Computerized Medical Imaging and Graphics 79 (2020) 101662

Contents lists available at ScienceDirect

Computerized Medical Imaging and Graphics

journal homepage: www.elsevier.com/locate/compmedimag

Automated detection of focal cortical dysplasia using a deep convolutional neural network

Huiquan Wang [a], S. Nizam Ahmed [b], Mrinal Mandal [a,*]

[a] Department of Electrical and Computer Engineering, University of Alberta, Edmonton, Alberta, T6G 2V4, Canada
[b] Department of Medicine, University of Alberta, Edmonton, Alberta, T6G 2B7, Canada

ARTICLE INFO

Article history:
Received 14 August 2018
Received in revised form 12 July 2019
Accepted 1 October 2019

Keywords:
Focal cortical dysplasia
Deep learning
Convolutional neural network
Magnetic resonance imaging
Computer-aided detection

ABSTRACT

Focal cortical dysplasia (FCD) is one of the commonest epileptogenic lesions, and is related to malformations of the cortical development. The findings on magnetic resonance (MR) images are important for the diagnosis and surgical planning of FCD. In this paper, an automated detection technique for FCD is proposed using MR images and deep learning. The input MR image is first preprocessed to correct the bias field, normalize intensities, align with a standard atlas, and strip the non-brain tissues. All cortical patches are then extracted on each axial slice, and these patches are classified into FCD and non-FCD using a deep convolutional neural network (CNN) with five convolutional layers, a max pooling layer, and two fully-connected layers. Finally, the false and missed classifications are corrected in the post-processing stage. The technique is evaluated using images of 10 patients with FCD and 20 controls. The proposed CNN shows a superior performance in classifying cortical image patches compared with multiple CNN architectures. For the system-level evaluation, nine of the ten FCD images are successfully detected, and 85% of the non-FCD images are correctly identified. Overall, this CNN based technique could learn optimal cortical (texture and symmetric) features automatically, and improve the FCD detection.

1. Introduction

Focal cortical dysplasia (FCD) is the malformation of the cortical development, which may be caused by reasons of cortical architecture or cytological abnormalities Kabat and Król (2012). It is the foremost cause of epilepsy in children and the third most significant cause in adults Lerner et al. (2009). Magnetic resonance imaging (MRI) is widely used in identifying FCD as it can provide images of soft tissue with high contrast and resolution. In Fig. 1, three T1-weighted MR images (i.e., one healthy and two FCD images) are shown. The regions enclosed by red rectangular boxes in Figs. 1 (b) and (c) show two typical MRI features of FCD: blurred gray matter (GM) - white matter (WM) boundary and increased cortical thickness. Manual identification of FCD lesion on MR images is a time-consuming and subjective task even for an experienced specialist due to the subtlety of FCD lesions and complexity of brain anatomic structure Rajan et al. (2009). In MR images, multiple tissue types (e.g., GM, WM, cerebrospinal fluid (CSF)) contribute to voxel/pixel intensities, which is known as partial volume effect, and it may also compromise the diagnosis of FCD. Therefore, it is important to develop a computerized system that can process brain MR images and detect subtle lesions automatically and objectively, to assist radiologists in analyzing images and making diagnosis.

Several techniques have been proposed for FCD detection in the literature. These techniques can be broadly divided into two categories: voxel-based morphometry (VBM), and surface-based classification (SBC). In the following, a brief review of techniques in these two categories is presented. Unless mentioned otherwise, T1-weighted MR images are assumed.

The VBM techniquesMechelli et al. (2005) typically normalize images to a standard stereotactic space, segment the normalized images into regions of interest (ROIs), smooth these ROIs, and finally performs a voxel-wise statistical analysis to highlight significant anatomical differences between patients and healthy groups. The techniques usually generate statistical feature maps that are used to detect FCD regions by choosing an optimum threshold. Colliot *et al.* Colliot et al. (2006) proposed to use GM concentration (GMC) map for FCD detection. The GMC map is calculated by segmenting a normalized brain into GM, WM, and CSF, smoothing the GM mask, and calculating z-score Kreyszig (1979) at each voxel in GM region. Note that z-score measures how many standard deviations a GMC value is above or below the mean of controls. A voxel is classified as FCD if its z-score value is greater than a predefined

