Brain Hemorrhage Image Segmentation and Classification

Ido Tzhori Luc Ravenelle Jason Pereira Isaya Acevedo
Department of Mathematics - Northeastern University

April 24, 2023

Abstract

Recently, The use of machine learning and computer vision algorithms has increased in the medical imaging industry, particularly in identifying and segmenting hemorrhages in brain CT scan images. This paper explores the application of classification and image segmentation methods for this purpose. We evaluate popular classification methods and implement the U-net architecture for segmentation. Our results indicate that the current data and methods do not meet regulatory accuracy standards for large-scale implementation, but the U-net architecture shows promise with room for further improvement.

1 Introduction

Our project aims to leverage the power of machine learning to classify and segment brain hemorrhages using CT scan images. The 5 classes of hemorrhages are intraparenchymal, subarachnoid, subdural, and epidural and multiple hemorrhages.

By finding a model that can accurately predict distinct types of brain hemorrhages, medical personnel can use this tool to assist in their hospital decisions. Since brain hemorrhages can be life threatening, faster, more accurate classification will save lives. To ensure the highest performing model, we will evaluate and optimize various metrics commonly used in medical imaging - recall, precision, accuracy, etc.

We plan to use the data provided by Zeta Surgical to label which type of hemorrhage is seen in each image. The images are split into different folders which will be wrangled and cleaned to get a flat file suited for a machine learning algorithm.

2 Data Cleaning

Clean data is arguably the most critical pieces of modeling and machine learning. It is important that machine learning models are not given an additional bias or inconsistency that could further impact their efficacy. Data is the only real truth, and it needs to maintain this legitimacy in order to make good decisions in industry and in research. This is why it is so important to spend time cleaning and ensuring data is correct and valid.

2.1 Exploration

To clean and process our data, we began with exploration. The two main pieces of data that concern our goal are brain images and their labels. Brain images are broken down into seven different categories. Five of which are different types of brain hemorrhages, one class is for multiple hemorrhages and another one gives images of normal brains without complications. The hemorrhage types given are epidural, intraparenchymal, intraventricular, subarachnoid, subdural, as well as one with multiple hemorrhages in the same image. Within each one of these categories are four different image types given from the CT scan. These different categories show a different perspective, and each gives its own unique information about the brain it is capturing. We will use these different image types to explore performance and what works best. With the exception of intraventricular, we were also given a label file for each hemorrhage type. Because the intraventricular labels were not supplied, we will have to ignore this hemorrhage type in our analysis. These files list the image id and coordinate for the location of the hemorrhage in the image. These csv files also serve as a source of truth to indicate which images correspond to each type of hemorrhage and we will use fact this later in our processing.

2.2 Processing

To perform analysis on images we needed to process them in a way that a machine can interpret and extract the important information. One way to do this is to use python libraries to flatten and convert the images into matrices of pixel values. The majority of images supplied were jpg format with size 512 x 512. We used the matplotlib image library to read these images and compose a list of all images with a size of 262,144 (512 x 512). We were not able to use all images, as not all images are labeled. We had to filter hundreds of thousands of images down to just several thousand because those are the only ones in which we had a source of truth for. We filtered out all images that did not have a ‘Labeled’ or ‘Gold Standard’ state in the label files. All other images either were not labeled or had not yet been completed. This left us with far less data, but data that is consistent and truthful. Using anything else could leave us with unintended sources of bias or images that were labeled incorrectly. Intuitively, for our normal images there are no labels. We did not need to filter these normal images except for the fact that there are far too many than necessary. Using all 10,000 images of normal brains would have an adverse effect on our model because the normal images would far outweigh the hemorrhage images. To avoid biases, we chose to randomly sample a similar amount of normal brain images as we have for our hemorrhaged brain images. Because our hemorrhage brain images range from as low as 300, to as high as 650, we chose to randomly sample 500 normal brain images to supply to our models.

To further clean this data we needed to also filter out some inconsistencies. There were some images that were of a different size. Somewhere along the line of data aggregation these images we given a different size and therefore will not work well with our model. If we needed to we could possible resize these images to work, but because there were so few we decided to not include them in the final training data. Modifying them could also change some of the information in the images and thus have adverse effects on our classification efforts. At the end of our cleaning effort we had roughly 1800 images flattened that we could then split into training and tests sets for our analysis.

2.3 Specific Model Modifications

At this point there is enough data cleaned to perform preliminary classification models like logistic regression and LDA, among others. Each model uses flattened images taking the form of 1835 x 262,144. Because we have access to large compute and memory using Northeastern’s discovery cluster, we chose not to lose information by downsizing the images.

