DeepASPECTS

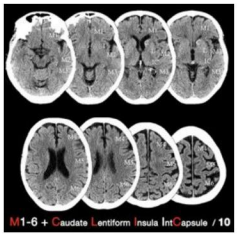
Rukhsana Yeasmin <ryeasmin>, Salmonn Talebi <stalebi>, and Tony Joseph <tonyjm>

Abstract

The Alberta Stroke Program Early CT Score (ASPECTS) is widely used to assess early ischemic changes for stroke victims. Currently the process of reviewing ASPECT images is tedious and susceptible to human error. We propose a deep learning model to automatically classify patient's CT scans with the correct aspect score. We use transfer learning with VGG16 to evaluate CNN models on 3 specific regions (M4, M5, and M6) of the brain. Two models were evaluated: a functional model that simultaneously predicts M4, M5, and M6 for the left and right side of the brain while the other had two sequential models which were used to independently evaluate 3 regions of left and right brain. Our best model achieves ~97.7% validation accuracy with a sensitivity of 0.79 and specificity of 0.98 on test data.

Introduction

Evaluation of non-contrast CT of the patient's head is crucial to assess the severity of an ischemic stroke (stroke caused due to restricted or blocked blood flow). Alberta Stroke Program Early CT Score (ASPECTS) is a common medical standard used to communicate the severity of a stroke and to determine treatment options[2]. The score is based on the presence or absence of ischemia (blood flow blockage) on a non-contrast CT of the brain. There are 10 specific locations (**Fig: 1A**): (caudate (C), putamen (P), internal capsule (IC), insula (I), and 6 areas of the middle-cerebral artery territory (M1-M6)) which are evaluated on each side of the brain to determine the score. ASPECT is scored for each side (R or L) out of 10 in which 1 point is deducted for each of the 10 locations affected. The regions are highlighted in Fig. (1). Calculated score is used to determine the treatment options and the prognosis.



STIR ID	Age	Gender	ASPECTS (Right)											ASPECTS (Left)										
			C	L	IC	I	M1	M2	M3	M4	M5	M6	Total	C	L	IC	I	M1	M2	M3	M4	M5	M6	Total
M-01	75	1	1	1	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	10
M-03	58	1	0	1	0	1	0	0	0	0	1	0	7	0	0	0	0	0	0	0	0	0	0	10
M-04	75	2	0	0	0	1	0	1	0	0	0	0	8	0	0	0	0	0	0	0	0	0	0	10
M-05	45	1	0	1	0	1	1	1	0	1	1	0	4	0	0	0	0	0	0	0	0	0	0	10
M-06	54	1	0	1	0	0	1	1	0	1	1	0	5	0	0	0	0	0	0	0	0	0	0	10
M-07	64	2	1	1	0	0	0	0	0	0	0	0	8	0	0	0	0	0	0	0	0	0	0	10
M-08	83	2	0	1	0	1	1	1	0	1	1	0	4	0	0	0	0	0	0	0	0	0	0	10
M-09	78	2	0	0	0	1	0	1	1	0	1	1	5	0	0	0	0	0	0	0	0	0	0	10

Fig. 1: (A) ASPECTS CT Scan reference slices. (B) Sample ASPECT score data set (stroke detected at right brain).

Related work

There have been a few attempts to use machine learning to classify ASPECT scores from CT scans. The most successful one is available as a commercial product called eASPECTS, which seems to have marginally better accuracy compared to 3 independent radiologist's[3] evaluation. Additionally, there have been attempts to segment the scans and use ML approaches like

random forest[4] to determine the ASPECT score but these have not been as successful. We did not find any reference of using deep neural networks for this class of problems.

Dataset and Features

For a given patient we receive approximately 30 CT slices in an imaging format called DICOM. Each slice will represent a 5 mm horizontal slice between the neck and top of the head. Only a subset of the CTs (8-13 slices) will contain the regions of interest. An expert radiologist has provided 171 patient scans with labeled slices. Each slice is accompanied by a 20 set label for all 10 regions of interest split between left and right side of the brain. Final processed dataset contains ~1700 lower brain slices, of which ~900 had ischemia. We performed a (80%, 10%, 10%) split of the shuffled labeled data into training, dev and test set.