* Corresponding author.
E-mail addresses: huiquan@ualberta.ca (H. Wang), snahmed@ualberta.ca (S.N. Ahmed), mmandal@ualberta.ca (M. Mandal).

https://doi.org/10.1016/j.compmedimag.2019.101662
0895-6111/

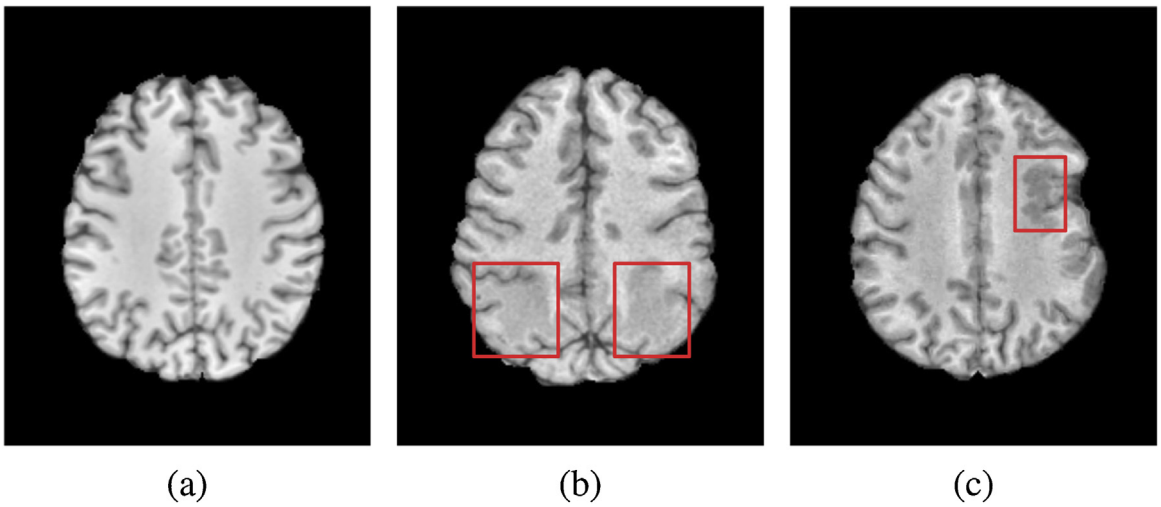

Fig. 1. T1-weighted MR images. (a) An axial slice of a healthy brain, (b) a FCD slice with blurred gray/white matter boundary, (c) a FCD slice with increased cortical thickness (typical values of cortical thickness for adults are between 1.5 mm and 3.0 mm Salat et al. (2004)).

threshold. Similarly, Pail *et al.* Pail et al. (2012) used GMC map to detect FCD within the temporal pole in patients with mesial temporal lobe epilepsy. Wagner *et al.* Wagner et al. (2011) proposed to use GM-WM junction (GWJ) and GM extension (GME) maps for FCD detection. The calculation of GWJ includes brain normalization, segmentation, binarization to obtain GM-WM junctions, smoothing and comparison with normal database. The GME map is calculated using the GM segment of brain. Wong-Kisiel *et al.* Wong-Kisiel et al. (2018) further illustrated the effectiveness of GWJ and GME maps in FCD detection on a larger database of patients and controls. House *et al.* House et al. (2013) also used GWJ map, but based on T2-weighted MR images, to detect FCD regions.