For our Convolutional Neural Net classification efforts, we chose to save memory by using Tensorflow’s Dataset class and functions. Rather that storing our information in numpy arrays we used the Tensorflow class to save each image as a Tensor. This allows us to not have to allocate memory beforehand to the pixel data and we can load them at runtime just with a file name and directory path. This allowed us to be able to run up to 50 epochs on a Macbook without the need for the discovery cluster. This may have not been possible without the tensorflow library.

3 Classification

3.1 Logistic Regression

When building classification models, it is best to start with the simplest method. In our case, we began with multinomial logistic regression, which is a natural extension of binary logistic regression to multi-class classification problems. We used the Sklearn library’s LogisticRegression class with the ’lbfgs’ optimization algorithm to classify six types of hemorrhages - epidural, intraparenchymal, subarachnoid, subdural, multiple hemorrhages, and normal brains.

We built two models; one with default parameters for the class, and another after using GridSearchCV in an attempt to optimize our hyperparameters. GridSearchCV is a library that will produce different models with different combinations of hyperparameters that you supply it. Although this is a very compute intensive process, we leveraged to discovery cluster’s GPUs. At the end of run time, the search function will give you the ’best’ model based on accuracy and the hyperparameters that it used to build that model. This process improved our results by a few percentiles, but the interesting part comes when you examine a heat map of the model’s predictions.

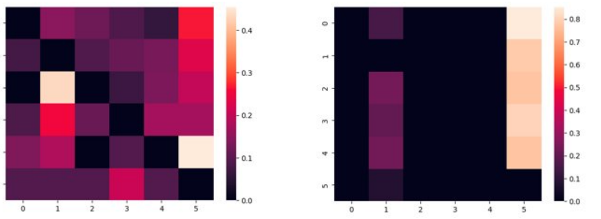


Figure 1: A heatmap of predictions before and after ’optimization’

In Figure 1, the image on the left shows the predictions and misclassifications of the pre-optimized baseline model. We can see that class two is being misclassified as class one, and four is being misclassified as five. In the image on the right we have the same thing but for the model in which GridSearchCV gave as the most accurate. We can see that the only classes being predicted are class one and class six! The library is clearly being tricked by the fact that the accuracy given by this model is higher, but it is only higher because all of the instances of one and six are being predicted correctly. This example goes to show that we cannot always take the answers given to us by tools like GridSearchCV for granted and we should always examine the predicted results.

3.2 Random Forest

Random Forest is an ensemble learning method that creates multiple decision trees during training and combines their outputs for final predictions. This approach reduces overfitting and increases the model's generalization capabilities. Random Forest can handle large datasets and high-dimensional feature spaces, making it suitable for our brain hemorrhage classification task. Our random forest model was tuned using GridSearchCV.

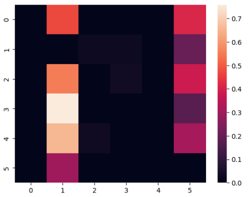


Figure 2: Random Forest parameter tuned heatmap

3.3 XGBoost

XGBoost, short for eXtreme Gradient Boosting, is another ensemble learning method that utilizes gradient boosted decision trees. The algorithm is designed to be highly efficient, scalable, and flexible. It incorporates regularization techniques to reduce overfitting and improve model performance. In our case, XGBoost can potentially outperform other methods due to its ability to handle a large number of features in the image data. We tuned the parameters using hyperopt instead of GridSearchCV.

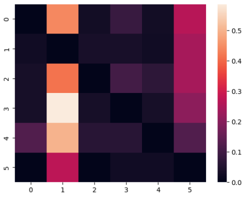


Figure 3: XGBoost parameter tuned heatmap

3.4 Support Vector Machines

Support Vector Machines (SVM) is a powerful algorithm for classification tasks that aims to find the optimal separating hyperplane between classes. It can efficiently handle high-dimensional data and can be extended to multi-class classification using techniques such as one-vs-one or one-vs-rest. SVM can be advantageous in our brain hemorrhage classification task as it can handle the complexity of the data, and the kernel trick can be used to capture non-linear relationships between features. SVM was tuned using GridSearchCV.

