Joshua Unrau[1,2], Martin Lepage[1,2], Katie Lavigne[1,2,3]

[1] McGill University, Montreal, Canada; [2] Douglas Mental Health University Institute, Montreal, Canada; [3] Montreal Neurological Institute-Hospital, Montreal, Canada

CivetQC: An Open-Source Automated Quality Control Tool for MRI Processing with CIVET

BACKGROUND

- The CIVET pipeline automates extraction of cortical surfaces and provides key measures of cortical surface morphometry (e.g., thickness, surface area) from structural magnetic resonance imaging (MRI) data.[1,2]
- CIVET provides users with extensive data for quality control (QC) purposes; however, manually reviewing these outputs can be time consuming and impractical when working with extremely large datasets.
- Previous research has demonstrated that supervised learning algorithms offer a feasible means for automated QC using other processing pipelines, such as FreeSurfer,[3] though these algorithms have yet to be validated with CIVET.
- Here, we present CivetQC, a fully automated QC pipeline for CIVET outputs based on scikit-learn.[4]

METHODS

Sample

- Data from five of our previous studies involving patients with psychotic disorders and healthy controls (N=1216) were processed using the CIVET pipeline.

Manual Quality Control

- We rated the output quality for each subject on a scale from 0 to 2 (0 = fail, 1 = questionable, 2 = pass) based on visual inspection of CIVET outputs.
- Ratings of 1 or 2 were considered acceptable (n = 1163, 95.6%), whereas ratings of less than one were considered unacceptable (n = 53, 4.4%).

Features

- We used a supervised machine learning algorithm to classify quality control outputs as acceptable or unacceptable based on quality metrics, including number of surface-surface intersections, self-intersections, and brain mask error, among others.

Modeling Procedure

1. Divide Data into Training and Test Sets
- Training (75%)
- Testing (25%)

2. Preprocessing
- Standardize Scale of Features (Based on Training Data)
- Balance Training Data: Synthetic Minority Oversampling Technique (SMOTE)[5]

3. Grid Search
- Evaluate Four Model Types: Random Forest Classifier, Gradient Boosting Classifier, K-Neighbors Classifier, and Ridge Classifier
- Metric: Area Under the ROC Curve
- Stratified Five-Fold Cross-Validation
- Hyperparameter Tuning (Bayesian Optimization)

4. Select Final Model (Performance During Cross-Validation)

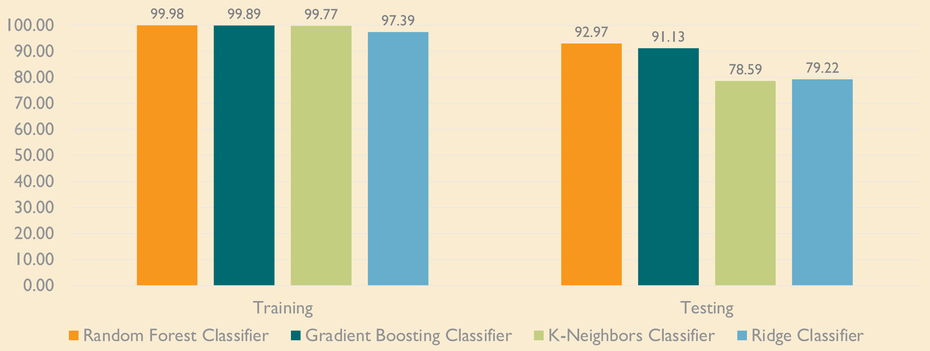
5. Evaluate Performance of Models on Test Set