The SBC techniques typically perform cortical reconstruction to obtain inner- and outer-cortical surfaces (using publicly available softwares such as FreeSurfer Laboratory for Computational (Neuroimaging)), extract features at each vertex, and finally classify lesion vertices using the machine learning methods, such as artificial neural network (ANN) and support vector machine(SVM). Different surface-based features and classification models have been proposed in these SBC techniques. Besson *et al.* Besson et al. (2008) calculated five features (i.e., cortical thickness, GM-WM blur, T1 hyper-intensity, sulcal depth, and curvature) at each vertex after cortical surface reconstruction, and performed both vertex-wise classification (using a bagged ANN) and cluster-wise classification (based on the statistic values of clusters). Hong *et al.* Hong et al. (2014) used similar surface features, but vertex-wise classification was performed using Fisher linear discriminant analysis and cluster-wise classification was based on the statistical moments of clusters. Ahmed *et al.* Ahmed et al. (2015) proposed to use bagged logistic regression to classify FCD vertices for "MRI-negative" patients. Adler *et al.* Adler et al. (2017) proposed additional "doughnut" features (calculated on a circle region of vertices on the inflated surface) in the vertex classification using ANN. Clustering was applied in the end to remove false positive FCD clusters. Tan *et al.* Tan et al. (2018) used surface features of both MRI and PET images, followed a two-step classification: a voxel-based SVM to maximize the sensitivity, and a patch-based classifier to remove the false positives. Jin *et al.* Jin et al. (2018) proposed a FCD detection technique similar to Adler et al. (2017), but with a larger database of images obtained from different epilepsy centers.

As observed, most techniques are based on either VBM or SBC. Typically, the VBM techniques use a few neurological features (e.g., GMC, GWJ, and GME), which are used for clinical diagnosis. Although, the VBM techniques have shown a good performance, these techniques are sensitive to artifacts including misalignment, misclassification, and differences in anatomical structures Ashburner (2009). On the other hand, the SBC techniques generate cortical surfaces, and calculate features that are robust to alignment and segmentation artifacts. These techniques are relatively efficient in FCD detection as they take into consideration the anatomical relationships across cortex Hong et al. (2014). But these techniques also have a high computational complexity because of the 3D surface reconstruction Riccelli et al. (2017). A further improvement may be possible by combining both voxel and surface features.

In recent years, deep learning techniques and especially convolutional neural networks (CNN) have shown great potentials in image classification and segmentation problems since they could learn optimal features automatically LeCun et al. (2015). Anthimopoulos *et al.* Anthimopoulos et al. (2016) proposed a lung pattern classification technique for interstitial lung disease based on a deep CNN architecture of 5 convolutional layers with 2×2 kernels, leaky rectifier linear unit (LReLU), and average pooling. The CNN architecture is trained and evaluated on non-overlapping image patches, and it shows superior classification performance over feature-based techniques (e.g., intensity, texture, LBP features). Pereira *et al.* Pereira et al. (2016) proposed a brain tumor segmentation technique based on a CNN architecture with 4-6 convolutional layers, 3×3 kernels, and LReLU. To the best of the authors knowledge, deep learning has not been applied to the detection of FCD. In this paper, we propose an automated FCD detection technique using a deep CNN architecture. The rest of this paper is organized as follows. In Section 2, the materials used in this work are introduced. The proposed methods are presented in Section 3, and the experimental results are reported in Section 4. Section 5 concludes this paper.

2. Materials

In this study, the patient group includes a retrospective cohort of 10 patients who underwent T1-weighted imaging on the 1.5T Siemens MRI scanner at the University of Alberta Hospital, with confirmed FCD lesions. The study was approved by the Alberta health services and University of Alberta research ethics board. The MR images were acquired using the T1-weighted magnetization-prepared rapid-acquisition gradient echo (MPRAGE) sequence (TR = 2130ms, TE = 3.91ms, and flip angle = 15°) with an isotropic voxel dimension of $1\times1\times1\ mm^3$. Each 3D MR image has $208\times256\times192$ (coronal$\times$sagittal$\times$axial) voxels. The FCD lesions are labeled man-

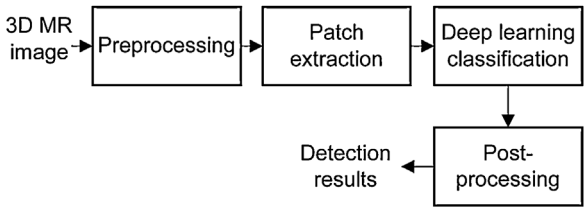

Fig. 2. Schematic of the proposed FCD detection method.

ually on axial slices of the 3D MR images by an expert neurologist at the University of Alberta Hospital.