Data Pre-processing

CT scans are measured in Hounsfield scale (HU), a measure of radiodensity. For a given patient CT scan, first we need to get slices in right order and then convert to HU unit[5, 8]. We can do this using information stored in the metadata: order slices using “InstanceNumber”, multiply by “RescaleSlope” and add “RescaleIntercept”. Next step is to filter out values outside the range of brain parenchyma, which typically ranges between -100 to 100 HU. However, to account for differences in CT scanners, we set the range from -200 to 200 HU. Any value outside this range has been set to the value of air (i.e., -1000 HU). Next, we normalize the image slices: values in the range -200 to 200 HU are scaled to 0.0 - 1.0, values below -200 HU are set to 0.0, and values higher than 200 HU are set to 1.0. Next step is to zero center all the images, where the mean is calculated from all the images of whole dataset. Fig. (2) shows sample processed slices.

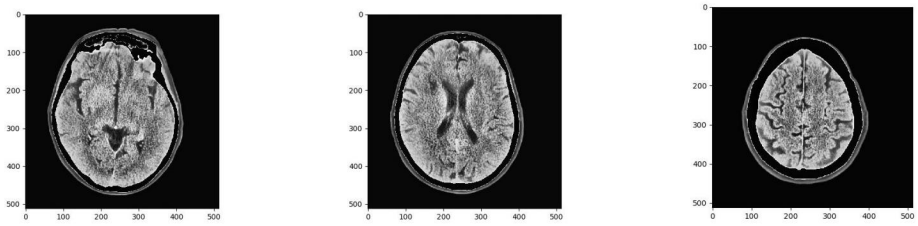


Fig. 2: Sample processed CT scan slices

Methods

Given the limited amount of patient data we have (171 patients), it will be difficult to train a deep neural network from scratch. Hence, we apply transfer learning from pre-trained VGG16[11] from Keras (github link). We have frozen all convolutional layers of VGG16 model and removed FC and dense layers. We experimented with 2 different types of models to train for lower part of the brain (M4, M5, M6), with frozen VGG16 as base:

1. **Sequential model:** Goal is to train one model for each side (left/ right) of the brains. For this model, a FC layer follows the output from VGG layer, followed by a 256-unit dense

layer with RELU activation. A final 3-output dense layer node with sigmoid activation is used to predict the M4, M5, M6 regions.

2. **Functional model:** A FC layer followed by a 256-node dense layer with RELU activation was inserted. A 2-output dense layer with softmax activation is connected to the 256-node dense layer to predict which side of the brain (left/ right) is impacted. Output from softmax layer is concatenated with 256-node dense layer, followed by a 6-unit dense layer with sigmoid activation to predict the M4, M5, M6 regions at both left and right side of brain.

Batch normalization was performed after each dense layer explained above, which helped improve model performance significantly. We experimented with additional dense layers with dropout. However, model performance decreased with this setup, possibly due to limited data size. Since our slices are grayscale images, we applied the same slice to all 3 channels of the VGG16 input layer.

Experiments/Results/Discussion

Hyperparameter tuning:

We evaluate the best set of hyperparameters (batch size, epochs, learning rate, and dense layer size) by using a random grid search. Fig (3) shows validation accuracy for different combination of parameters. Final set of parameters were selected based on max validation accuracy (~ 97.5%): lr 0.001, batch size 32, number of epochs 20, 256 dense layer units.

