JrnlID 12021_ArtID 9387_Proof# 1 - 14/06/2018

Neuroinformatics
https://doi.org/10.1007/s12021-018-9387-8

ORIGINAL ARTICLE

Using Deep Learning Algorithms to Automatically Identify the Brain MRI Contrast: Implications for Managing Large Databases

Ricardo Pizarro[1,2] · Haz-Edine Assemlal[2] · Dante De Nigris[2] · Colm Elliott[2] · Samson Antel[2] · Douglas Arnold[2] · Amir Shmuel[1]

Abstract

Neuroimaging science has seen a recent explosion in dataset size driving the need to develop database management with efficient processing pipelines. Multi-center neuroimaging databases consistently receive magnetic resonance imaging (MRI) data with unlabeled or incorrectly labeled contrast. There is a need to automatically identify the contrast of MRI scans to save database-managing facilities valuable resources spent by trained technicians required for visual inspection. We developed a deep learning (DL) algorithm with convolution neural network architecture to automatically infer the contrast of MRI scans based on the image intensity of multiple slices. For comparison, we developed a random forest (RF) algorithm to automatically infer the contrast of MRI scans based on acquisition parameters. The DL algorithm was able to automatically identify the MRI contrast of an unseen dataset with <0.2% error rate. The RF algorithm was able to identify the MRI contrast of the same dataset with 1.74% error rate. Our analysis showed that reduced dataset sizes caused the DL algorithm to lose generalizability. Finally, we developed a confidence measure, which made it possible to detect, with 100% specificity, all MRI volumes that were misclassified by the DL algorithm. This confidence measure can be used to alert the user on the need to inspect the small fraction of MRI volumes that are prone to misclassification. Our study introduces a practical solution for automatically identifying the MRI contrast. Furthermore, it demonstrates the powerful combination of convolution neural networks and DL for analyzing large MRI datasets.

Keywords Convolutional neural network · Deep learning · Magnetic resonance imaging · Database management · Automatic contrast identification

Introduction

A recent trend in the neuroimaging community has been to increase dataset size in order to improve the power of studies (Marcus et al. 2013; Weiner et al. 2013; Zuo et al. 2014). Successfully managing large datasets requires multiple servers for storage, software for efficient access and management, and personnel, i.e., system administrators and software developers. There have been great efforts to develop the framework and software to facilitate the setup of a successful neuroimaging database (Cheng et al. 2009).

Large neuroimaging databases have been setup in academic imaging centers and high-tech companies, often for accumulating data from multiple clinical trials. A generic schematic for the hierarchy of the parties involved in acquiring brain magnetic resonance imaging (MRI) data is illustrated in Fig. 1. In the dataset we analyzed, distinct neuroimaging sites acquire brain scans, then typically, a neuroimaging processing company serves as the MRI reading center for the trial and analyzes the data to provide informative results. Clinical trials investigate the efficacy of a drug in one of two ways: (1) make a cross sectional comparison between one patient group taking the drug of investigation and another patient group taking either placebo or the standard of care drug or (2) make a longitudinal comparison within a group of patients, before and after taking the drug. Investigators working in the clinical trials provide specific detailed guidelines including

✉ Ricardo Pizarro
ricardo.pizarro@mcgill.ca

✉ Amir Shmuel
amir.shmuel@mcgill.ca

[1] Montreal Neurological Institute, Departments of Neurology, Neurosurgery, Physiology, and Biomedical Engineering, McGill University, 3801 University, Room 786, Montreal, QC H3A 2B4, Canada

[2] NeuroRx Research, Montreal, QC, Canada

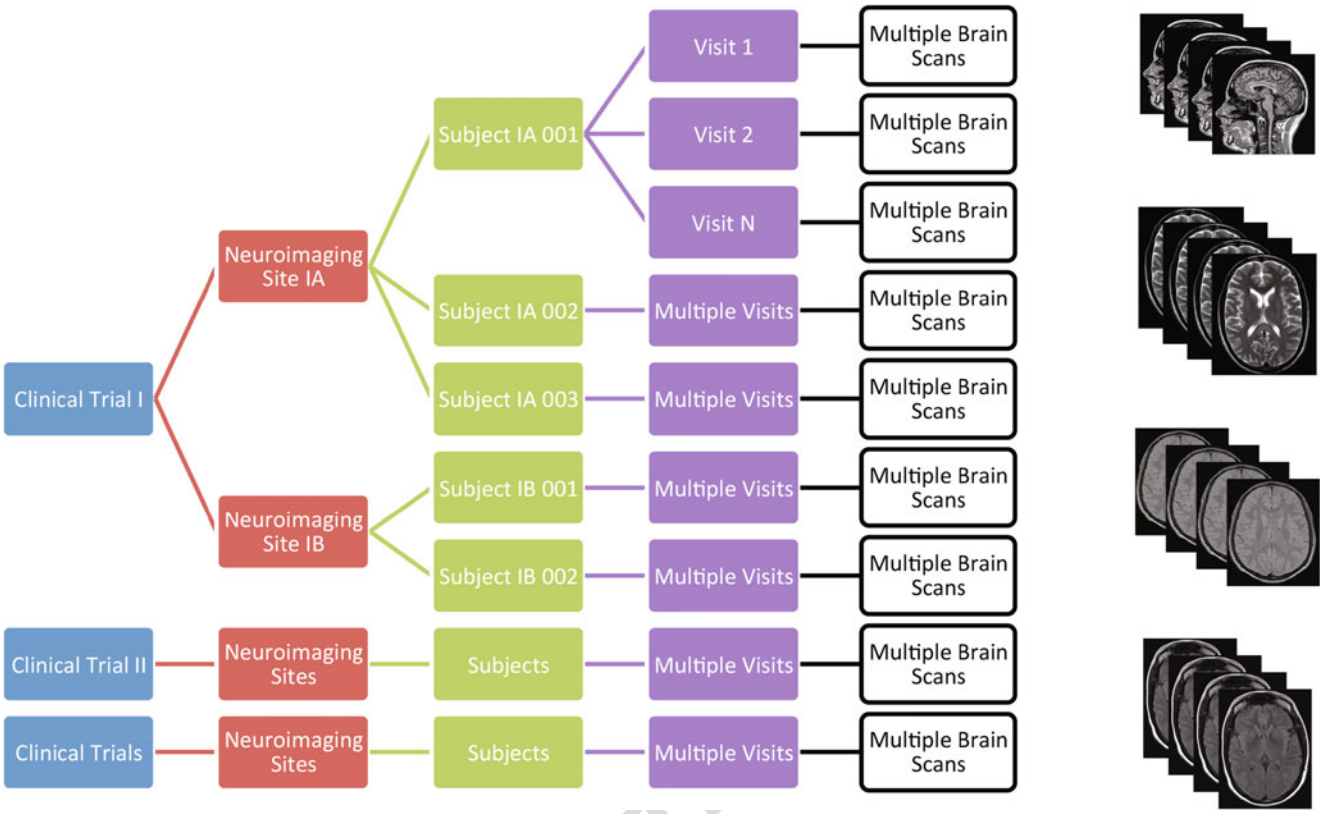

Fig. 1 Schematic diagram to illustrate hierarchy of parties involved in acquiring the neuroimaging dataset we analyzed. For each trial, different neuroimaging sites were hired to scan multiple subjects. Each subject was scanned multiple times over different days. In each visit, a subject was scanned multiple times with different MRI contrasts. The MRI datasets accounting for over 10^5 scans are transferred for systematic and objective analysis

acquisition parameters and instructions for the patients to acquire several different types of brain MRI contrasts acquired in different scans. This helps to minimize variability between datasets and increase precision of the results.

A typical clinical trial is associated with heterogeneity in scanner platforms and data export workflows across several MRI facilities acquiring data for the study. Therefore, the MRI contrast cannot be reliably determined from the names of the corresponding data files or the description of the acquisition included within the file metadata. The MRI technicians may not strictly follow the guidelines or use slightly altered acquisition parameters. In addition, during data conversion, the DICOM header may be corrupted and not have the necessary information. Finally, the scan parameters could not help to distinguish T1-weighted images before and after contrast (i.e. T1P and T1C). Manual entries are required by the MRI technicians to distinguish T1P and T1C, a process that is prone to error. The gold standard for identifying the contrast of a MRI volume is for a trained technician to use visual inspection. The number of MRI scans can reach into the thousands for a single trial, therefore human visual inspection takes up valuable time and resources (Gardner et al. 1995; Pizarro et al. 2016). Thus, the ability to identify the contrast of the scan automatically would represent a significant reduction of time, labor and cost.

Current in-house practice for brain MRI contrast identification is a semi-automated process. The first step is to use a decision-tree (DT) algorithm that exploits the acquisition parameters recorded within the metadata of a MRI volume. In a second step, the MRI volumes undergo interactive quality control, at which time the operator can manually rename MRI volumes, if the contrast has been incorrectly identified by the DT algorithm. This semi-automated process is limited in cases when the metadata does not contain sufficient information to correctly distinguish between similar contrasts e.g., T1-weighted images before and after the injection of gadolinium contrast. Another limitation is the requirement for continual updates of the DT algorithm when new contrasts are introduced, making the algorithm difficult to maintain. Finally, the current approach fails to utilize any of the information contained within the intensity values of the images, relying solely on 1% of the available data.

To our knowledge, no method currently exists to automatically identify the contrast of a MRI scan based on the image contrast. Such a fully automated algorithm can potentially overcome the problems that arise from existing solutions.

