Brain Hemorrhage Classification with Computed Tomography (CT) Images

David Pogrebitskiy Scott Biggs Thomas Walewski Jared Garfinkel Karenna Ng Sarah Baker
Department of Mathematics, Northeastern University

December 9, 2022

Abstract

Intracranial hemorrhages are life threatening injuries which are primarily diagnosed with computed tomography (CT) scans. The brain is imaged in slices, and radiologists look through the image stacks to identify if the patient has a hemorrhage. This process is slow and error prone, which could be the difference between life and death for a patient. In this paper, we develop and test machine learning algorithms to identify the occurrence and type of hemorrhage present in a CT scan, increasing speed and accuracy of the diagnosis. Our best model uses a convolution neural network, and we found it to be 70% accurate in predicting the hemorrhage type over 1463 downscaled test images.

1 Introduction

The ability to identify the type of hemorrhage quickly and reliably in a patient is a critical step between admittance and treatment. Typically, this would require a lengthy process of imaging and identification by skilled medical professionals. This process is done through computerized tomography (CT) scans, which consist of a series of x-ray images around the head to generate an image. A program that could quickly and reliably identify the type of hemorrhage in a patient from a single CT scan image would be a critical time saver in rapid response to life threatening hemorrhages. It would also enable hospitals without the specialized personnel and equipment of larger locations to offer lifesaving care in time. The goal of this project is to develop a model that can do this, using gigabytes of CT scan images.

In this paper, we present a method to classify the type of hemorrhage in a given CT scan. Data was cleaned using Java and Python. We compared the performance of several models, including a SoftMax multiclass classification model and a handful of iterations of neural networks and convolutional neural networks (CNN), in Python.

2 Related work

Advanced machine learning in action: identification of intracranial hemorrhage on computed tomography scans of the head with clinical workflow integration [1] Working with about 40,000 full head CT scans, researchers were able to develop a CNN that can identify intracranial hemorrhage on the 3-dimensional head scan. They took the full stack of images from each CT scan through five convolutional layers and two fully connected layers before producing an output. In a clinical setting, their algorithm was able to reduce the time of diagnosis by 96% across the outpatient setting, however there were limitations in the accuracy of the training data causing higher rates of false readings. Overall, their work produced a solid starting place for analyzing 3-dimensional image stacks of the brain for hemorrhages.

Detecting hemorrhage types and bounding box of hemorrhage by deep learning [2] Deep learning models can detect intracranial hemorrhages on CT scans with some accuracy, but we can improve that by adding a bounding box to the model. These researchers found that by identifying a bounding box around the area of the hemorrhage in 3-dimensional space, the accuracy of the deep learning model increases to over 90% in determining the type of hemorrhage. They used a deep learning model to identify the bounding box, and followed with a one-stage detecting architecture to identify the type of hemorrhage in the bounding box. Their work was able to significantly increase the accuracy of hemorrhage identification.

Deep learning algorithm in detecting intracranial hemorrhages on emergency computed tomographies [3] These researchers used a previously trained artificial intelligence algorithm to identify hemorrhages in CT scans. The results of the algorithm were compared to the official reports from the hospitals and to a second opinion from a neuroradiologist that had not seen any of the images before. They found that the algorithm only overcalled 2% of the scans as hemorrhages, and correctly labeled hemorrhages on scans originally reported as normal for an increase in accuracy of 12%. Their study verified the effectiveness of an AI algorithm to work in conjunction with a doctor's reading of the CT scan.

3 Formulation

The dataset included four different CT windows: brain_bone, brain, max_contrast, and subdural. A preliminary model was developed using logistic regression on three hemorrhage categories to test which imaging window performed best. The brain_bone_window was chosen for our later models because offered the highest accuracy.

The first step in the modeling is reading the image files. For this, an image file is decomposed into its pixel information in a matrix. However, the pixels of colorized images are represented by three eight-bit color values for red, green, and blue (RGB) respectively. This leads colorized images to be decomposed into three $n \times m$ matrices where each matrix is the dimensions of the original image file and each of the three matrices contain values from 0 to 255 describing the vibrancy of each RGB value. However, since the image files are generated from x-ray scans, they do not contain color. In this case each image can be as one $n \times m$ matrices with each index containing values from 0 to 255 describing the brightness of the pixel from black to white.