The control group includes 10 healthy subjects and 10 temporal lobe epilepsy (TLE) patients. The healthy controls were obtained from the IXI dataset Centre for the Developing (Brain), and the T1-weighted images were acquired on a 3.0T Philips MRI scanner at Hammersmith Hospital (TR = 9.6ms, TE = 4.6ms, and flip angle = 8°) with a voxel dimension of $0.9\times0.9\times1.2\ mm^3$ and an image resolution of $256\times256\times150$. The TLE controls were obtained from the Henry Ford Hospital (HFH) dataset Hosseini et al. (2016), Henry Ford (Hospital), and the T1-weighted images were acquired on a 1.5T GE MRI scanner (TR = 7.6ms, TE = 500ms, and flip angle = 20°) with a voxel dimension of $0.8\times0.8\times2\ mm^3$ and an image resolution of $256\times256\times124$. All of the control images were reviewed by an neurologist at the University of Alberta, and confirmed with no FCD lesions.

3. Methods

The overall schematic of the proposed technique is shown in Fig. 2. It includes four modules: 1) preprocessing, 2) patch extraction, 3) deep learning classification, and 4) post-processing. Details of the proposed methods are presented in the following sections.

3.1. Preprocessing

T1-weighted MR images are preprocessed using FMRIB Software Library (FSL) v5.0 Jenkinson et al. (2012), which involves the use of several FSL tools. First, the bias field correction algorithm in the automated segmentation tool Zhang et al. (2001) is applied to correct intensity non-uniformities caused by the low-frequency artifacts. Second, the linear image registration tool Jenkinson et al. (2002) is performed to align the image with a standard brain atlas Fonov et al. (2009). The brain extraction tool Smith (2002) is then used to remove the non-brain tissues from the whole-head image. Finally, intensity normalization is applied to keep the image intensity in the range [0, 1].

3.2. Patch extraction

Since FCD is a malformation of cortical development, neurologists are typically interested in the cortical regions and GM-WM junctions. Therefore, we extract rectangular patches of brain cortex on axial MR slices for FCD detection. The patch extraction is performed by first extracting patch masks on the gray matter probability map (GMPM) of the brain atlas Fonov et al. (2009), and then applying the masks to the preprocessed brain images. Patch masks are extracted using the following procedure.

1) Each axial slice of GMPM is divided into horizontal strips of height h (h = 16 pixels in this work). An example of GMPM division is shown in Fig. 3(a), and a horizontal strip is shown in Fig. 3(b).
2) On each horizontal strip, the Radon transform Deans (2007) is performed vertically to obtain the projection onto the horizontal axis. Fig. 3(c) shows the line plot of the Radon transform of the horizontal strip in Fig. 3(b).
3) To locate the cortex region in a horizontal strip, the values of its Radon transform are checked from left to right. Once the value changes from 0 to a positive value, a rectangular box of size $h \times w$ ($w = 32$ pixels) is placed at this point to be one patch mask. This is shown as the left green box in Fig. 3(d). Based on the symmetric property of the brain atlas and the mirror reflection about the middle sagittal line, the right-side patch mask is obtained, shown as the right green box in Fig. 3(d).
4) Cortical patches along the inter-hemispheric fissure are extracted, shown as the blue rectangles in Fig. 3(e).
5) Repeat step 1) - 4) on every axial slice of GMPM, the patch masks of the whole brain are extracted.

For a preprocessed image, the cortical patches are extracted based on the patch masks obtained from the above procedure. One example slice is shown in Fig. 4, and the extracted image patches are shown as the green and blue rectangular regions. In order to capture both the texture and symmetric properties of cortex, we use a pair of patches as the input to a classification model. Each cortical patch is paired with the mirror reflection of the corresponding patch on the other hemisphere. For example, in Fig. 4, patch pair 1 is formed by the image patch P_1 and the reflection of patch P_2 (noted as $\bar{P}_2$), and patch pair 2 is formed by P_2 and the reflection of P_1 (noted as $\bar{P}_1$). Therefore, the axial slice in Fig. 4 could generate 42 patch pairs since there are 42 patches in this slice. If a patch (e.g., P_1 or P_2) has any overlap with the ground truth FCD regions, it is labeled as FCD. The patch pair has the same label as the image patch.