Here we present a deep learning (DL) algorithm developed to automatically identify the contrast of a brain MRI volume. We discuss the model architecture for neural networks used sequentially and how the DL algorithm was trained, validated, and tested. The DL algorithm predicted data from an unseen dataset containing five contrasts with 0.15% error rate and another dataset containing eight contrasts with 0.19% error rate. For comparison, we developed a random forest (RF) algorithm to automatically identify the contrast of a brain MRI volume. The RF algorithm predicted data from an unseen dataset containing eight contrasts with 1.74% error rate. We demonstrate how smaller dataset sizes decrease the performance of the DL algorithm. We computed receiver operating characteristic (ROC)-based contrast-specific probability thresholds, developed for the user of the DL algorithm. Finally, we discuss the utility of this algorithm and how it has been implemented in practice. We used the notation throughout the manuscript that boldface symbols represent vectors and matrix size is specified as $k \times l \times m$.

Methods

The database we analyzed, courtesy of NeuroRx Research, was constructed through a processing pipeline to process over 10^5 MRI datasets. The contrasts of the MRI volumes were identified in the beginning of the pipeline with a semi-automated process consisting of a DT algorithm and manual intervention process. We used the results of our semi-automated process as the ground truth for this project. We developed two algorithms to automatically identify the contrast of a MRI volume based on the ground truth. The first method used a RF classifier that considered scan parameters and basic image statistics as input features, while the second method used a DL algorithm based on convolutional neural network. We assessed and compared the performance of the two algorithms in predicting an unseen dataset. We assessed the DL algorithm in more detail to describe its efficiency in predicting the various contrasts and develop an ROC-based contrast-specific probability threshold.

Neuroimaging Dataset

The hierarchy of the parties involved in acquiring the neuroimaging data is illustrated in Fig. 1. In the dataset we analyzed, an overall hundreds of neuroimaging sites were contracted to acquire patient brain scans. Since part of the clinical trials required longitudinal imaging, each subject had up to N timepoints, where N is based on the study aim; a defined set of contrasts was acquired at each timepoint.

A pipeline was developed for efficient processing of the data. The pipeline consisted of multiple sequential phases, meaning successful output of a previous phase was a prerequisite prior to processing the next phase. MRI contrast identification is a critical task performed at the initial phase of the pipeline. We evaluated the DL algorithm performance in two stages. In stage I, we developed the DL algorithm and investigated how the dataset size influenced performance by generating equally balanced datasets of smaller size. We used a balanced dataset with 40,283 MRI scans, containing the five most common contrasts, and referred to, hereon, as the reference dataset. After developing the DL algorithm in stage I, we included additional and less common contrasts in stage II to test whether the algorithm was able to retain high performance when an unbalanced sample was used. In stage II, we used a dataset with 45,785 MRI scans, incorporating three additional contrasts, and referred to, hereon, as the extended dataset.

Target Contrast, the Ground Truth

We previously developed a semi-automated algorithm in house to identify the MRI volume contrast using the following two steps. In step 1, a customized DT algorithm was used on acquisition parameters contained in the metadata of each MRI scan to identify the contrast. In step 2, trained MRI experts visually inspected the dataset slice-wise, as part of an interactive quality control process. If necessary, MRI volumes were manually renamed to reflect the correct contrast. We used the semi-automated process to generate the target contrast, $\mathbf{c}_i$, defined as a binary class vector of size $C \times 1$ where C is the total number of contrasts. The target contrast corresponded to our ground truth, which were used to train, validate, and test the DL and RF algorithms evaluated in this manuscript.

Deep Learning (DL) with Neural Networks

We developed a DL algorithm with neutral networks to infer the contrast of a MRI volume based on image intensity. The DL algorithm consisted of an initial convolutional neural network (CNN), which inferred the contrast on a single slice of the MRI, and a subsequent dense neural network (DNN), which relied on the CNN output to infer a contrast for the entire MRI volume. We made the implementation openly available on GitHub (https://github.com/AS-Lab/Pizarro-et-al-2018-DL-identifies-MRI-contrasts) and developed the algorithm in Python with a Theano backend (Al-Rfou et al. 2016) and compiled on Keras (Chollet 2015). Keras is a high-level software package that provides extensive flexibility to easily design and implement DL algorithms. We manually selected all of the parameters that defined the network architecture, including the number and type of layers, the number of layer nodes, and C, the number of final possible contrasts.

Convolutional Neural Network (CNN) Architecture

The CNN architecture was based on an existing convolution-based neural network, made publicly available in Keras (Chollet 2015). The architecture was previously developed to identify the content on 32 × 32 images from CIFAR10 (Krizhevsky and Hinton 2009). In particular, it consisted of a sequential model written in modular form as illustrated in Fig. 2. The first three modules were convolutional modules with a final downsampling operation. Each convolutional module consisted of the following seven operations: 2D convolution (Krizhevsky et al. 2012), batch normalization (Ioffe and Szegedy 2015), rectified linear unit (ReLU) activation (Dahl et al. 2013), 2D convolution, batch normalization, ReLU activation, and max pooling (Krizhevsky et al. 2012). The final CNN module was essentially a fully connected network (Bengio 2009) that inferred the contrast of the MRI volume. It consisted of reshaping the output of the final convolutional module to a linear array, comprising the following operations: fully connected network (i.e. dense), batch normalization, ReLU activation, fully connected network, and a softmax activation (Dunne and Campbell 1997).

The cited references provide in-depth detail regarding each operation; however, a brief description and motivation for each layer follows. The input layer prepared the data to have size $1 \times 32 \times 32$ with the image processing steps described in 2.5.2. A 2D convolution layer generated the convolution of an image and a kernel of size 3×3. For instance, the first 2D convolution layer estimated 32 kernels of size 3×3. An image of size $1 \times 32 \times 32$ was convolved with each kernel to generate 32 images, making the output of size $32 \times 32 \times 32$. A 2D convolution "viewed" different areas of the image, and as depth increased, the scope widened. Batch normalization, as the name implies, normalized each image by removing the intensity mean and standard deviation. Batch normalization was used to accelerate the training of a deep network, shown to reach optimum parameters in less steps (Ioffe and Szegedy 2015). In a ReLU operation, any intensity value <0 was set to 0, while any value ≥0 was unchanged. A ReLU operation introduced a nonlinear function and efficiently replaced the previously used sigmoid operation. A max pooling operation extracted the element with the highest value within a window of size 2×2, effectively reducing the image size by 2. Max pooling was used to avoid over-fitting and reduce the computational cost. A softmax activation operation transformed the arbitrary values to probability values between [0, 1]. A softmax activation was used to make a final selection output as a contrast probability, $\mathbf{p}_C$, of size $C \times 1$.

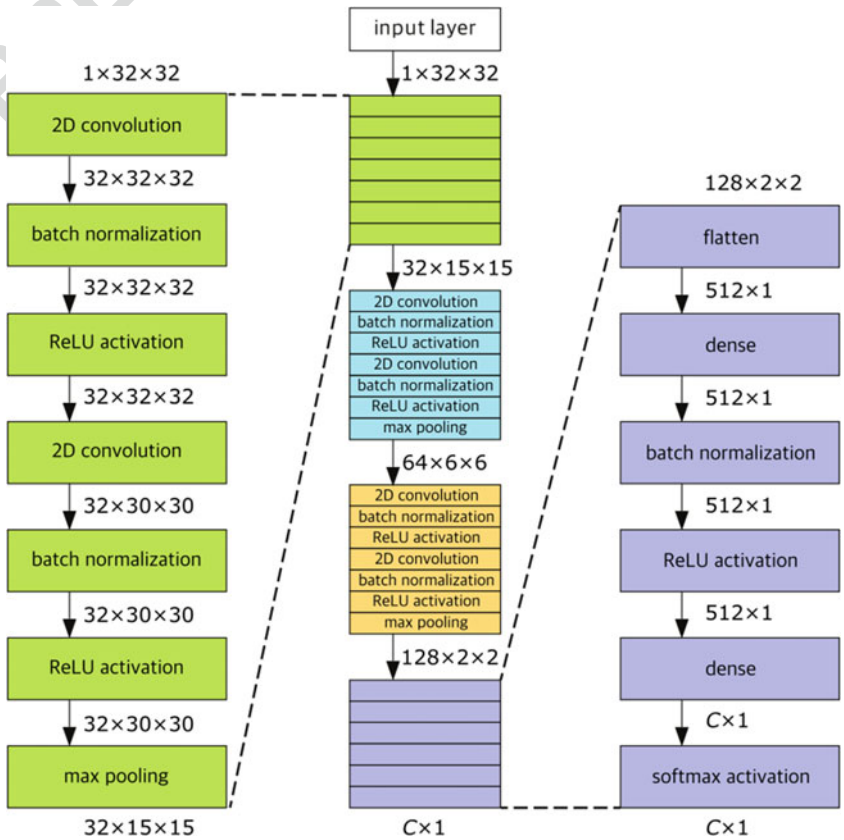

Fig. 2 The convolutional neural network (CNN) architecture was comprised of 27 sequential layers. There were three repeating modules (green, blue, orange) of seven layers. The first module is detailed on the left; rectified linear unit is abbreviated as ReLU. The purple module was a fully connected network that contained the bulk (90%) of the network parameters and inferred the contrast on each slice. The data size is presented above in parentheses before and after each layer. For example, the input was one slice with size $1 \times 32 \times 32$, the input to the fully connected network (in purple) was of size $128 \times 2 \times 2$, and the final output was the inference with size $C \times 1$, where C is the total number of contrasts

Dense Neural Network (DNN) Architecture

The CNN generated inference, $\mathbf{p}_C$, on a per-slice basis, yet we were interested in making an inference on the entire MRI volume. We developed an additional dense neural network (DNN), illustrated in Fig. 3, to compute a volumetric inference, $\mathbf{p}$. First, the input to DNN of size $n \times C$ was generated by computing CNN-generated $\mathbf{p}_C$ on $n = 30$ slices. The DNN architecture was made from similar types of layers as the fully connected portion of the CNN algorithm (purple module in Fig. 2). The DNN output, $\mathbf{p}$, of size $C \times 1$ approximated the probability the MRI volume belonged to one of the possible contrasts, C.