In the case that the images need to be downscaled, an algorithm can be applied to each of the jpeg images to make a new array where each value in the array is the average of its surrounding indices in the original array (1). Ideally, this would only be necessary if the processing for the model required a large amount of computation time. However, downscaling the images should be done with caution as it might affect the results of the regression and the outcome of the confusion matrix. Two different methods of downscaling were used. The first method takes every nth pixel in the array and then compiles them into a new image. This resulted in downsampled images which have "copied" patterns appearing. The second follows a method where each entry in the output matrix is taken by the local average of the input matrix. In this case the size of the local area is determined by the amount of down sampling.

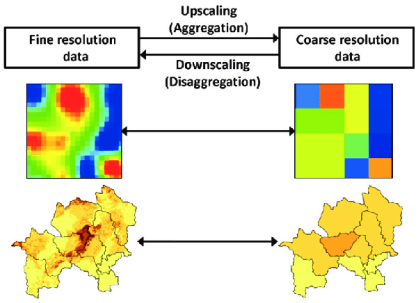

Figure 1: An example of array downsampling. [4]

The downsampling process is guided by a .csv containing the data labels, which was provided by Zeta Surgical. Three major types of models were attempted. These include, logistic regression, a neural network, and convolution neural network (CNN). The logistic regression model was done using the Scikit learn library in Python and the neural networks were enabled through the TensorFlow package. The first neural network follows a simple pattern of two flattening layers and two dense layers. The CNNs follow a more extensive process using downscaled images. The neural network and CNN models are unique architectures with room for further optimization.

Reviewing the literature on CT scan-based hemorrhage detection, we noticed that recent models use several different CT scans, taken from different heights up the subject's head. Replicating this was a challenge for us, since our image data does not make clear which scans are from what elevation on which subject. We resolved this issue by training our model to run on only

a single image slice, rather than needing several [1]. However, if these models are revised to train on stacks of CT scan images, rather than individual slices, they would likely improve significantly.

3.1 Softmax

We wrote a program in Java to parse through the CT scan images and knock out ones that were excessively dark or light. Empty space shows on a CT as black, and bones and other dense tissue show as white. Taking the average brightness of the entire image, scans of sufficient quality were found to be about twice as bright as those that were not usable. A ceiling of about 35 million units was set, and the 2000 best scans with the 'normal' label were chosen for training and testing. This process cleaned out images with lots of empty space and with excessive dense tissue. For example, low height scans that included the upper jaw and eye sockets, and high height scans with little to no brain tissue imaged. We believe that this cleaning step helped the model's overall performance. Then, we ran a SoftMax multiclass regression on these cleaned images.

3.2 First Convolutional Neural Network

For our first CNN model, we downscaled the CT images to be 128×128. We downscaled by a factor of 16 along the 'x' (horizontal) axis for two primary reasons. Firstly, to eliminate noise. A high-resolution CT scan can include many extraneous features, it can show the texture of brain tissue or folds, as well as traces of skin, hair, or even dental work. Secondly, to decrease training time, since running the algorithm on thousands of full resolution images was very slow with our GPU. One of these images, shown in figure 2, shows a squashed image with features repeated four times along the horizontal axis. The CNN trained on these images significantly improved on the results of the SoftMax regression.

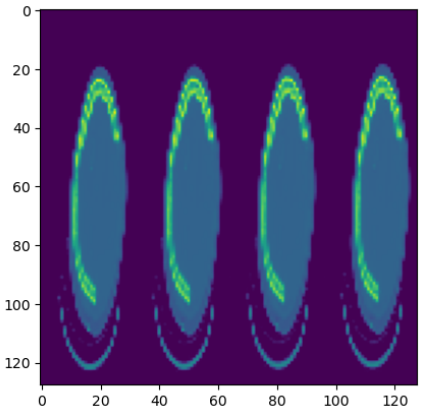

Figure 2: Downsampled scan, with features repeated four times along the horizontal axis.