6. Evaluate Feature Importance in Best Model
- Mean Decrease in Impurity (Random Forest)

RESULTS

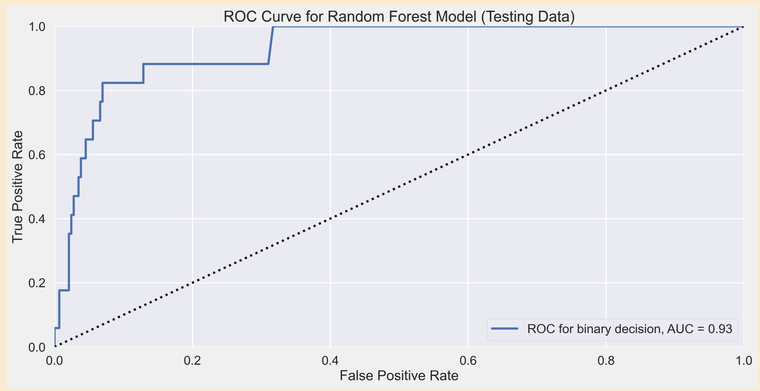
- The training data included 876 acceptable quality scans and 36 unacceptable quality scans, whereas 287 acceptable quality scans and 17 unacceptable quality scans were held back for final model evaluation.
- Following hyperparameter tuning, a Random Forest Classifier was selected as the optimal model based on an average area under the ROC curve of 99.98 during cross-validation.
- This model yielded an area under the ROC curve of 92.97 on holdout data.
- The five most informative features included the mean, minimum, and maximum Laplacian value in gray surface convergence, as well as final residual for gray surface convergence (both left and right).

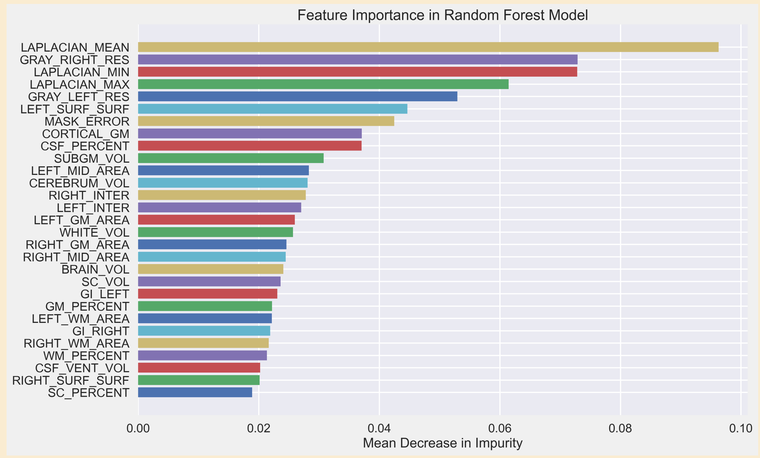
Area Under the Receiver Operating Characteristic Curve (All Models Assessed)



Performance of Random Forest Model (Testing Data)

	N	Precision	Recall	F1-Score
Acceptable	287	1.00	0.99	0.99
Unacceptable	17	0.89	0.94	0.91





CONCLUSION

- These results demonstrate that comparably high recall and precision can be achieved for automated QC of CIVET outputs as has been demonstrated with FreeSurfer.[3]
- Although model performance in the holdout set was generally high (AUC = 92.97) this was notably lower than the near-perfect performance observed during cross-validation (AUC = 99.98), which may indicate potential overfitting during hyperparameter optimization.
- Nevertheless, the model still yielded strong performance in the holdout set.
- Additional model refinement and cross-validation in a larger sample is necessary to ensure optimal performance.
- Further work is required to determine if altering the decision threshold of the model would be of practical value, given that recall of unacceptable scans would be of greater importance than precision, given the limited number of unacceptable scans.
- The source code and documentation for CivetQC is available freely available on GitHub (https://github.com/joshunrau/civetqc).
- CivetQC is also available as a command line tool and can be installed via the Python Package Index (https://pypi.org/project/civetqc/).

References:

1. Ad-Dab'bagh, Y. L., et al. (2006). The CIVET image-processing environment: a fully automated comprehensive pipeline for anatomical neuroimaging research. The 12th Annual Meeting of the Organization for Human Brain Mapping, Florence.
2. Lepage, C., et al. (2017). Human MR evaluation of cortical thickness using CIVET v2.1. The 23rd Annual Meeting of the Organization for Human Brain Mapping, Vancouver.
3. Klapwijk, E. T., et al. (2019). Qoala-T: A supervised-learning tool for quality control of FreeSurfer segmented MRI data. Neuroimage 189: 116-129.
4. Pedregosa, F., et al. (2011). Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research 12: 2825-2830.
5. Chawla, N. V., et al. (2002). SMOTE: Synthetic Minority Over-Sampling Technique. Journal of Artificial Intelligence Research 16: 321-357

Contact:
Email: joshua.unrau@mail.mcgill.ca
Web: joshuaunrau.com