Neural Network Parameters Estimation

The neural network parameters were estimated to minimize the loss, defined in Eq. (2) as the categorical cross-entropy (Murphy 2012). The parameter space was trained using Adam RMSprop with Nesterov momentum (Dozat 2015). The DL algorithm was compiled to run on a Nvidia Quadro K2200. The GPU ran about 20 times faster when compared to a CPU. This increase in speed provided an efficient way to iteratively explore and improve the DL algorithm. The full training process took approximately 20 h to complete.

We used a cross-validation scheme described in 2.5.4 to generate training, validation, and testing subsets. We used the training subset to estimate the neural network parameters. The training subset was loaded in batches, with sizes empirically determined by the limit of the GPU memory. The validation subset was used to estimate performance at the end of each estimation epoch. The algorithm ran for a total of 1000 estimation epochs, where each epoch consisted of 20 estimation steps. During each step, the algorithm used 600 slices, consisting of $n = 30$ slices taken from 20 randomly selected MRI volumes. The estimation procedure used 400 volumes per epoch for 1000 epochs, resulting in the algorithm going through the entire training subset approximately 10 times. This redundancy increased the probability that the algorithm used data from each MRI volume in the training subset at least once. After the estimation procedure completed 1000 epochs, the neural network parameters were saved to predict the data from the testing subset.

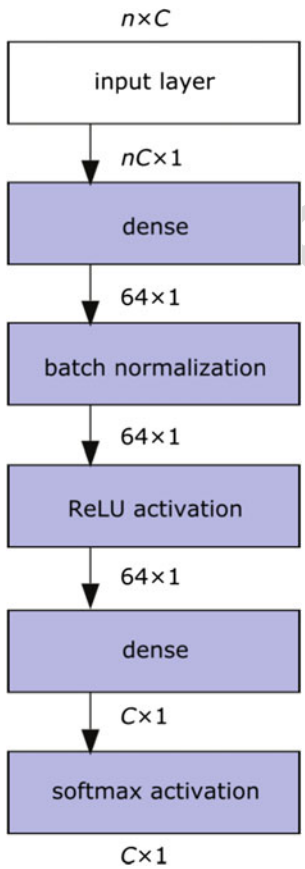

Fig. 3 A dense neural network (DNN) was used to make a volumetric inference from n slices. The DNN was comprised of five sequential layers and made a final inference on the MRI volume; rectified linear is abbreviated as ReLU. The parentheses represent the data size as input and output to each layer

Random Forest (RF) Approach, Developed for Comparison with DL

We developed a RF classifier to characterize the baseline performance for a comparable algorithm that can automate the inference of a MRI contrast. A RF algorithm is a discriminative classifier that consists of an ensemble of DT classifiers, where the final classification is determined by summing the votes cast by each individual tree (Breiman 2001). RFs have been shown to be a powerful automatic classification approach in a wide range of classification tasks. The RF input features used in this work consisted of the acquisition parameters, including the echo time (TE), repetition time (TR), and flip angle, which were extracted from the MRI scan metadata. Additional input features included basic image intensities statistics: percentiles and mean. The complete list of features used for the RF algorithm can be found in section "Feature extraction for the RF algorithm".

Automated Algorithms Evaluation

We evaluated the DL and RF automated algorithms in the following way. First, we defined the metrics used to estimate performance of the DL and RF algorithms at different developmental stages. Second, we processed all the MRI volumes to prepare the slices for input into the CNN. Third, we extracted features for the RF algorithm from the DICOM header and MRI intensity profile. Fourth, we developed a cross-validation scheme to generate uncorrelated subsets and evaluate the two algorithms with unseen data. Fifth, we assessed the algorithms

in two stages, with datasets comprising of five contrasts and eight contrasts, respectively. Sixth, we studied how the dataset size affected performance for five contrasts. Finally, we plotted the distribution of the inferences made to describe how to compute a contrast-specific probability threshold.

Performance Evaluation

We evaluated the DL and RF algorithms at different developmental stages by computing metrics that estimated performance. We computed the error rate, ε, of each algorithm, based on accuracy, to focus our attention on the incorrectly identified MRI contrasts. We estimated performance of an algorithm by measuring ε, defined to be:

$$\epsilon = 1 - accuracy = 1 - \frac{1}{N}\sum_{i=1}^{N} c_i \bullet \mathbf{p}_i \quad (1)$$

where $\mathbf{c}_i$ is the target contrast and $\mathbf{p}_i$ is the algorithm-generated contrast probability. The error rate was estimated by averaging the total number of MRI volumes used over N.

The DL algorithm was trained to minimize the loss defined to be the categorical cross entropy, H. We defined the cross entropy to be a measure of the distance of the algorithm-generated contrast probability from the target contrast. For one MRI volume, i, we defined H_i as follows:

$$H_i(\mathbf{c}_i, \mathbf{p}_i) = -\sum_{j=1}^{C} c_i(j) \log p_i(j) \quad (2)$$

The cross entropy was estimated over all possible modalities, C. For multiple MRI volumes, the categorical cross entropy was averaged over N, as in Eq. (1).

We generated confusion matrices to visualize the DL and RF algorithms' classification performance per contrast. We tracked the number of MRI volumes per contrast where each algorithm-generated prediction agreed or disagreed with the ground truth. The vertical axis of the confusion matrix is the contrast as determined by the DL or RF algorithm, while the horizontal axis of the confusion matrix is the ground truth. The numbers along the diagonal of the confusion matrix reflect the number of MRI volumes when the algorithm-generated prediction and the ground truth agreed. The numbers off of the diagonal of the confusion matrix reflect the number of MRI volumes when the algorithm-generated prediction and the ground truth disagreed.

We characterized the performance of the DL and RF algorithms by computing the sensitivity and specificity to estimate the ability of detecting each contrast. Sensitivity estimates the algorithm's capacity to correctly identify that a MRI volume is a particular contrast while specificity estimates the algorithm's capacity to correctly identify that a MRI volume is *not* a particular contrast. The two metrics were defined as follows:

$$\begin{aligned} sensitivity &= \frac{TP}{TP + FN} \\ specificity &= \frac{TN}{TN + FP} \end{aligned} \quad (3)$$

where TP is the true positive count, FN is the false negative count, TN is the true negative count, and FP is the false positive count.

We developed contrast-specific probability thresholds for the user of the algorithm, which minimized the errors made by the algorithm, reflected by maximizing sensitivity and specificity. A contrast-specific probability threshold, as opposed to taking the maximum value of the probability vector, would increase the confidence in the algorithm's ability of making the prediction. To that end, we computed ROC curves to find the operating point that equally maximizes the algorithm's sensitivity and specificity. We considered any probability value in the range [0, 1] to be a candidate threshold. For each candidate value, we computed the true positive rate (TPR) and false positive rate (FPR), as follows:

$$\begin{aligned} TPR &= sensitivity = \frac{TP}{TP + FN} \\ FPR &= 1 - specificity = \frac{FP}{TN + FP} \end{aligned} \quad (4)$$

We then computed the operating point by weighing TPR and FPR equally and maximizing Youden's index (Youden 1950), defined as:

$$Youden's\ \ index = TPR - FPR \quad (5)$$

Image Processing

We processed all MRI volumes to prepare the slices for input into the CNN. All MRI volumes were masked, down-sampled, and normalized, as follows. (1) We defined a mask to select $n = 30$ slices centered on the central slice of the volume. (2) Each slice was down-sampled to a 32×32 resolution, chosen empirically to provide sufficient data for distinguishing contrasts. Higher resolution slices did not improve results and caused memory issues. (3) Each down-sampled slice was then normalized over the intensity by subtracting the intensity mean and dividing by the intensity standard deviation. The processed images were then recombined and used as input for the DL algorithm. In summary, each MRI volume, i, generated n images with 32×32 resolution, each labeled with $\mathbf{c}_i$ that specified the target contrast.

t1.1 **Table 1** DICOM acquisition parameters used as input features for automatic MRI contrast identification using a Random Forest classifier

Parameter Name	Dicom Field	Units
Repetition Time	0018 × 0023	ms
Echo Time	0018 × 0081	ms
Echo Train Length	0018 × 0091	ms
Inversion Time	0018 × 0082	ms
Slice Spacing	0018 × 0088	mm
Percent Sampling	0018 × 0093	Percent of acquisition matrix lines acquired
Percent Phase Field of View	0018 × 0094	Percent
Pixel Bandwidth	0018 × 0095	Hz
Flip Angle	0018 × 1314	Degrees
SAR	0018 × 1316	Watts per kilogram
Contrast Media	0018 × 0010	Name of contrast agent, if present[a,d]
Sequence Variant	0018 × 0021	Name of Sequence Variant[b,d]
Scan Options	0018 × 0022	Name of Scan Options[c,d]

[a] Encoded as 0 if no contrast agent present, 1 if any contrast agent present

[b] Encoded as magnetization transfer (MT) = 1, inversion recovery (IR) = 2, SAT (saturation band) = 3, VB (variable bandwidth) = 4, other/empty = 0

[c] Encoded as MTC\crSP = 1, SK\crSP\crMP\crOSP = 2, SK\crSP\crOSP = 3, SS\crSP = 4, SK\crSP = 5, SP = 6, SK = 7, other/empty = 0, where MTC = magnetization transfer contrast, SP = spatial presaturation, MP = magnetization prepared, OSP = oversampling phase, SK = segmented k-space, SS = steady state

[d] Manually entered by operator

Feature Extraction for the RF Algorithm

We extracted features for the purpose of automatic contrast identification using a RF algorithm. Table 1 summarizes the acquisition parameters extracted from the MRI DICOM header file. Non-numerical parameters were mapped to integer values prior to being used as an input feature.