3.3 Second Convolutional Neural Network

This model was trained on CT images downscaled to 128×128, a 4 times reduction on both axes. We downscaled the images for the same reasons discussed earlier. The down sampling took the average value of the pixels in a 4×4 pixel block in the original sized image and gave one pixel that average value in the new downscaled image. One of these images is shown below in figure 3. We thought that this approach would yield better results than the earlier CNN model because the shape of the image, and therefore the location of features therein, would be better maintained. We hoped that this would translate to more accurate results. However, it only offered a slight improvement over the previous CNN model.

3.4 Third Convolutional Neural Network

Small changes were made to improve upon the performance of the previous CNN model. Namely, the dimesionality of the output space of the first convolutional layer was increased, filtering the input more gradually. We hoped this would result in less information lost during the initial convolutions, and thus higher accuracy. This model was also run on the same average-value downscaled data as the second CNN model (above).

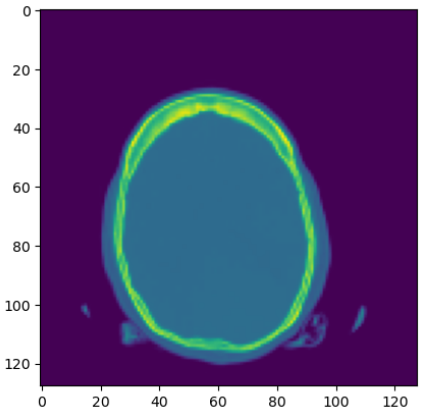

Figure 3: Downsampled scan, maintaining location of features on both axes.

4 Implementation

Here we discuss the implementation of the machine learning algorithms used in the development of our models.

SoftMax The SoftMax (logistic) regression algorithm was from the Scikit learn package with default settings. As the data is multiclass, the "multinomial" setting was used to minimize the multinomial loss across the distribution.

Neural Networks The initial neural network trial had no convolution layers. This network contains a flattening layer followed by two dense layers that decrease the output space to 1000 and 6 respectively. This algorithm was run on the original images at full scale (512×512 px), and on the downsampled 128×128 data to test if run time could be reduced without sacrificing accuracy. A diagram of the layers is shown in figure 4.

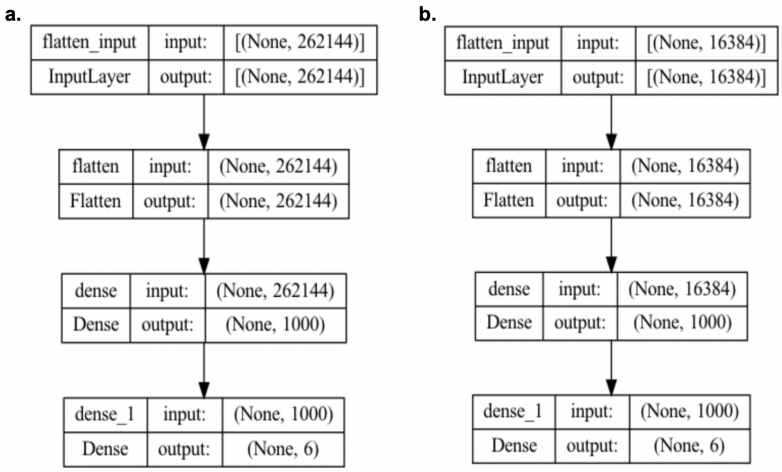

Figure 4: Layer diagram of neural networks. **(a)** Designed for full scale data. **(b)** Designed for downscaled data.

Building on the previous trials, we designed three new CNN models. They were all trained on 128×128 downsampled images, and had nine model layers. First are two sets of a convolutional layers, followed by a maximum pooling layer. These layers apply a filter to each image, and then reduces its dimensionality. A dropout layer is applied to reduce overfitting of the model by dropping 25% of the data. Then the data is flattened and run through a dense layer, where 50% is randomly dropped. Finally, a dense layer classified the data.

