

ONLINE APPENDIX

Machine Learning Approach

SVM is a popular supervised learning algorithm for classification and regression.¹ SVM has proved powerful for several medical imaging applications.^{2,3} Given labeled training data, the algorithm finds a hyperplane that maximally separates the classes. SVM uses kernels to make the separating hyperplane nonlinear. The trade-off between performance on the training data and the test data is regulated by the hyperparameter C.

One of the limitations of SVM is that it may not produce accurate results because of its simplicity and multicollinearity. Ensemble learning is a way to boost the performance of simple learning algorithms. Ensemble learning relies on training a large number of weak models and then combining them to generate a strong model. The most frequently used weak learners are decision tree models. Random forests consist of creating many copies of training data and then applying the weak learner to each copy to create multiple weak models and then ultimately combining them. Using multiple samples of the original dataset, RF reduces the variance of the final model.⁴ Another effective ensemble algorithm is gradient boosting.⁵ In this, we train consecutive trees to fix the errors of the current model and add this new tree to the existing model. The number of trees and the depth of trees are the hyperparameters in GB. Boosting reduces the bias instead of the variance, causing overfitting issues.⁵

Proposed Convolutional Network

Online Fig 1 contains the architecture of the proposed CNN. The proposed network consists of 4 convolutional blocks and 3 dense blocks. Each convolution block consists of a convolutional layer with filter size 3, followed by batch normalization (BN), Rectifier Linear Unit (ReLU) activation, and max pooling. ReLU has several advantages, including stable gradients at the extreme values of optimization.⁶ Stacking convolutional and nonlinear activation functions allows a CNN to model higher order complex feature representation. Feature maps are down-sampled from the previous layer using max pooling. Batch normalization was used after each convolution and dense layer to limit drift of layer activations during training.⁷ Drop-out of 0.5 was applied to all fully connected layers to limit overfitting and add stochasticity to the training process.^{8,9} The number of channels for each convolution layer was considered to be 32. The number of neurons in the dense layer was 32, 16, and 4. Softmax was used as the activation function in the last layer because it is a multiclass classification problem. Cross-entropy was the taken as the loss function, and accuracy was used as the metric. Training was implemented using the Adam optimizer, an algorithm for first-order gradient-based optimization of stochastic objective functions, based on adaptive estimates of lower-order moments.¹⁰ Parameters were initialized using the method developed by He.¹¹ Training was performed with a minibatch size of 64 for 30 epochs. Keras with backend as TensorFlow (https://www.tensorflow.org/api_docs/python/tf/keras/backend) was used

for implementation.¹² Experiments were performed on a single NVIDIA GeForce GTX Titan X (<https://www.nvidia.com/en-us/geforce/graphics-cards/geforce-gtx-titan-x/>).

Online Fig 2 illustrates how each dataset was preprocessed. The segmentation masks obtained after preprocessing are used to extract section-by-section images of the tumor from all the modalities, with each section being 150×150 . Most vote was used to determine the grade of the tumor. The detailed split of the data is given in the Online Table. The hyperparameters of the network were chosen on the basis of the validation data. Data augmentation was performed by rotating the section at a random angle and horizontal/vertical flipping to balance the dataset.