In addition to the acquisition parameters described in Table 1, we extracted percentile intensities, as follows. We linearly normalized each MRI volume to the range 0–100, then we included the 70th, 80th, 90th, 99th, and 99.5th percentile intensities for each volume as features for input to the RF classifier. Lower percentile intensities were not considered, as in general approximately 66.7% of the MRI volume is background. These features were included as a coarse representation of intensity histogram shape, primarily in an effort to help distinguish T1-weighted volumes with and without a contrast agent, i.e., T1C and T1P in Table 2.

Cross-Validation Scheme

We developed a cross-validation scheme to divide the dataset into uncorrelated subsets: training, validation, and testing (Ripley 2007). The dataset contained MRI scans that were highly correlated. Subjects were typically scanned in the same site and scanner, causing the MRI volumes in a trial to be correlated not only in terms of underlying anatomical structures, but also in terms of the image formation model. There was a need to provide a training subset which effectively characterized the variability of the scans. Our cross-validation scheme consisted of using a training subset with scans from all clinical trials, which proved to be the key to getting generalizable results. To that end, we incorporated the following two steps into splitting the data. First, we used a single timepoint for each subject, even if multiple timepoints were acquired. The first step ensured the subjects did not repeat, thus reducing correlation across subsets. Second, we constructed the training, validation, and testing subsets, with randomly selected MRI volumes from all clinical trials and imaging centers. The second step ensured we characterized scanner variability.

The training subset corresponded to 60% of the MRI volumes from the dataset and was used to estimate the DL and RF algorithms' parameters. The validation subset corresponded to 20% of the MRI volumes from the dataset and was used exclusively to track the DL algorithm performance after each estimation epoch completed, without feedback to the training. The testing subset corresponded to 20% of the MRI volumes from the dataset and was used to assess performance of the algorithms after training completed.

We used the cross-validation scheme described above to divide the reference dataset containing five contrasts used in stage I. The stage I testing subset was used to generate the confusion matrix in Fig. 4. The stage I training and validation subsets were used to track the performance by training epoch

Q1 t2.1 **Table 2** Contrast distribution of MRI volumes used for cross-validation is shown below

Contrast	Abbreviation	Cross-validation subsets		
		Training	Validation	Testing
fluid-attenuated inversion recovery	FLR	4897	1615	1616
proton-density weighted	PDW	4854	1606	1606
T1-weighted post-contrast	T1C	4800	1590	1582
T1-weighted pre-contrast	T1P	4825	1593	1593
T2-weighted	T2W	4880	1612	1615
Reference dataset (stage I)		**24,256**	**8016**	**8011**
high-resolution T1-weighted	T1G	545	181	170
magnetic transfer ON	MTON	1399	446	450
magnetic transfer OFF	MTOFF	1408	449	453
Extended dataset (stage II)		**27,608**	**9092**	**9085**

The contrast abbreviation and number of MRI volumes in cross-validation subsets for training, validation and testing

plotted in Fig. 5. We repeated the cross-validation scheme described above to divide the extended dataset used in stage II containing eight contrasts. The stage II testing subset was used to generate the confusion matrix in Fig. 6.

Stage I – Five Contrasts

In stage I, we used the reference dataset to develop the DL algorithm and assessed how the size of the dataset affected performance. As shown in Table 2, we divided the reference dataset as follows: 24,256 in the training subset, 8016 in the validation subset, and 8011 in the testing subset. The MRI scans were labeled to be one of $C = 5$ contrasts,whose name we abbreviated as follows: fluid-attenuated inversion recovery (FLR), proton-density weighted (PDW), T1-weighted post-

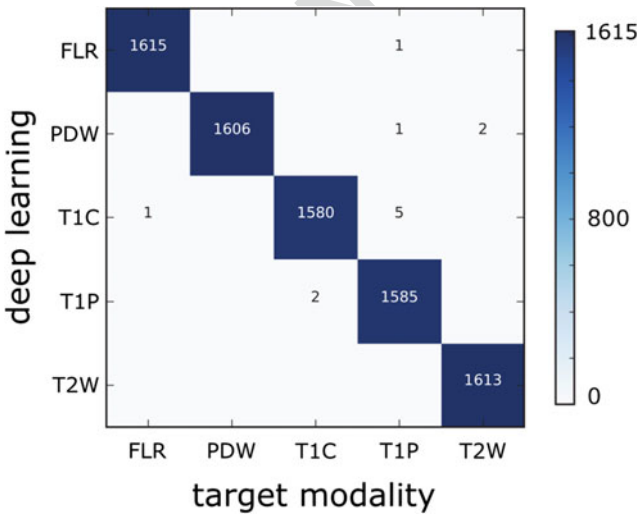

Fig. 4 The confusion matrix is illustrated above for five contrasts. The target contrast was taken from the ground truth identification. The inferred contrast was determined using the proposed deep learning algorithm

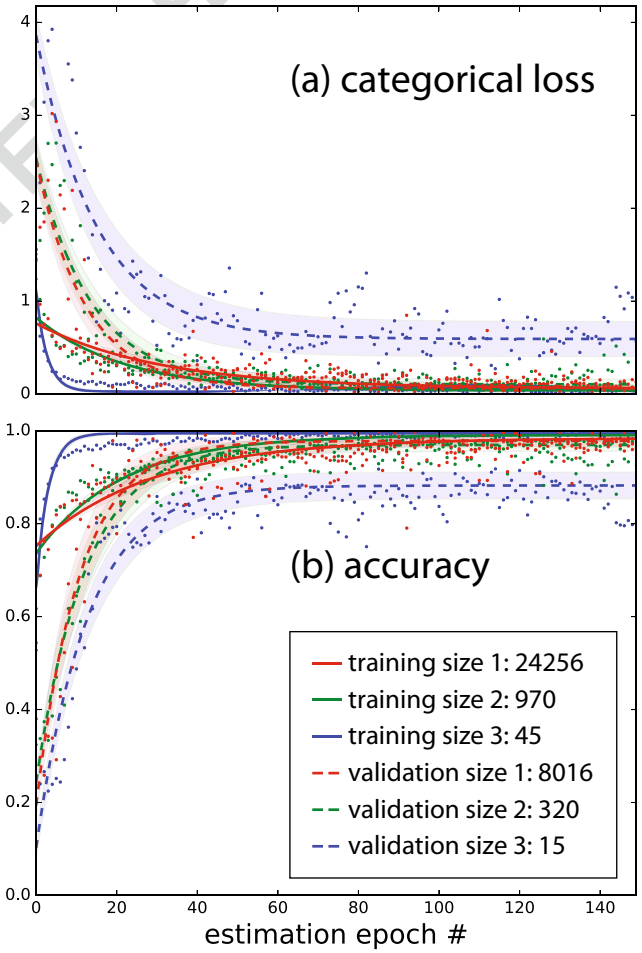

Fig. 5 (**a**) Categorical loss and (**b**) accuracy for the deep learning as a function of estimation epoch for three dataset sizes. The solid line is training accuracy and the dashed line is validation accuracy. The shaded region is an estimate for the standard error (SEM). The legend specifies the size of the training set and validation set. We repeated size 1 five times. We tracked 13 randomly selected parts of size 2 and 25 randomly selected parts of size 3

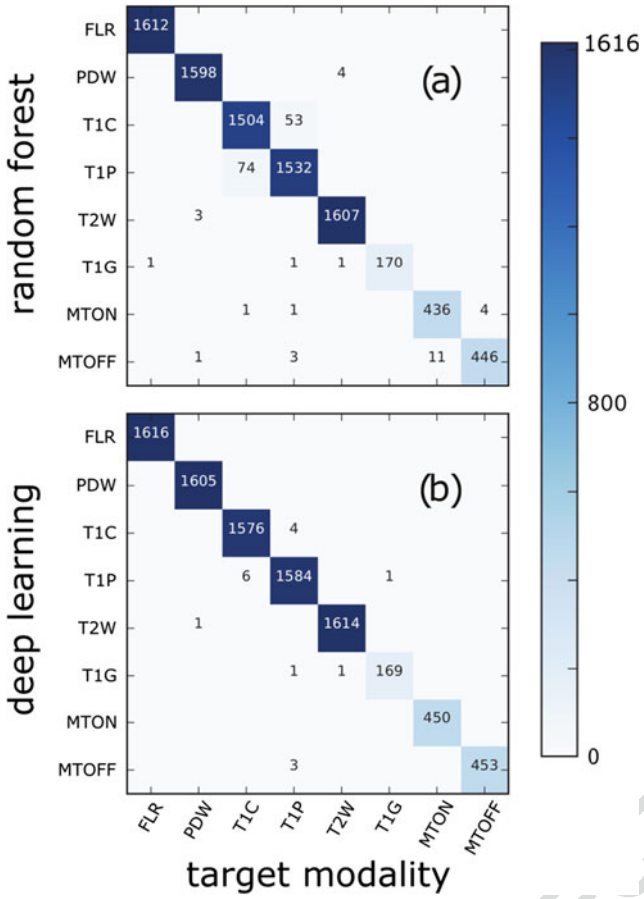

Fig. 6 Confusion matrices are illustrated for eight contrasts generated using (**a**) the random forest algorithm and (**b**) the deep learning algorithm. The target contrast was defined as the ground truth identification. The inferred contrast was determined using the two different algorithms

contrast agent (T1C), T1-weighted pre-contrast agent (T1P), and T2-weighted (T2W). The DL algorithm was developed using the reference dataset and results were summarized for the testing subset as a confusion matrix in Fig. 4.