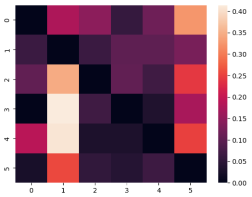


Figure 4: SVM parameter tuned heatmap

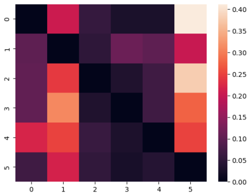


Figure 5: KNN parameter tuned heatmap

3.5 Tuned Model Results

All of our tuned models printed similar confusion matrices. Accuracy only increased between 2-3% by tuning while original models with default parameters were better able to classify most of the different hemorrhage types just at lower accuracies around 10%.

The confusion matrix for all the tuned models shared similar results. The models didn't perform well in classifying epidural hemorrhages. They had a small number of true positives but also a bigger amount of false positives where epidurals were misclassified as intraparenchymal. Also, they had a high amount of false positives where other types were classified as epidural.

The models performed best on intraparenchymal hemorrhages, having true positives ranging from 50-60 depending on the model used. However, they still have a significant number of false negatives and false positives. The models have a hard correctly classifying subarachnoid, subdural, and multi hemorrhages with all of them having 0 to 3 true positives. The models perform relatively better on normal images with a high number of true positives. However, it still has a significant number of false negatives and false positives.

3.6 Convolutional Neural Network

Lastly, for our CNN models, we thought it would be best to implement a pre-trained CNN model that uses the ImageNet database. That way the model already knows important features from the data as well as improves performance and reduces training time. In order to get the best results, we thought it would be best to transform our cleaned data into a TensorFlow data set consisting of 256x256 images each with 3 color channels. First implementing VGG-16, a 16-layer CNN model that has 13 convolution layers and 5 pooling layers, we were able to achieve an accuracy and F1 score of 33.23% and 20%. The poor results indicate that the model struggled with predicting images other than normal as well as poor performance. After seeing similar results to our other classification models we moved on to an Xception model which has 36 layers. This model gave us our best accuracy results with 61.83% and a F1 score of 22%. The best way to improve these models would be to add more data to our training data by performing data augmentation as well as finding a pre-trained model that deals specifically with medical images.

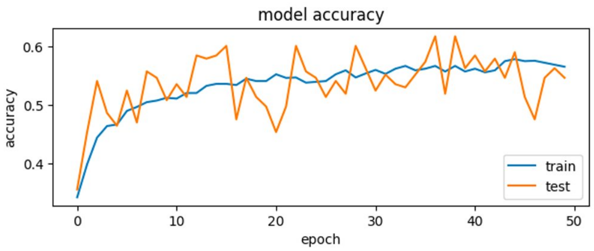


Figure 6: Xception model Accuracy

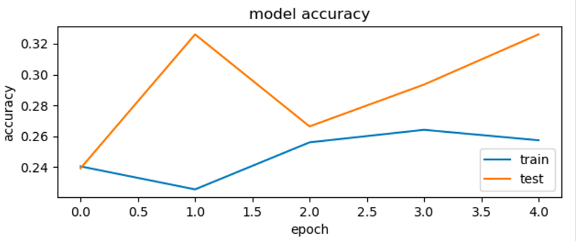


Figure 7: VGG-16 model Accuracy

3.7 Model Statistics

Model	**Accuracy**	**F1**
Logistic	31.5%	31.50%
LDA	33.7%	35.00%
Random Forest	42.5%	42.50%
XGBoost	41.8%	41.80%
SVM	38.26%	38.26%
KNN	37.54%	37.54%
Xception	61.83%	22.00%
VGG-16	33.23%	20.00%

3.8 Conclusion

While hyperparameter tuning is a valuable step in the model development process, it is not always guaranteed to yield significant improvements in performance. It is crucial to consider other aspects of the modeling process, such as data quality, algorithm choice, and feature engineering, to achieve the best results for our brain hemorrhage classification.

4 Hemorrhage Segmentation

In this section, we describe the steps taken to segment the hemorrhage region.

4.1 Making Labels

The first step in preparing the data is to create labels for the images. In our project, we used a binary segmentation approach to identify pixels that correspond to hemorrhages. We received manually annotated x, y coordinates of the hemorrhaging region. We used Open-CV to create the mask where 0 represented the background and 1 represented hemorrhage for each pixel. This resulted in a data set of 854 images, each with a corresponding binary mask that identified the hemorrhaging region.

4.2 Cleaning the Data

Next, we cleaned the data by removing any images with poor or overlapping masks. We also removed any images that didn't meet a minimum or maximum percentage of the total image. This was determined by subsequent model performance on hemorrhage areas that were too small.