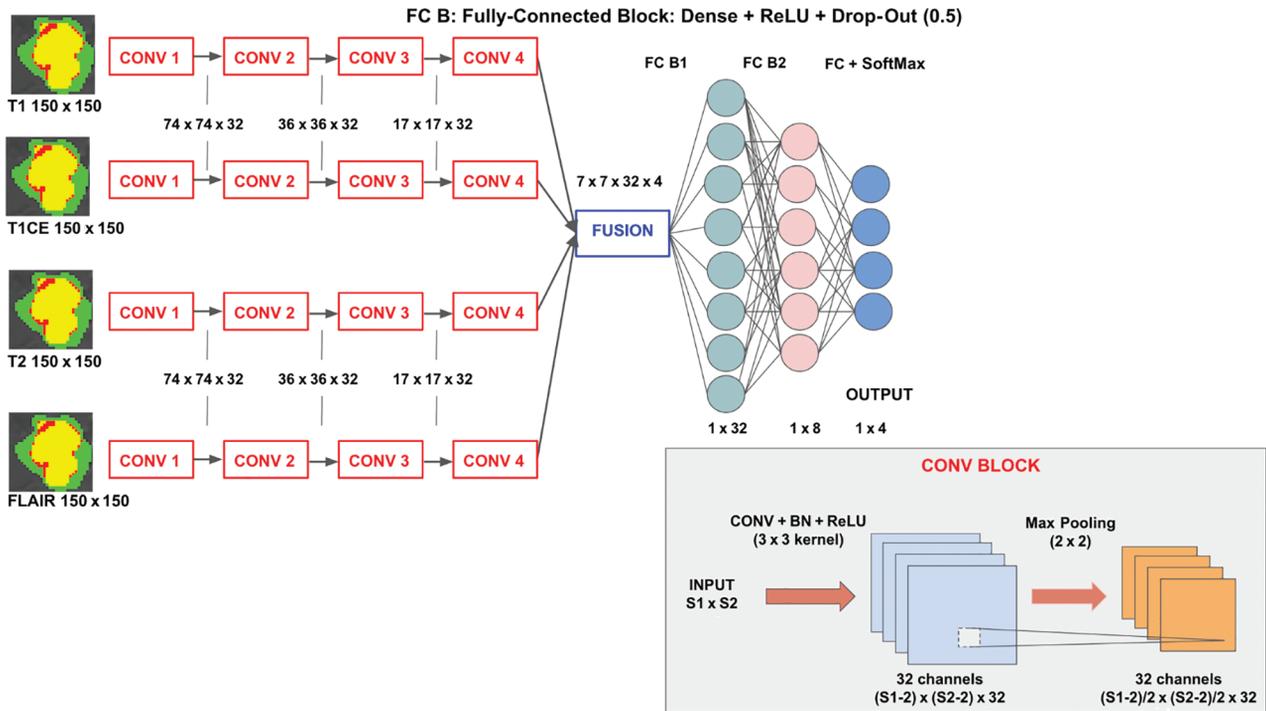
REFERENCES

1. Cortes C, Vapnik V. **Support-vector networks.** *Mach Learn* 1995;20:273–97 CrossRef
2. Larroza A, Moratal D, Paredes-Sánchez A, et al. **Support vector machine classification of brain metastasis and radiation necrosis based on texture analysis in MRI.** *J Magn Reson Imaging* 2015;42:1362–68 CrossRef Medline
3. Blumenthal DT, Artzi M, Liberman G, et al. **Classification of high-grade glioma into tumor and nontumor components using support vector machine.** *AJNR Am J Neuroradiol* 2017;38:908–14 CrossRef Medline
4. Breiman L. **Random forests.** *Mach Learn* 2001;45:5–32 CrossRef
5. Friedman BJ. **Greedy Function Approximation: a gradient boosting machine.** *Annals of Statistics* 2001;29:1189–1232
6. Nair V, Hinton GE. **Rectified linear units improve restricted Boltzmann machines.** In: *Proceedings of the International Conference on Machine Learning*. Haifa, Israel. June 21–24, 2010:807–14
7. Ioffe S, Szegedy C. **Batch normalization: accelerating deep network training by reducing internal covariate shift.** 2015. <https://arxiv.org/pdf/1502.03167.pdf>. Accessed April 10, 2020
8. Baldi P, Sadowski P. **The dropout learning algorithm.** *Artif Intell* 2015;210:78–122 CrossRef Medline
9. Srivastava N, Hinton G, Krizhevsky A, et al. **Dropout: a simple way to prevent neural networks from overfitting.** *J Mach Learn Res* 2014;15:1929–58 <https://dl.acm.org/doi/abs/10.5555/2627435.2670313>. Accessed April 10, 2020
10. Kingma DP, Ba JL. **Adam: a method for stochastic optimization.** <https://arxiv.org/abs/1412.6980>. Accessed April 10, 2020
11. He K, Zhang X, Ren S, et al. **Delving deep into rectifiers: surpassing human-level performance on ImageNet classification.** In: *Proceedings of the IEEE International Conference on Computer Vision*, Santiago, Chile Dec. 7, 2020–Dec. 13, 2020, 2015:1026–34
12. Arbadi M, Agarwal A, Barham P, et al. **Tensorflow: large-scale machine learning on heterogeneous distributed systems.** <https://arxiv.org/abs/1603.04467>. Accessed April 10, 2020
13. Dogdas B, Shattuck DW, Leahy RM. **Segmentation of skull and scalp in 3-D human MRI using mathematical morphology.** *Hum Brain Mapp* 2005;26:273–85 CrossRef Medline
14. Smith SM, Jenkinson M, Woolrich MW, et al. **Advances in functional and structural MR image analysis and implementation as FSL.** *Neuroimage* 2004;23(Suppl 1):208–19 CrossRef Medline
15. Wang G, Li W, Ourselin S, et al. **Automatic brain tumor segmentation based on cascaded convolutional neural networks with uncertainty estimation.** *Front Comput Neurosci* 2019;13:56 CrossRef Medline

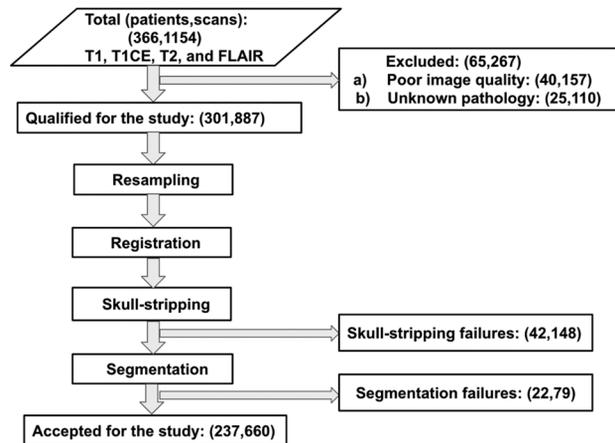
Online Table: Data split into training, validation, and testing^a

Grade	Training	Validation	Testing
I	(11, 17)	(3, 7)	(3, 3)
II	(41, 100)	(9, 27)	(9, 17)
III	(32, 119)	(7, 38)	(7, 27)
IV	(79, 204)	(18, 48)	(18, 53)

^aThe first and second integers in the parentheses correspond to the number of patients and the number of scans, respectively. The splitting of data was performed on the number of patients.



ONLINE FIG 1. Proposed multichannel CNN architecture for feature extraction. There are 4 convolutional layers and 3 densely connected layers. The network's input is 2D (150×150), with each channel corresponding to a section in the multicontrast image, and the output is the decision that the patient belongs to a particular grade. The details of the convolutional block are shown in the *lower right*, with each block consisting of convolution, batch normalization (BN), and Rectifier Linear Unit (ReLU) activation, followed by pooling. CONV indicates convolution; FC, fully-connected.



ONLINE FIG 2. Preprocessing. Acquired images were resampled to 1-mm isotropic resolution using BrainSuite,¹³ then, registration was performed on the resampled images using FSL.¹⁴ The registered images were skull-stripped with the BrainSuite toolbox,¹³ then, segmentation was performed using the network proposed by Wang et al¹⁵ to extract the tumor from all the input multicontrast images.