We investigated how the size of the dataset affected the performance of the algorithm, which allowed us to estimate the number of samples needed to maintain performance. In other words, how much data would we need to replicate an algorithm performing with similar accuracy? We tracked performance as the value of the accuracy, 1-ε, and loss function, H, for both the training and validation subsets after each epoch completed. We explored three sizes: 40,283 MRI scans, 1610 MRI scans, and 75 MRI scans. First, we estimated the network parameters for the reference dataset of size 40,283 MRI scans to compute the performance mean and variance by epoch. We repeated the estimation procedure for the reference dataset a total of five runs. Each run involved a random partition of the reference dataset to the training, validation, and testing subsets. Second, we divided the reference dataset into equal parts, each consisting of 1610 MRI scans. Each part was then divided into three subsets: 970 in training subset, 320 in validation subset, and 320 in testing subset. We tracked the performance by estimation epoch in 13 randomly selected parts. Using only 13 parts proved sufficient to estimate performance based on the statistical stationarity of the mean and standard error of the mean (SEM) of the accuracy and loss function. Third, we divided the reference dataset into equal parts, each consisting of 75 MRI scans. Each part was then divided into three subsets: 45 in training subset, 15 in validation subset, and 15 in testing subset. We tracked the performance by estimation epoch in 25 randomly selected parts. Using only 25 parts proved sufficient to estimate performance based on the statistical stationarity of the mean and SEM of the accuracy and loss function. To summarize the performance, we computed the arithmetic mean over the iterations for each size. We then fit a decaying exponential function to the mean and SEM with parameters that minimized the error between fit and true value. The resulting performance by epoch for all three sizes is plotted in Fig. 5.

Stage II – Eight Contrasts

In stage II we compared the performance between the DL algorithm and the RF algorithm. We then explored the ability for the DL algorithm to work as an identifying tool for the different contrasts and generated ROC curves to develop a contrast-specific probability threshold. We used the extended dataset which comprised of 45,785 MRI scans, which we divided into three subsets: 27,608 in training subset, 9092 in validation subset, and 9085 in testing subset. The MRI scans were labeled to be one of $C = 8$ contrasts, including the five stage I contrasts and the three additional contrasts: high-resolution T1-weighted (T1G), magnetic transfer ON (MTON), and magnetic transfer OFF (MTOFF). The contrast acronym and number of MRI volumes in each of the three cross-validation subsets are summarized in Table 2.

Results

We developed a DL algorithm to automatically identify the contrast of a brain MRI. We designed the DL architecture to make inferences on unseen MRI data. We generated a confusion matrix to summarize the results obtained in stage I for five contrasts. We plotted training and validation performance metrics as a function of estimation epoch for three dataset sizes from stage I. For comparison to the DL algorithm, we designed a RF algorithm to make inferences based on features extracted from the MRI metadata and image statistics. In stage II, we generated confusion matrices to summarize the results obtained for eight contrasts for the DL and RF algorithms. We characterized the DL and RF algorithms' capacity as an identification tool for each contrast. Finally, we developed a method to maximize performance by selecting a contrast-specific probability threshold accessible to the users of the algorithm.

Stage I – Five Contrasts

In stage I, we developed the DL algorithm for the five MRI contrasts that were acquired most frequently in the dataset we analyzed. First, we evaluated the relationship between slice orientation and resolution. Most volumes in the database have approximately 60 axial slices. Therefore, testing the performance with high resolution images was limited to the axial orientation. We compared the following three cases using the proposed algorithm for five contrasts (Table 3):

In all cases, we selected the slices centered around the corresponding midline. Using the high-resolution images with axial orientation did not improve the results. Therefore, we continued developing and exploring sagittal orientation and 32×32 dimensions. It was encouraging that the algorithm performed with low error rate; however, we were surprised the higher resolution image did not improve the results.

We summarized the classification results from stage I with a confusion matrix detailing the number of MRI volumes where the DL algorithm and the ground truth agreed and disagreed. Fig. 4 illustrates the confusion matrix for five contrasts generated with the DL algorithm on the testing subset. The DL algorithm and the ground truth identification inferred the same contrast on nearly all MRI volumes quantified along the diagonal. There were some volumes where the DL algorithm and the ground truth identification did not agree, as quantified by the numbers off of the diagonal. The error rate of the DL algorithm for five contrasts was $\varepsilon = 0.15\%$. This result raised the question of when the DL algorithm breaks down due to smaller dataset sizes and including additional contrasts, as we did in stage II.

Next, we investigated how the size of the dataset affected the performance of the algorithm in order to estimate the dataset size required to preserve performance obtained when using the reference dataset. We tracked accuracy and categorical loss of the training and validation subsets of three different sizes after each estimation epoch. Fig. 5 presents the accuracy and categorical loss as a function of estimation epoch for the training and validation subsets plotted for different dataset sizes. We plotted performance for estimation epochs [0, 150] as the trends remained constant above 150 epochs. The mean value is plotted as dots by estimation epochs and the fits are plotted for the training and validation subsets in solid and dashed lines, respectively. The performance generated from the data of size 1, in red, did not significantly differ from the performance generated from the data of size 2, in green. The DL algorithm reached peak performance in less estimation epochs when the dataset size was reduced to size 3, as reflected by the solid blue line compared to the solid green and red lines. However, the DL algorithm was not able to generalize as well when the dataset size was reduced to size 3, as reflected by the dashed blue line compared to the dashed green and red lines.

Table 3 Orientation and dimension explored with the DL algorithm

Orientation	# Slices per volume	Dimension	Error rate (%)
sagittal	30	32×32	0.15
axial	30	32×32	0.35
axial	10*	128×128	0.49

*- 10 slices were used because of memory issues

Table 4 The capacity of the DL and RF algorithms to detect each contrast, as summarized by measures of sensitivity and specificity. Sensitivity estimates the capacity of the algorithm to correctly identify that a MRI volume is a particular contrast and specificity estimates its capacity to correctly identify that a MRI volume is not the particular contrast

Contrast	deep learning (DL)		random forest (RF)	
	Sensitivity	Specificity	Sensitivity	Specificity
FLR	100.00%	100.00%	99.94%	100.00%
PDW	99.94%	100.00%	99.75%	99.95%
T1C	99.62%	99.95%	95.25%	99.29%
T1P	99.50%	99.91%	96.35%	99.01%
T2W	99.94%	99.99%	99.69%	99.96%
T1G	99.41%	99.98%	100.00%	99.97%
MTON	100.00%	100.00%	97.54%	99.93%
MTOFF	100.00%	99.97%	99.11%	99.83%

Stage II – Eight Contrasts

In stage II, we extended the application of the DL algorithm to eight MRI contrasts, and compared the DL algorithm performance to that of the RF algorithm. We summarized the classification results with a confusion matrix detailing the number of MRI volumes where the two algorithms and the ground truth agreed and disagreed. Fig. 6 illustrates the confusion matrices for eight contrasts generated with the RF and DL algorithms on the testing subset. The metadata was not readable for 21 MRI volumes, resulting in 9066 total number of volumes tested with RF instead of the possible 9087. The RF algorithm misclassified several MRI contrasts that were correctly classified by the DL algorithm. The DL algorithm outperformed the RF algorithm across all contrasts, except for T1G. The error rate of the DL algorithm for eight contrasts was $\varepsilon = 0.19\%$. The error rate of the RF algorithm for eight contrasts was $\varepsilon = 1.74\%$. A lower error rate generated by the DL algorithm indicates that there is relevant information in the image intensity that is not captured by the features used in the RF algorithm.

We characterized the performance of the DL and RF algorithms as a contrast-identifying tool to describe the results generated from the testing subset obtained in stage II. We computed the sensitivity and specificity describing the

capacity for the DL algorithm to identify whether a MRI volume is or is not a particular contrast. We computed sensitivity and specificity based on each contrast and summarized the two metrics in Table 4. The resultant DL values were all >99.41% and there were multiple contrasts with 100.00% sensitivity and specificity. These metrics break down the accuracy by highlighting that the DL algorithm could improve overall accuracy by improving the sensitivity in detecting T1 contrasts: T1P, T1C, and T1G. The DL algorithm outperformed the RF algorithm across all contrasts except for the sensitivity generated from the T1G contrast. This indicates that the RF algorithm is using a feature extracted from the DICOM header or image intensity profile that is helping its classification of the T1G contrast.

We visually inspected the MRI volumes that were misclassified by the DL algorithm to better understand the reason for misclassification. Our visual inspection identified three groups of misclassification.

(i) The following list identified six cases where the ground truth identification process failed and the DL algorithm succeeded in correctly inferring the contrast:

- Two T1C volumes were mislabeled as T1P and correctly identified by the DL algorithm.
- Three mtOFF volumes were provided by the clinic as T1P. These cases were actually mtOFF and identified as such by the DL algorithm.
- One T1C volume did not have sufficient contrast failing the quality control process. The DL algorithm identified the volume as T1P.

