation accuracy. In future learning models, dropouts that limit the amount of overfitting, might be beneficial.

7 Further Research

We were able to experiment with a variety of methods in our research, ranging from simple logistic regression to more advanced image segmentation. However, there are some additional models that could perform well in this context. Further models we would have liked to test include k-nearest neighbors, support vector machines, and tree methods (e.g. random forest, AdaBoost). Additionally, our current models could have benefited from further experimentation around optimizing hyperparameters and training methods. In the case of logistic regression, we would have liked to investigate the impact of switching to a mini-batch training method. Our neural network-based models (NN, CNN, transfer learning, image segmentation) could also have benefited from running more epochs, better optimizing hyperparameters, or modifying the underlying architecture.

References

[1] J. Amich, and R. Sha. Brain Imagery Data-set. 2021. 1

[2] D. Yeager. AI Speeds Brain Hemorrhage Detection In RT, 2017. 2

[3] L. Smith Neural Network Hyper-Parameters In US Naval Research Laboratory, 2018. 3

[4] S. Ruder Gradient Descent Optimization Algorithms 2016. 3

[5] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever and R. Salakhutdinov Prevent Neural Networks from Overfitting In JMLR, 2014. 3

[6] O. Ronneberger, P. Fischer and T. Brox Convolutional Networks for Biomedical Image Segmentation 2015. 4