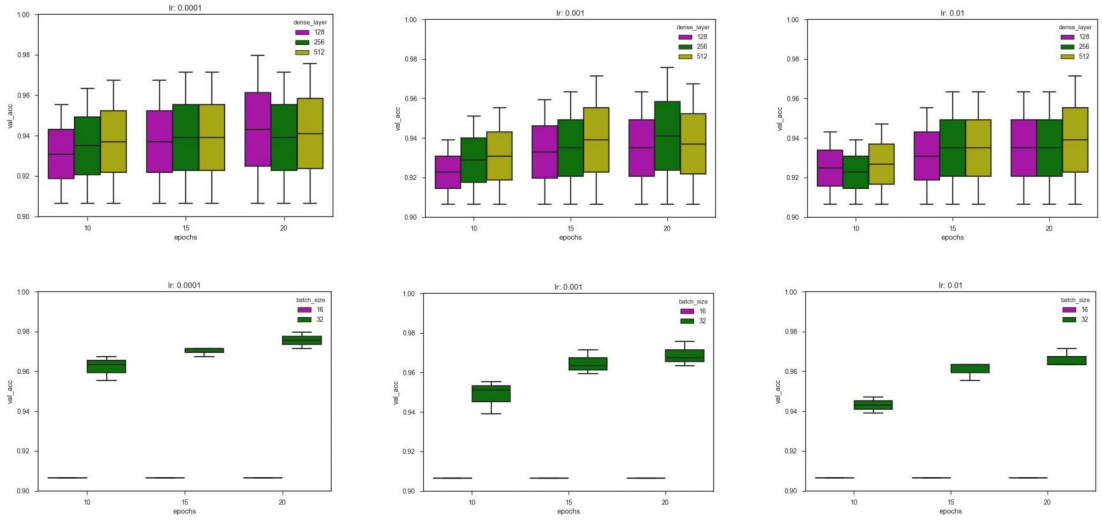


Fig 3: Validation accuracy for different learning rate, epochs and for varying number of (A) dense layer units, (B) Batch size.

Performance Metrics:

Keras reported accuracy was used to evaluate model performance during training. Fig. (4) shows accuracy and loss plots from Sequential and Functional models extracted from kears reported model train history.

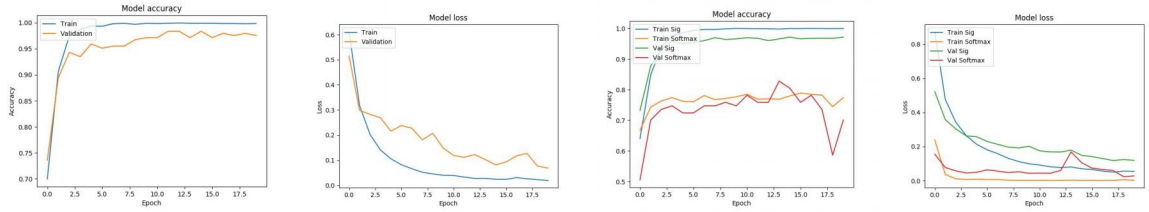


Fig. 4: Accuracy and loss plot from (A) Sequential model trained for left brain (sigmoid with binary cross-entropy). (B) Functional model trained for entire lower brain (softmax with categorical cross-entropy for left/ right selection, sigmoid with binary cross-entropy for region selection).

We had to be careful in selecting appropriate model performance metrics for stroke detection. It is critical that the model does not misclassify a healthy person as having stroke. Similarly model should detect all affected regions of brain accurately for a patient with stroke. Hence sensitivity and specificity are two important performance metrics we need to evaluate. Along with those, we used exact matching score and hamming loss reported by scikit-learn. We measure these metrics on a slice level instead of a patient level due to limited data. Below table summarizes performance scores on test data:

	Sequential (left)	Sequential (right)	Functional (whole)
Specificity (%)	98.59	95.16	98.11
Sensitivity (%)	78.57	65.22	61.54
Exact-match-score (%)	98.04	94.51	97.47
Hamming-loss (%)	1.96	5.49	2.53

Error Analysis:

For most of the mispredicted slices, model prediction had partial matching with actual results, which resulted in comparatively lower sensitivity score. For some slices, multiple neighbor regions were impacted, but the model failed to detect all of those. In some other cases, the model picked more regions as impacted than actual results. In some rare cases, when a region is impacted there is a possibility of blood in nearby regions which might be low enough to be ignored by a doctor. Similarly scoring method may vary for different doctors. All of these will impact model performance. Old strokes may impact the performance of the model as well. These are not considered for ASPECT score calculation. However, model may fail to ignore these cases because of the trace of blood. Scans with motion (patient moved while taking the scan) or tilt also impact model performance, specifically it will impact the sensitivity. However we should be able to reduce this error with more data.

With enough variation in the dataset model should be able to improve performance on all of these cases. To partially overcome the data issue, we applied careful augmentation. Considering human brain is symmetric for stroke detection, we applied mirroring on impacted slices, and flipped labels accordingly. This helped us improve model performance partially.