(ii) The DL algorithm wrongly inferred the contrast in the following seven cases, as a result of an acquisition error:

- In two cases, T1P volumes confused by the algorithm as T1C because they contained high-intensity voxels at the extreme right edge of the image due to a ghosting artifact.
- In three cases, the delay between gadolinium injection and acquisition of the T1C volume was too short or too long, resulting in minimal gadolinium enhancement, thereby confusing the algorithm to infer the T1C volumes as T1P volumes
- In one case, the wrong parameter for TR was used, causing the PDW to appear similar to a T2W volume.
- In one case, a wrap-around artifact at the top of the head caused the algorithm to infer a T1G volume as a T1P volume.

(iii) There were three cases where the DL algorithm failed to infer the correct contrast, without a clear explanation:

- One T1P was wrongly inferred as a T1G.
- One T1C was wrongly inferred as a T1P.
- One T1C showed the effect of the contrast agent, but the effect was not as bright as in the regular case. This MRI volume was wrongly inferred as a T1P.

In order to develop a contrast-specific probability threshold, we computed the DL-generated probability for each MRI volume in the testing subset to belong to a specific contrast, c_s. Fig. 7 presents the distributions of the probabilities in the corresponding subplot for each c_s. Let c_t be the target contrast of a MRI volume. When $c_s = c_t$ the distribution of the probabilities was identified in green; conversely, when $c_s \neq c_t$ the distribution of the probabilities was identified in red. This presentation indicates a successful identification if data points in green are close to 1 and data points in red are close to 0. The DL algorithm was designed to generate a probability vector $C \times 1$ with high probability for c_t, and low probability for all other entries of the probability vector. The corresponding global distribution with the range [0, 9000] is shown in the inset of each subplot where only the MRI volumes whose probabilities were 0.0 or 1.0 can be clearly seen. The zoomed perspective distribution with the range [0, 20] is shown in the subplot, where the MRI volumes whose probability was anywhere between [0, 1] can be seen as well. The distribution in Fig. 7 is reflective of the results in the confusion matrix in Fig. 6(b). The algorithm selected the contrast whose value was the highest across the generated probability vector. It can be seen that the specified contrasts that resulted in more errors in Fig. 6(b), such as T1P and T1C, generated probability distributions with higher entropy in Fig. 7. Conversely, the contrasts with fewer errors in Fig. 6(b), generated a nearly binary probabilities distribution in Fig. 7.

Next, we developed a contrast-specific probability threshold for the DL algorithm to minimize the errors reflected by maximizing sensitivity and specificity. For each c_s and for each candidate threshold, we computed TPR and FPR. We generated ROC curves by plotting TPR versus FPR in Fig. 8 for each c_s. The inset shows a global perspective of the ROC curve with the computed operating point as a red circle in the upper left corner. For all contrasts, the red circle is proximal to the ideal operating point located in the upper left corner. In the magnified corner of the ROC curve, we included the ideal operating point with a green star and the operating point with a red circle. The red lines oriented at 45° reflect that we computed the operating point by weighing TPR and FPR equally to maximize Youden's index (Youden 1950). Compared to other contrasts, the T1G and T1P contrasts were associated with a larger gap between the green star and the red circle.

In addition, we tracked each candidate threshold whose Youden's index exceeded 0.98 to characterize how the threshold influences the performance. We plotted Youden's index as a function of candidate threshold in Fig. 9. By comparing to the results in Fig. 7, it is visually possible to identify how the

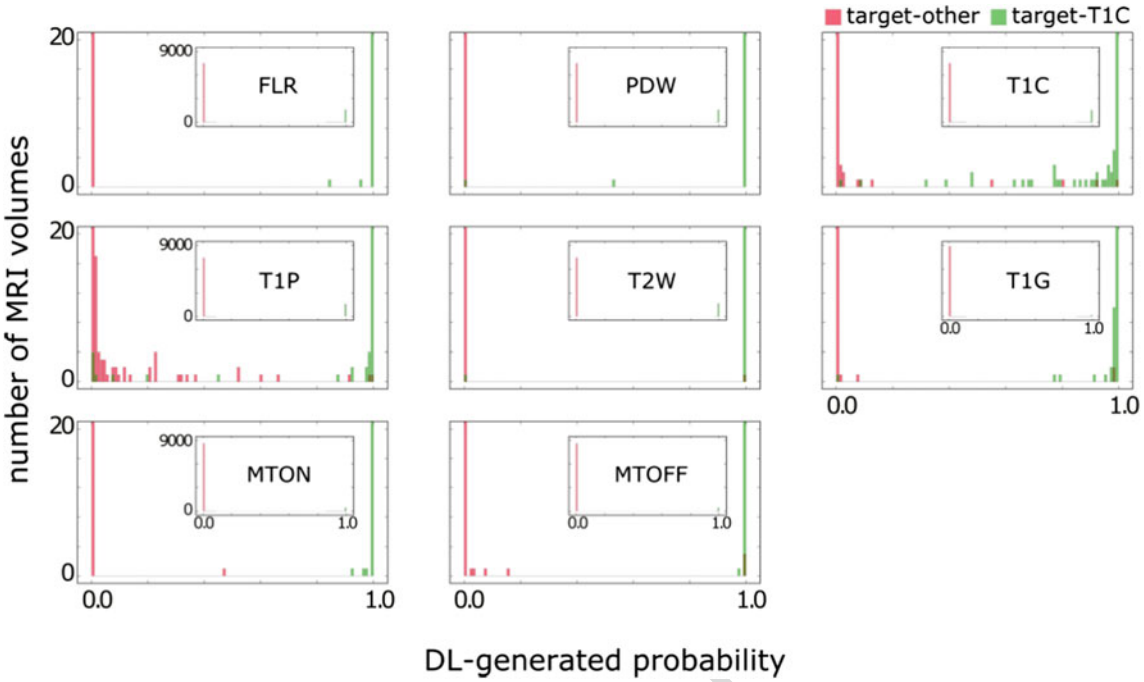

Fig. 7 Distribution for the deep learning (DL) generated probabilities by target MRI contrast. The subtitle specifies the target contrast. The inset is the same plot with the range from [0, 9000] to provide a global perspective. The green bars reflect the probability for the MRI volumes targeted by the specified contrast, and the red bars reflect the probability when the target was not the specified contrast

DL-generated probability distribution determined Youden's index. The plots in Fig. 9 are equivalent to Fig. 8, but they illustrate that the threshold corresponding to the final operating point was selected as the highest threshold to maximize Youden's index. In the general case, there are multiple thresholds that equally maximize Youden's index. For some contrasts, a particular threshold is critical to maximize Youden's index. However, for other contrasts there exists a range of candidate thresholds that provide identical results. We selected the highest threshold in this range because it is the most conservative threshold, thus providing a safety zone that minimizes classification errors. Importantly, the algorithm classifies each MRI volume, and outputs not only the selected class but also the estimated probability that the classification is correct. For probabilities lower than the contrast-specific probability threshold, the user can inspect the volume in question, and make the final decision.

Discussion

An overview of the results is presented in this paragraph to frame the discussion. We developed a DL algorithm to automatically identify the contrast of a MRI volume. For five contrasts, the DL algorithm identified the contrast in unseen testing data with $\varepsilon = 0.15\%$. The algorithm converged to optimum parameters in fewer estimation epochs when training on smaller subsets. However, the DL algorithm did not generalize when the size was reduced to size 3, reflected by a decrease in performance on the validation subset. For eight contrasts, the RF algorithm identified the contrast from the testing subset with $\varepsilon = 1.74\%$ and the DL algorithm with $\varepsilon = 0.19\%$. We characterized the DL algorithm's capacity to detect each contrast with sensitivity that ranged between [99.41%, 100.00%] and specificity that ranged between [99.91%, 100.00%]. We characterized the RF algorithm's capacity to detect each contrast with sensitivity that ranged between [95.25%, 100.00%] and specificity that ranged between [99.01%, 100.00%]. A contrast-specific probability threshold was computed for the DL algorithm with ROC analysis to indicate the user when to double-check particular contrasts.

We modified an existing algorithm to develop a new neural network to perform DL and this application has proven useful. The algorithm has been successfully implemented into the processing pipeline. The tool we have developed can save database management teams valuable resources, including hours spent by technicians doing the trivial task of identifying the contrast of the MRI volume. In addition, this tool provides the methods for inputting MRI into DL for new applications.