Using the above procedure, 129780 (30 images × 4326 patches/image) patches are extracted from the dataset (of 30 MR images) in this study. Out of these, 2302 patches are labeled as FCD and 127478 as non-FCD. To overcome the imbalance between FCD and non-FCD patches, data augmentation is applied. The traditional affine transform based data augmentation is not suitable for the cortical dysplasia images because the texture properties of a lesion region may be changed inappropriately after transformation. In this study, we use up-sampling method to increase the number of FCD patches, which is achieved by re-extracting FCD patches starting from the ith ($2 \le i \le h$) row on each axial slice. This results in a total of 36786 FCD patches (with an up-sampling factor of approximately 16). The same number of non-FCD patches are selected randomly to balance the two classes. All patches are divided into training, validation, and testing sets at ratios of 70%, 15%, and 15%, respectively.

3.3. Deep learning classification

Deep learning techniques, especially CNNs, have been used extensively in image classification problems due to their superior performance. In this paper, a CNN architecture is proposed for cortical patch classification, as shown in Fig. 5. The input is one patch pair, and output is the classification result (i.e., probability of being FCD). The CNN architecture consists of five convolutional layers (Conv1 - Conv5), one pooling layer, and two fully-connected layers (FC1 and FC2). The details of this architecture are listed in Table 1, and the following types of CNN layers are involved.

1) *Convolutional layers:* In a convolutional layer, a neuron is only connected to a local area of input neurons instead of full-connection so that the number of parameters to be learned is reduced significantly and a network can grow deeper with fewer parameters. In the proposed CNN, each convolutional layer consists of three operations: convolution, batch normalization, and ReLU activation.

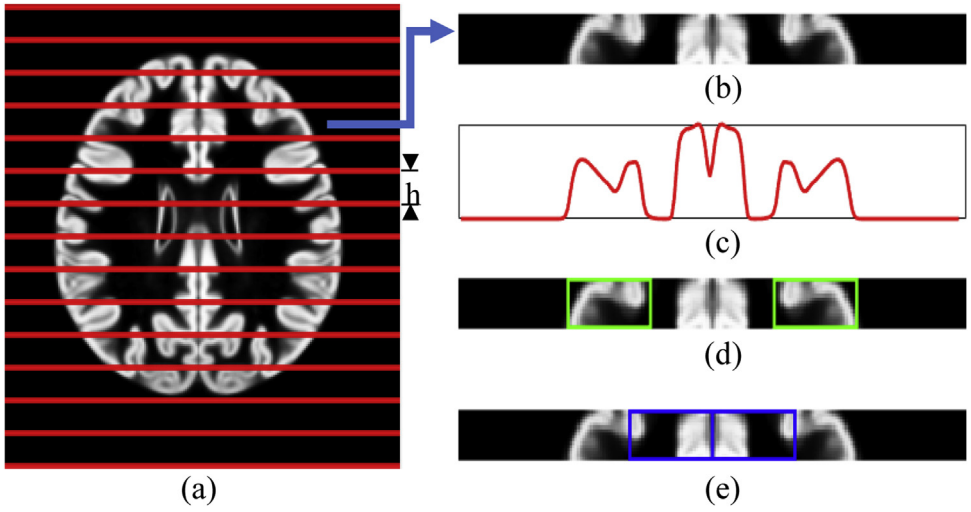

Fig. 3. Patch masks extraction on one axial slice of GMPM. (a) The axial slice divided into horizontal strips of height h, (b) one horizontal strip, (c) line plot of the Radon transform of (b), (d) two external patches (the green boxes), and (e) two middle patches (the blue boxes) along the inter-hemisphere fissure. Note that all patch masks have the same size of $h \times w$.