4.3 Reshaping

The input images had a shape of $512 \times 512 \times 3$, where each layer corresponds to the pixel intensity of each color in RGB format. The target data had a shape of $512 \times 512 \times 1$, where each pixel was either a 0 or 1, corresponding to background or hemorrhage, respectively.

8 shows two examples of brain CT scans with their corresponding binary masks.

4.4 Splitting

To evaluate the performance of our model, we split the data set into training, validation, and test sets. We used a 90/10 split for the training and test sets.

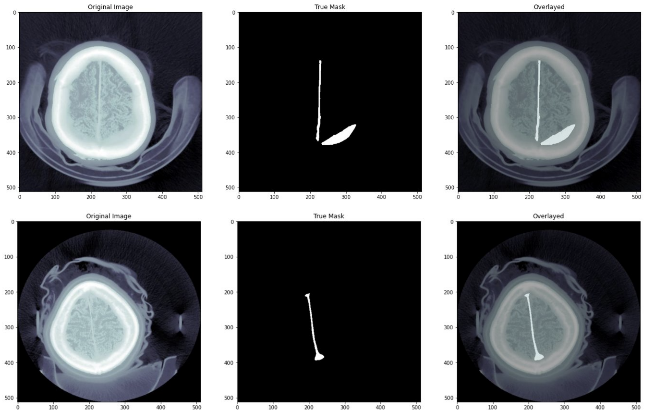


Figure 8: Examples of brain CT scans and their corresponding binary masks. The original image represents the input and the true mask represents the desired output.

4.5 Evaluation Metrics

To evaluate the performance of our hemorrhage segmentation model, we used the following metrics: pixel accuracy, Intersection over Union (IoU), and F1 score.

Pixel Accuracy (PA) measures the percentage of correctly classified pixels. It is calculated as:

$$PA = \frac{TP + TN}{TP + TN + FP + FN}$$

where TP, TN, FP, and FN are true positives, true negatives, false positives, and false negatives, respectively.

Intersection over Union (IoU), also known as Jaccard Index, measures the overlap between the predicted and ground truth masks. It is calculated as:

$$IoU = \frac{TP}{TP + FP + FN}$$

F1 score is the harmonic mean of precision and recall. It is calculated as:

$$F1 = 2 \cdot \frac{precision \cdot recall}{precision + recall}$$

where precision is defined as:

$$precision = \frac{TP}{TP + FP}$$

and recall is defined as:

$$recall = \frac{TP}{TP + FN}$$

Each evaluation metric provides a different perspective on the performance of our model. Pixel Accuracy measures overall classification accuracy. Keep in mind that pixel accuracy alone is not the best metric, as the relevant areas (e.g. brain tissue, hemorrhage) is often much smaller than the background (e.g. skull, headrest), resulting in class imbalance. Intersection over Union (IoU) captures the overlap between predicted and ground truth masks. F1 score takes both precision and recall into account, giving an equal weight to false positives and false negatives. This is particularly useful in brain segmentation where false positives (i.e., misclassifying hemorrhage as not hemorrhage) and false negatives (i.e., missing parts of hemorrhage) can both have significant impact downstream.

By considering each metric, we can obtain a more comprehensive understanding of the strengths and limitations of our model.

4.6 Model and Unet Architecture

UNET is a specialized CNN designed for medical image segmentation tasks. Its encoder-decoder structure captures contextual information and precisely localizes segments, generating pixel-wise maps.

Dataset	Shape
X_train	768x512x512x3
y_train	768x512x512x1
X_val	86x512x512x3
y_val	86x512x512x1

Table 1: Breakdown of X_train, y_train, X_val, and y_val datasets

4.7 Methodology & Results

We utilized a sigmoid activation function to assign a probability to each pixel value. To classify each pixel as either white or black, a threshold value had to be determined. We tested various threshold values on the test set and selected the one that yielded the best F1-score and IoU results. The optimal threshold was found to be 0.245, providing the best balance between F1-score and IoU for our binary segmentation task.

$$t_{optimal} = 0.245 \tag{1}$$

After 50 epochs of training, with a batch size of 6 (due to memory limitation), we found a consistent, well performing model. These were the best performance metrics achieved using the optimal threshold value.