A convolutional neural network is analogous in function to the visual system. The first few layers extract low level features such as edges and background. As the layers get deeper, the low level features get combined to increase abstraction. We traced the output of all the layers of the CNN when using different MRI contrasts as input to reveal that:

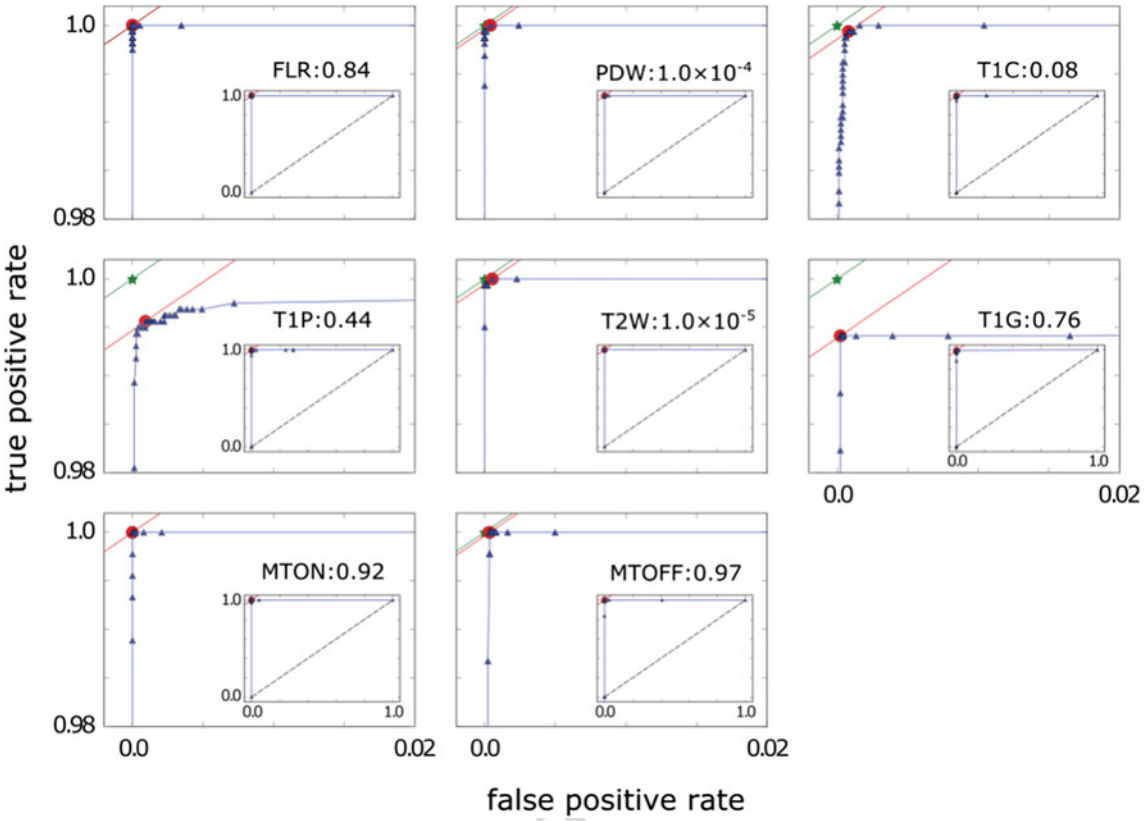

Fig. 8 Receiver operating characteristic (ROC) curves are plotted in blue as true positive rate (TPR) versus false positive rate (FPR) for each contrast. We computed (FPR, TPR) points for each threshold based on the DL-generated probabilities distribution in Fig. 7. The green star indicates the ideal point (0, 1) with the corresponding TPR and FPR equality line. The red point indicates the Youden's index, defined in Eq. (5), with the red line corresponding to TPR and FPR equality. The corresponding threshold is included in the subtitle with the contrast name. The inset provides the global perspective with the range [0.0, 1.0] and the black-dashed line represents the random chance of correctly selecting the contrast, generated with a naïve contrast identifier

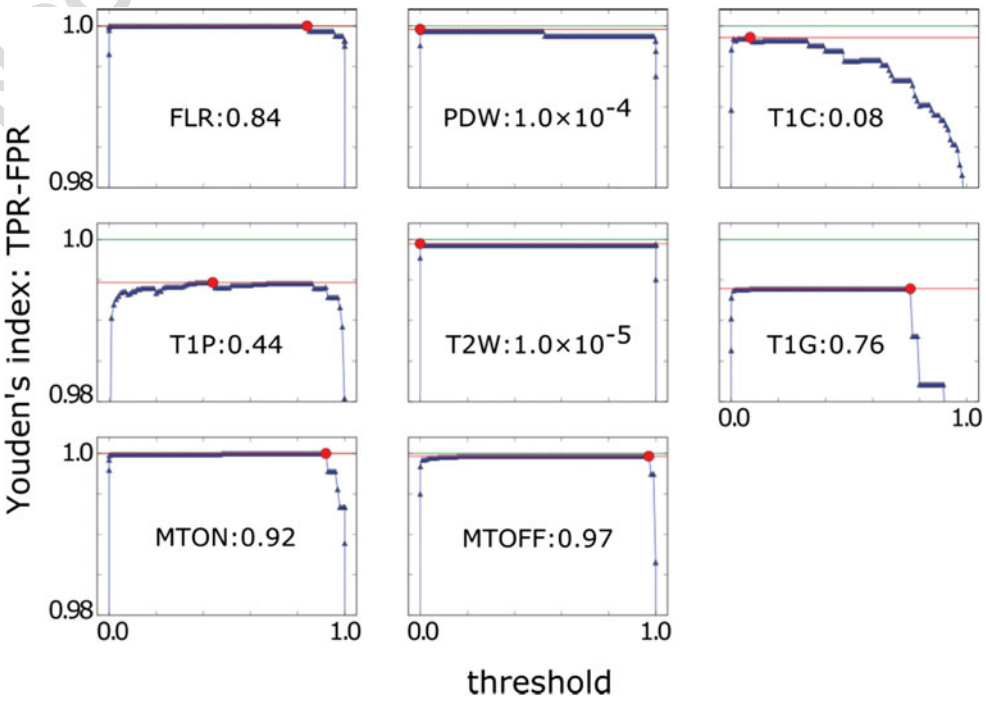

Fig. 9 Youden's index is plotted for the candidate thresholds. The red point indicates the selected threshold corresponding to the operating point. The green line indicates the optimal operating point with no errors. Note: the red point was selected to be the maximum threshold to preserve the maximum value for Youden's index. The T2W contrast curve subtly increases at the selected threshold 1.0×10^{-5}

- When introducing a T1C image into the CNN, the different layers show that the bright spots are emphasized and highlighted, effectively behaving as a vessel dilator.
- Much to our surprise, a 32 × 32 sample of the entire image was sufficient to generate such high performance. A higher density image was not necessary to reach high accuracy levels.

The flexible nature of the neural network architecture of the DL algorithm allows for any alterations that may be necessary for future studies. Within contrast identification problems, the developer can change the number of contrasts by editing the number of possible outputs, *C*. The interested investigator could then either: (1) estimate the network parameters for the entire architecture of the DL algorithm from the beginning or (2) use the stored parameters and only estimate the parameters for the part of the architecture that pertains to the new number of contrasts, i.e., the last two layers of the CNN. In addition, the investigator could explore the CNN architecture depths by adding or removing convolutional modules. Finally, it is possible to restrict the choices of outputs to those that are relevant for a given clinical trial. One possible way is to incorporate a bias before the final layer of the CNN to restrict specific choices that are not indicated within a study.

Our performance versus dataset size analysis, summarized in Fig. 5, brought forth two concepts regarding data sampling. First, reducing the dataset to size 2 did not significantly affect the performance of the algorithm, as illustrated by the red and green curves. The DL algorithm did not generalize across imaging sites during our initial attempts. The key to getting robust generalizable results stemmed from taking representative samples from each site when we implemented cross-validation. Second, reducing the dataset to size 3 significantly reduced the performance of the DL algorithm, as illustrated in Fig. 5 by the red and blue curves. A dataset size of 45 samples was unable to represent the reference dataset, resulting in worse performance metrics. This result emphasized the point that more data improves the power of the study and that a minimal number of samples is required to represent the reference dataset such that performance is not compromised.

We computed a contrast-specific probability threshold with ROC curves from the generated probability distribution. As illustrated in Fig. 9, six of the eight contrasts produced a range of values that generated identical maximum values for Youden's index. The wider this range is, the more likely the algorithm makes correct classification decisions. We selected the most conservative value. In other words, although there is a range of thresholds that give the same Youden's index, we selected the highest threshold in the range. This translates to a 'safety zone', assuring that whenever the reported probability for correct classification is higher than the threshold, the possibility of an error is reduced.

It should be stated that the DL algorithm is orders of magnitude slower to implement than the RF algorithm. The parameters for the DL algorithm were estimated on a GPU. The DL estimation procedure ran for 1000 epochs lasting approximately 6 h. However, the GPU ran about 20 times faster than the CPU; therefore, a similar estimation run on a CPU would take approximately 120 h. After training, the DL algorithm was used for testing, a process that lasted approximately 78 min on the GPU, equivalent to 24 h on a CPU. In comparison, the RF algorithm ran for 6 min during the parameter estimation process and 3 min during the testing phase, both realized on a CPU.

One possible improvement to the DL algorithm is to include the acquisition parameters, which we used as features for the RF algorithm, as additional input to some layer deeper than the convolution process. We demonstrated that RF was able to classify all T1G volumes with 100% accuracy. However, as a result of the feedback from users regarding the single T1G volume misclassified by the DL, we attributed the misclassification to technical acquisition errors. Acquisition parameters could improve classification for one contrast, but it may confuse the DL algorithm and cause additional error, as in our experience with the RF algorithm. In our experience, relying on the header file is problematic for the following reasons: (i) the header file is corrupted or missing altogether, (ii) the operator manually enters parameters incorrectly, and (iii) inconsistent parameters across different neuroimaging sites. We work around these problems by exploiting the advantage of the CNN approach in that it doesn't rely on any headers, thus reducing the errors that arise from manual intervention.

The ground truth identification process failed in the six cases when MRI technicians made mistakes. Human work is prone to error which is one motivation behind developing an automated algorithm. However, using supervised learning to develop a deep learning algorithm requires "ground truth" data. It is the goal of an automated deep learning algorithm to learn and supersede the current process. The six cases highlight the value in using a DL algorithm to avoid mistakes made by technicians. Note that the ground truth identification process does not suffer a systematic bias, since there were multiple annotators, and a scan can be assigned to any given annotator. Thus, the noisy labels are uncorrelated and have

Table 5 Comparison of error rate results generated from deep learning by using with 2D and 3D convolution

Algorithm	Error rate (%)	
	stage I – five contrasts	stage II – eight contrasts
deep learning 2D	0.15	0.19
deep learning 3D	0.49	0.87

little effect on the overall analysis, considering the size of the dataset.