There were subtle differences between each of the three runs. The first trial was run on the original downscaled data. We later found that our initial downscaling method simply took every 16th pixel in the x-direction as opposed to taking the average of every 4×4 pixel square, and so the dataset was re-downscaled using the latter method and had a second trial where this new data was run through the same CNN algorithm. The third CNN contained the same order of layers, but the dimensionality of the output space of the convolutional layers was changed to scale down more gradually. A diagram of the layers is shown in figure 5.

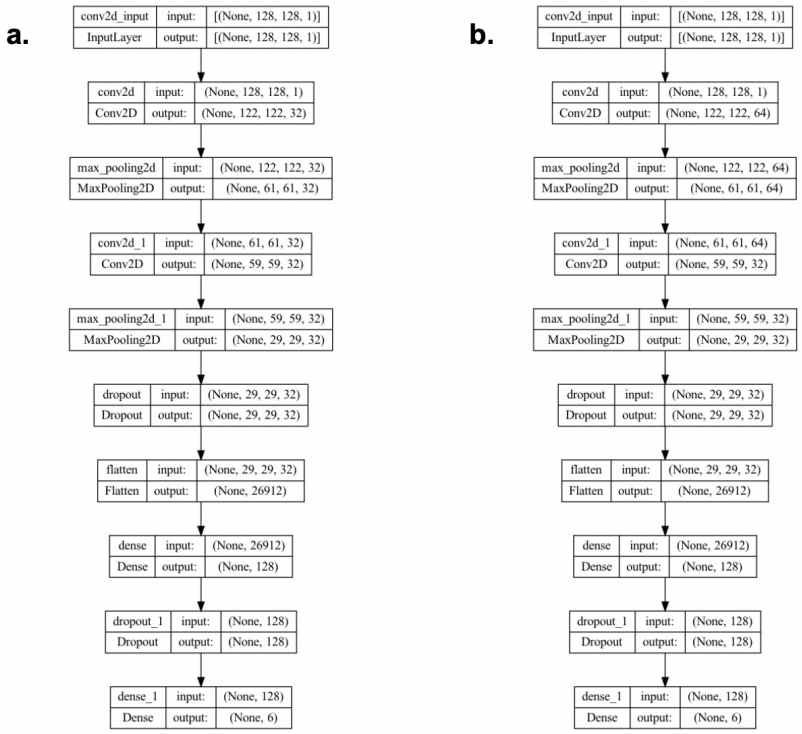

Figure 5: Layer diagram of convolutional neural networks. **(a)** Algorithm used for first two runs. **(b)** Algorithm used on third run. Note the output size change in the first convolutional layer.

5 Results

5.1 Initial Logistic Regression and Softmax

The confusion matrix from the initial logistic regression test data is shown in figure 6. It shows that each of the categories has an accuracy rating from around 37% to 68% accuracy for the normal category. Note that three categories have accuracy ratings of around 37% while the two other non-normal categories have accuracy ratings of around 52%. By taking the trace of this confusion matrix and dividing it by the sum of all entries, the overall accuracy of the model on test data is found to be roughly 53%. However, due to the comparatively high accuracy in the normal category, the accuracy of the model is significantly supported by the model's ability to detect normal cases. Since the objective of this model is to accurately determine the type of hemorrhage present in a CT scan, it proves quite poor. The ability to predict a hemorrhage should matter more than the ability to not predict a hemorrhage.

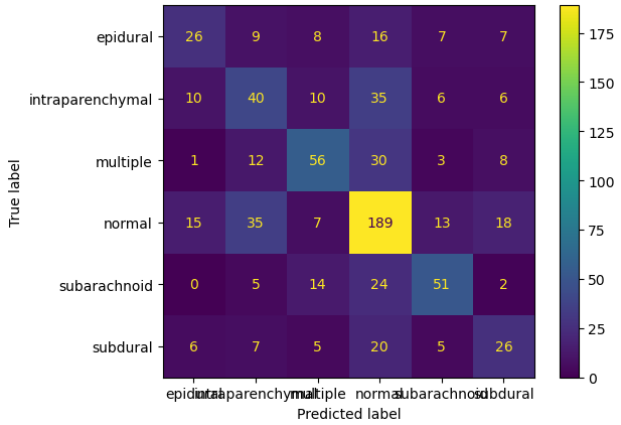

Figure 6: Logistic regression of sample data confusion matrix.