Conclusion/Future Work

Data preprocessing improvements:

Acquiring additional labeled data for each patient is a cumbersome process. Of the 30 patient slices less than 15 slices actually contain information for the 10 regions, which can be mapped to 8 reference slices shown in Fig. (1). We plan to evaluate two methods to determine if a slice contains any of the 10 regions. First approach is to create a standard set of CT scans, each of which contains only the desired set of slices. Union of all first slices from the standard scans are taken into one set and union of all last slices are taken into another set. Each slice of these two standard sets go through same pre-processing as explained earlier. Next, SSIM (structural similarity) score is calculated for each slice of interest from the patients CT scans using the standard set of slices as reference. Slices with high SSIM scores (above pre-set threshold) are selected as candidate slices for model training.

The second method is to use deep learning to train a CNN to determine if a slice is similar to any of the reference slices, and hence should be evaluated for parenchymal changes. We plan to train the model with all 30 CT scan slices in which each slice will be labeled and mapped to a reference slice. The model then should be able to predict if any given slice is similar to a known reference slice containing regions.

Model improvements:

Currently our model can predict only 3 regions (M4, M5, M6). Our next step is to expand the model's capability to predict stroke across all 10 regions. Initial training results for our functional model on all 10 regions look promising. To further improve the model performance on all 10 regions we have to collect more training data. We will continue working with our radiologist to collect more labeled data and further improve the model. Once our model can accurately predict all 10 regions we will use it to predict the full ASPECT score for a patient.

Contributions:

Elizabeth Tong from the Stanford School of Medicine provided us with annotated patient data to use for training. Rukhsana contributed to the development of the preprocessing python data scripts with some support from Tony and Salmonn. Tony contributed to setting up the AWS server and getting it to run our python scripts. Rukhsana, Salmonn, and Tony equally contributed to the model creation, evaluation, hyperparameter search, and test results. Rukhsana, Salmonn, and Tony equally contributed to the final report and poster.

References:

1. http://www.aspectsinstroke.com
2. Barber, P.A., Demchuk, A.M., Zhang, J., Buchan, A.M., & Group, F.T. (2000). Validity and reliability of a quantitative computed tomography score in predicting outcome of hyperacute stroke before thrombolytic therapy. *The Lancet, 355*, 1670-1674.
3. Nagel, S., Sinha, D., Day, D., Reith, W., Chapot, R., Papanagiotou, P., ... & Walter, S. (2017). e-ASPECTS software is non-inferior to neuroradiologists in applying the ASPECT score to computed tomography scans of acute ischemic stroke patients. International Journal of Stroke, 12(6), 615-622.
4. Kuang, Hulin & Najm, Mohamed & Chakraborty,, Debabrata & N Maraj, Nicholas & Il Sohn, Sung & Goyal, Mayank & Hill, Michael & Demchuk, Andrew & Menon, Bijoy & Qiu, Wu. (2018). Automated ASPECTS on Non-Contrast CT Scans in Acute Ischemic Stroke Patients Using Machine Learning. American Journal of Neuroradiology. 10.3174/ajnr.A5889.
5. https://www.kaggle.com/gzuidhof/full-preprocessing-tutorial
6. Mohammad Havaeia, Axel Davy, David Warde-Farley, Antoine Biard, Aaron Courville, Yoshua Bengio, Chris Pal, Pierre-Marc Jodoin, and Hugo Larochelle (2016). Brain Tumor Segmentation with Deep Neural Networks. Medical Image Analysis.
7. Ivana Despotović, Bart Goossens, and Wilfried Philips (2015). MRI Segmentation of the Human Brain: Challenges, Methods, and Applications. Compute Math Methods Med. 2015: 450341
8. Dandu Ravi Varma (2012). Managing DICOM images: Tips and tricks for the radiologist. Indian J Radiol Imaging. 2012 Jan-Mar; 22(1): 4–13
9. Dinggang Shen, Guorong Wu, and Heung-Il Suk (2017). Deep Learning in Medical Image Analysis. Annu Rev Biomed Eng, PMC 2017
10. Brain Tumor Segmentation and Radiomics Survival Prediction: Contribution to the BRATS 2017 Challenge (2018). Isensee et al.
11. Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.