Another potential improvement could be to implement 3D convolution rather than 2D convolution. We edited the architecture of the DL algorithm to incorporate 3D convolution instead of 2D convolution. We tested the trained algorithm in stages I and II and summarize the results here (Table 5):

Incorporating 3D convolution did not reduce the error rate in detecting the contrast of the MRI volume. One possibility is that when using 2D convolution, each MRI volume generates multiple samples at once. With 3D convolution, there are less samples as the entire volume is loaded. In addition, 3D convolution requires excessive computational power, potentially causing memory issues. Meanwhile, 2D convolution results in a simpler algorithm requiring less memory and time to execute, two key concepts when dealing with a large neuroimaging database.

The DL algorithm on its own did not accurately predict the entire testing subset with 100% accuracy. This indicates that the algorithm cannot be used alone. One workaround is to alert the user once the probability of correct classification reported by the algorithm is lower than the operating threshold. The user can then inspect the MRI volume in question. When employing this approach, our proposed algorithm combined with the user alert process resulted in 100% success rate. Yet another measure that can be added alongside our proposed algorithm is the DT portion of the semi-automated current approach. In the rare case that the two methods disagree, the user can go back and take a second look at the volume to resolve which of the two methods is incorrect.

The algorithm was implemented into the processing pipeline and is currently being used by technicians to validate contrasts of unknown MRI scans. The algorithm makes a prediction on the contrast within 4–5 s. The results thus far show that, considering the user's inspection in cases of probabilities lower than the contrast-specific probability thresholds, the overall success rate is 100%. This implementation will generate feedback from a user perspective to allow for improvements in the future.

Conclusions

We developed an automated algorithm to identify the contrast of a MRI volume using DL with CNN architecture. The CNN inferred contrast over n sagittal slices, followed by realizing the volumetric inference using a DNN. The DL algorithm identified between five and eight contrasts with a $< 0.2\%$ error rate. We developed a RF algorithm for comparison and obtained a higher, 1.74% error rate for identification amongst eight contrasts. We demonstrate that reducing the number of MRI volumes used for training to size 2 did not affect the performance of the DL for five contrasts. Reducing the number of MRI volumes used for training to size 3 significantly reduced the algorithm's capacity to generalize. We characterized the DL algorithm for eight contrasts as a contrast-specific identifying tool and computed contrast-specific probability thresholds as a reference for the end-user.

Information Sharing Statement

We made the implementation of our software openly available on GitHub (see link below). We developed the algorithm in Python with a Theano backend and compiled on Keras. Keras is a high-level software package that provides extensive flexibility to easily design and implement deep learning algorithms. We created a Python virtual environment named "deep_env" with the requirements found in the following GitHub link: https://github.com/AS-Lab/Pizarro-et-al-2018-DL-identifies-MRI-contrasts

Acknowledgements This work was supported by the Mathematics of Information Technology and Complex Systems (Mitacs) Canada through the Mitacs Elevate grant. This research was undertaken thanks in part to funding from the Canada First Research Excellence Fund, awarded to McGill University for the Healthy Brains for Healthy Lives initiative. We thank Laura Diamond and Micah Watts for English editing.

References

Al-Rfou, R., Alain, G., Almahairi, A., Angermueller, C., Bahdanau, D., Ballas, N., et al. (2016). Theano: A Python framework for fast computation of mathematical expressions *arXiv preprint arXiv:*1605.02688.

Bengio, Y. (2009). Learning deep architectures for AI. Foundations and trends® in Machine Learning, 2(1), 1–127 %@ 1935–8237.

Breiman, L. (2001). Random forests. Mach Learn, 45(1), 5–32 %@ 0885–6125.

Cheng, X., Pizarro, R., Tong, Y., Zoltick, B., Luo, Q., Weinberger, D. R., et al. (2009). Bio-swarm-pipeline: A light-weight, extensible batch processing system for efficient biomedical data processing. *Front Neuroinform, 3*, 35. https://doi.org/10.3389/neuro.11.035.2009.

Chollet, F. (2015). Keras.

Dahl, G. E., Sainath, T. N., & Hinton, G. E. (2013). *Improving deep neural networks for LVCSR using rectified linear units and dropout* (acoustics, speech and signal processing (ICASSP), 2013 IEEE international conference on): IEEE.

Dozat, T. (2015). Incorporating Nesterov momentum into Adam. Stanford University, Tech Rep, 2015.[Online]. Available: http://cs229.stanford.edu/proj2015/054_report.pdf.

Dunne, R. A., & Campbell, N. A. (1997). *On the pairing of the softmax activation and cross-entropy penalty functions and the derivation of the softmax activation function* (Vol. 185, proc. 8th Aust. Conf. On the neural networks, Melbourne, 181).

Gardner, E. A., Ellis, J. H., Hyde, R. J., Aisen, A. M., Quint, D. J., & Carson, P. L. (1995). Detection of degradation of magnetic resonance (MR) images: Comparison of an automated MR image-quality analysis system with trained human observers. *Acad Radiol, 2*(4), 277–281.

Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift *arXiv preprint arXiv:*1502.03167.

Krizhevsky, A., & Hinton, G. (2009). Learning multiple layers of features from tiny images.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). *Imagenet classification with deep convolutional neural networks* (advances in neural information processing systems).

Marcus, D. S., Harms, M. P., Snyder, A. Z., Jenkinson, M., Wilson, J. A., Glasser, M. F., Barch, D. M., Archie, K. A., Burgess, G. C., Ramaratnam, M., Hodge, M., Horton, W., Herrick, R., Olsen, T., McKay, M., House, M., Hileman, M., Reid, E., Harwell, J., Coalson, T., Schindler, J., Elam, J. S., Curtiss, S. W., van Essen, D., & WU-Minn HCP Consortium. (2013). Human connectome project informatics: Quality control, database services, and data visualization. *Neuroimage, 80*, 202–219. https://doi.org/10.1016/j.neuroimage.2013.05.077.

Murphy, K. P. (2012). *Machine learning : a probabilistic perspective* (adaptive computation and machine learning). Cambridge, mass.: MIT Press.

Pizarro, R. A., Cheng, X., Barnett, A., Lemaitre, H., Verchinski, B. A., Goldman, A. L., Xiao, E., Luo, Q., Berman, K. F., Callicott, J. H., Weinberger, D. R., & Mattay, V. S. (2016). Automated quality assessment of structural magnetic resonance brain images based on a supervised machine learning algorithm. *Front Neuroinform, 10*, 52. https://doi.org/10.3389/fninf.2016.00052.

Ripley, B. D. (2007). *Pattern recognition and neural networks:* Cambridge university press.

Weiner, M. W., Veitch, D. P., Aisen, P. S., Beckett, L. A., Cairns, N. J., Green, R. C., Harvey, D., Jack, C. R., Jagust, W., Liu, E., Morris, J. C., Petersen, R. C., Saykin, A. J., Schmidt, M. E., Shaw, L., Shen, L., Siuciak, J. A., Soares, H., Toga, A. W., Trojanowski, J. Q., & Alzheimer's Disease Neuroimaging Initiative. (2013). The Alzheimer's disease neuroimaging initiative: A review of papers published since its inception. *Alzheimers Dement, 9*(5), e111–e194. https://doi.org/10.1016/j.jalz.2013.05.1769.

Youden, W. J. (1950). Index for rating diagnostic tests. *Cancer, 3*(1), 32–35.

Zuo, X. N., Anderson, J. S., Bellec, P., Birn, R. M., Biswal, B. B., Blautzik, J., Breitner, J. C. S., Buckner, R. L., Calhoun, V. D., Castellanos, F. X., Chen, A., Chen, B., Chen, J., Chen, X., Colcombe, S. J., Courtney, W., Craddock, R. C., di Martino, A., Dong, H. M., Fu, X., Gong, Q., Gorgolewski, K. J., Han, Y., He, Y., He, Y., Ho, E., Holmes, A., Hou, X. H., Huckins, J., Jiang, T., Jiang, Y., Kelley, W., Kelly, C., King, M., LaConte, S. M., Lainhart, J. E., Lei, X., Li, H. J., Li, K., Li, K., Lin, Q., Liu, D., Liu, J., Liu, X., Liu, Y., Lu, G., Lu, J., Luna, B., Luo, J., Lurie, D., Mao, Y., Margulies, D. S., Mayer, A. R., Meindl, T., Meyerand, M. E., Nan, W., Nielsen, J. A., O'Connor, D., Paulsen, D., Prabhakaran, V., Qi, Z., Qiu, J., Shao, C., Shehzad, Z., Tang, W., Villringer, A., Wang, H., Wang, K., Wei, D., Wei, G. X., Weng, X. C., Wu, X., Xu, T., Yang, N., Yang, Z., Zang, Y. F., Zhang, L., Zhang, Q., Zhang, Z., Zhang, Z., Zhao, K., Zhen, Z., Zhou, Y., Zhu, X. T., & Milham, M. P. (2014). An open science resource for establishing reliability and reproducibility in functional connectomics. *Sci Data, 1*, 140049. https://doi.org/10.1038/sdata.2014.49.

AUTHOR QUERIES

AUTHOR PLEASE ANSWER ALL QUERIES.

Q1. Please specify the significance of the data with bold emphasis in Table 2.
Q2. Missing citation for Table 3 was inserted here. Please check if appropriate. Otherwise, please provide citation for Table 3.
Q3. Missing citation for Table 5 was inserted here. Please check if appropriate. Otherwise, please provide citation for Table 5.
Q4. Please check if the URL captured correctly.