5.2 First Convolutional Neural Network

The confusion matrix for the first CNN can be seen in figure 7. This model ran with a higher number of images, 1464, but these images were down sampled 16 times, turning the original 512×512 pixel images into 128×128 pixel images. This neural network had an overall accuracy rating of 67.62%. In this model the "normal" classification accuracy rating of 84% boosts the overall model accuracy, but does not support its deficient classification abilities in other categories. This model's accuracy rates in non "normal" categories ranges from roughly 44% to 71% , an increase of 10-20 percentage points over the logistic model.

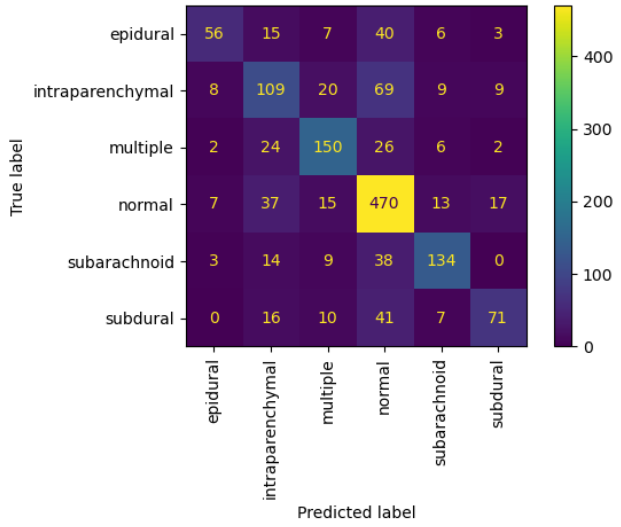

Figure 7: CNN model 1 confusion matrix.

It should be noted that the logistic model and the CNN have similar performance patterns. In the logistic model, epidural, intraparenchymal, and subdural had the poorest classification scores, around 37%, and multiple and subarachnoid had better scores at around 52%. Similarly, in the convolution neural network, epidural, intraparenchymal, and subdural have similar scores, around 47%, while multiple and subarachnoid have scores around 70%. If these classes are grouped together we see that the former group of three (epidural, intraparenchymal, and subdural) showed a 28% increase in accuracy between the two models while the other group (multiple and subarachnoid) shows a 33% improvement. From this trend one could infer that better models would have an easier time classifying "multiple" and "subarachnoid" classes before "epidural", "intraparenchymal", and "subdural" classes. Perhaps weighting the model to prioritize these classes could correct this behavior.

Logistic								
True Label Down	epidural	intraparer	multiple	normal	subarachn	subdural	**Accuracy Rate**	**Total Accuracy Rate**
epidural	26	9	8	16	7	7	35.62%	53.01%
intraparenchymal	10	40	10	35	6	6	37.38%	
multiple	1	12	56	30	3	8	50.91%	**Accuracy Without Normal**
normal	15	35	7	189	13	18	68.23%	42.94%
subarachnoid	0	5	14	24	51	2	53.13%	
subdural	6	7	5	20	5	26	37.68%	**Total Entries**
								732
CNN								
True Label Down	epidural	intraparer	multiple	normal	subarachn	subdural	**Accuracy Rate**	
epidural	56	15	7	40	6	3	44.09%	**Total Accuracy Rate**
intraparenchymal	8	109	20	69	9	9	48.66%	67.62%
multiple	2	24	150	26	6	2	71.43%	**Accuracy Without Normal**
normal	7	37	15	470	13	17	84.08%	56.10%
subarachnoid	3	15	9	38	134	0	67.34%	**Total Entries**
subdural	0	16	10	41	7	71	48.97%	1464

Figure 8: Accuracy calculations for the logistic regression and CNN model 1.

5.3 Second Convolutional Neural Network

This model used better downsampled data, where images were downscaled to 128×128 by the average of every 4×4 pixel square. Surprisingly, there was a negligible change in overall and class specific performance.