These metrics indicate that the overall segmentation performance of our model is quite promising. The relatively high pixel accuracy showcases the model's effectiveness in correctly classifying most of the pixels in the images. Meanwhile, the IoU and F1-score values suggest a good balance between precision and recall, demonstrating the model's

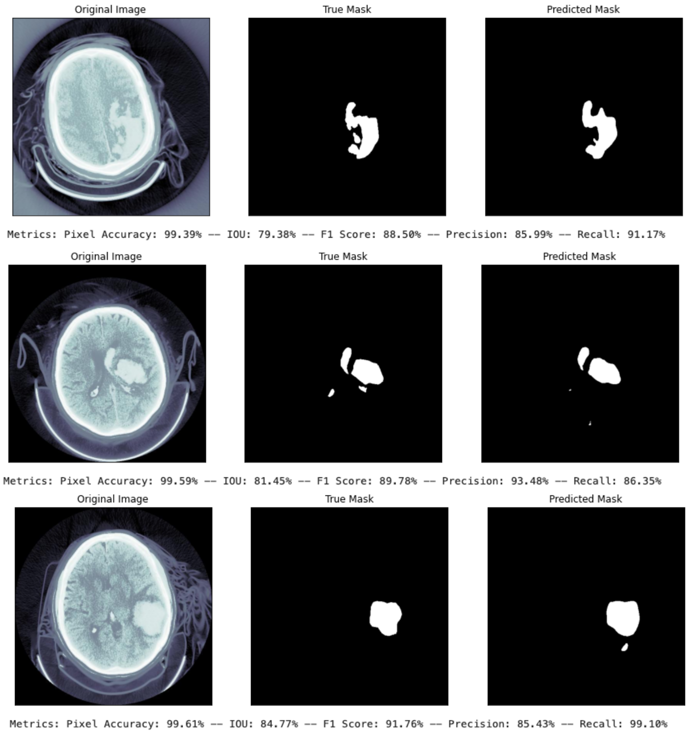
Metric	Value
IoU	0.401
F1-score	0.577
Pixel Accuracy	0.990

Table 2: Best performance metrics for UNET binary segmentation

ability to accurately identify the regions of interest in the medical images.

4.8 Examples

In this section, we present examples of brain CT scans, their corresponding binary masks, and the predicted masks with their corresponding metrics. Three examples of brain CT scans are shown. For each example, we also show the values of the pixel accuracy, IOU, F1 score, precision, and recall.



```
Metrics: Pixel Accuracy: 99.39% -- IOU: 79.38% -- F1 Score: 88.50% -- Precision: 85.99% -- Recall: 91.17%
Metrics: Pixel Accuracy: 99.59% -- IOU: 81.45% -- F1 Score: 89.78% -- Precision: 93.48% -- Recall: 86.35%
Metrics: Pixel Accuracy: 99.61% -- IOU: 84.77% -- F1 Score: 91.76% -- Precision: 85.43% -- Recall: 99.10%
```

Figure 9: Examples of brain CT scans, their corresponding binary masks, and the predicted masks with their corresponding metrics

As shown in the figures above, the predicted masks generally match the true masks very well, with high pixel accuracy, IOU, and F1 scores. Our model achieves good performance in the binary segmentation of brain CT scans, demonstrating the potential of deep learning for medical image analysis.

5 Appendix

5.1 Acknowlegdements

We would like to thank Zeta Surgical for providing data and a fantastic basis for our project this semester.

5.2 GitHub Repositories

- Model Development - lrav35/ML_7243
- Web Application POC - lrav35/brain-segmentation-app

References

[1] Y. Aytar, L. Castrejon, C. Vondrick, H. Pirsiavash, and A. Torralba. Cross-modal scene networks. *PAMI*, 2016.

[2] K. Bousmalis, N. Silberman, D. Dohan, D. Erhan, and D. Krishnan. Unsupervised pixel-level domain adaptation with generative adversarial networks. In *CVPR*, 2017.

[3] Murtadha D. Hssayeni, M. S., Muayad S. Croock, Ph. D., Aymen Al-Ani, Ph. D., Hassan Falah Al-khafaji, M. D., Zakaria A. Yahya, M. D., and Behnaz Ghoraani, Ph. D. Intracranial Hemorrhage Segmentation Using Deep Convolutional Model. 2019.

[4] Kolařík, M.; Burget, R.; Uher, V.; Říha, K.; Dutta, M.K. Optimized High Resolution 3D Dense-U-Net Network for Brain and Spine Segmentation. 2019.Appl. Sci. 2019.

[5] Jacopo Teneggi, Paul H. Yi, and Jeremias Sulam. Weakly Supervised Learning Significantly Reduces the Number of Labels Required for Intracranial Hemorrhage Detection on Head CT. 2022.

[6] Olaf Ronneberger, Philipp Fischer, Thomas Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation.Medical Image Computing and Computer-Assisted Intervention (MICCAI), 2015.