5.4 Third Convolutional Neural Network

The third CNN was trained on the 4×4 downsampled data as well. We changed the dimensionality of the output of this first convolutional layer to more gradually filter the input into its predicted class. The confusion matrix and calculations can be seen in figures 9 and 10, respectively.

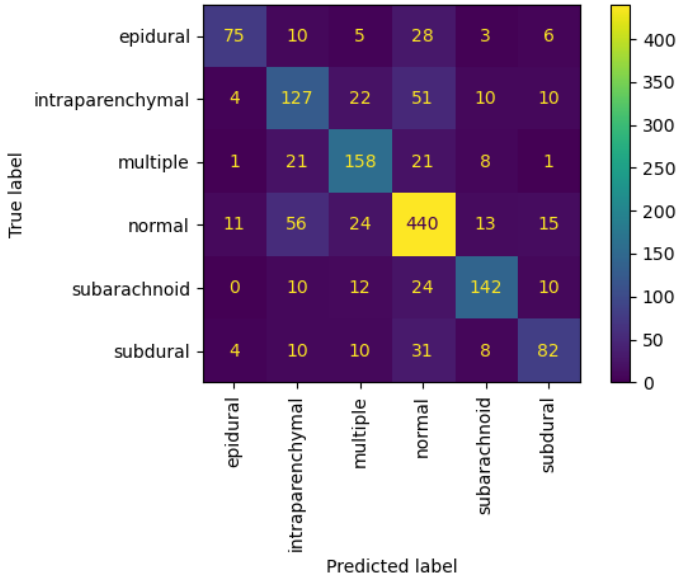

Figure 9: Model 3 CNN confusion matrix.

Despite a meager 2% overall accuracy improvement over the previous CNN, the accuracy for each of the classes, except ”normal”, has significantly improved. The ”normal” class accuracy actually dropped, from 84% to about 79%. The other categories show increases of up to 10%.

Based on the results of previous mdoels, we had theorized that increases in accuracy would disproportionately benefit "multiple" and "subarachnoid" classification. However, that is not the case in the updated model. In this model, the other three non-"normal" classes saw more increases in classification accuracy than "multiple" and "subarachnoid". Even though this model has a decreased ability to detect "normal" cases, its increased ability to accurately classify other cases increases its usefulness.

True Label Down	epidural	intraparench	multiple	normal	subarachnoi	subdural	Accuracy Rate	Total Accuracy Rate
epidural	75	10	5	28	3	6	59.06%	69.99%
intraparenchymal	4	127	22	51	10	10	56.70%	Accuracy W/o Normal
multiple	1	21	158	21	8	1	75.24%	63.85%
normal	11	56	24	440	13	15	78.71%	Total Entries
subarachnoid	0	10	12	24	142	10	71.72%	1463
subdural	4	10	10	31	8	82	56.55%	

Figure 10: Accuracy calculations for CNN 3.

6 Acknowledgement

We would like to thank Zeta Surgical for providing the CT scan images and data labels we used. We would also like to thank Prof. He Wang for lectures and resources, TAs, and the friends we made along the way.

References

[1] M. Arbabshirani, B. Dornwalt, G. Mongelluzzo, J. Suever, B. Geise, A. Patel, and G. Moore. Advanced machine learning in action: identification of intracranial hemorrhage on computed tomography scans of the head with clinical workflow integration. *Nature*, 2018. 1, 3

[2] O. Faruk Ertugrul and M. Fatih Akil. Detecting hemorrhage types and bounding box of hemorrhage by deep learning. *BSPC*, 2022. 1

[3] A. Kundisch, A. Honning, S. Mutze, L. Kreissl, F. Spohn, J. Lemcke, M. Sitz, P. Sparenberg, and L. Goelz. Deep learning algorithm in detecting intracranial hemorrhages on emergency computed tomographies. *PLoS ONE*, 2021. 2

[4] N.W. Park, Y.S. Kim, and G.H. Kwak. An Overview of Theoretical and Practical Issues in Spatial Downscaling of Coarse Resolution Satellite-derived Products *Korean Journal of Remote Sensing*, 2019. 2