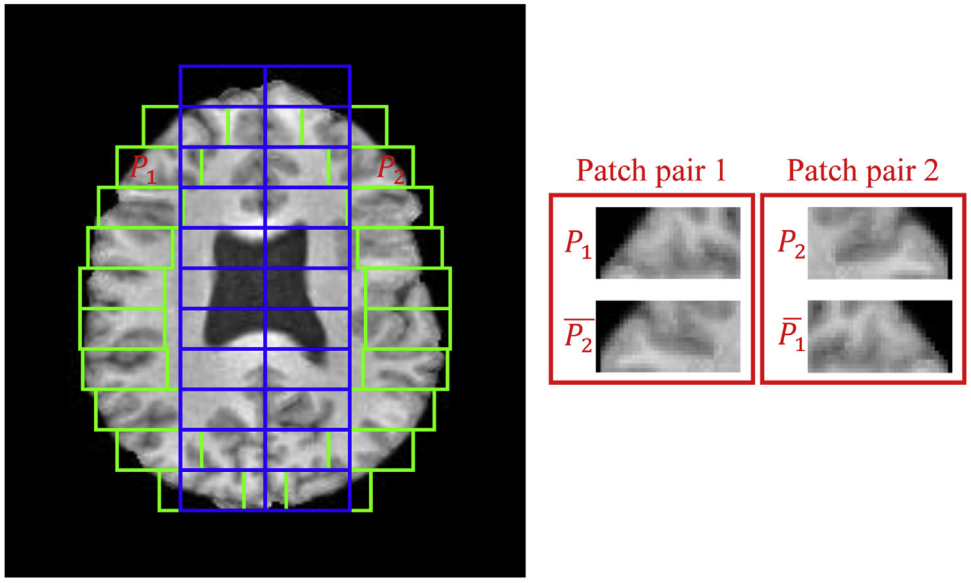

Fig. 4. Patches extracted on one axial slice.

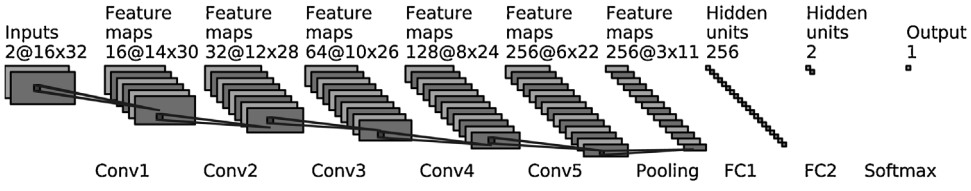

Fig. 5. Schematic of the proposed CNN architecture (with 5 convolutional layers).

(a) *Convolution* The output of a convolution operation is computed by convolving the input with a number of filters, as shown in the following equation:

$$\mathbf{x}_j = \mathbf{I} * \mathbf{W}_j + b_j, j = 1, 2, ..., N \quad (1)$$

where $\mathbf{I}$ is the input, N is the number of filters, $\mathbf{x}_j$ is the output corresponding to the jth convolution filter, $\mathbf{W}_j$ is the weights of the jth filter, and b_j is the jth bias. In the first convolutional layer of the network in Fig. 5, $\mathbf{I}$, $\mathbf{W}_j$, and $\mathbf{x}_j$ have dimensions of $16 \times 32 \times 2$, $3 \times 3 \times 2$, and 14×30, respectively. Note that $\mathbf{x}_j$ does not include any partial convolution output. As a result, the feature map dimension decreases after each convolution operation. In CNN, the number of filters and the filter size are two hyperparameters that need to be specified. The number of filters, N, at each convolutional layer, is defined as follows:

$$N_l = k * 2^{l+1}, l = 1, 2, ..., L \quad (2)$$

Table 1
Details of the proposed CNN architecture with 5 convolutional layers, 3×3 filters, and scale parameter $k = 4$.

Layer	Type	Filters (#, size)	Activations	Learnables
Inputs	Input	-	16×32×2	
Conv1	Convolution	16, 3×3	14×30×16	Weight: 3×3×2×16 Bias: 16
	Batch Normalization	-	14×30×16	Offset: 16 Scale: 16
	ReLU	-	14×30×16	-
Conv2	Convolution	32, 3×3	12×28×32	Weight: 3×3×16×32 Bias: 32
	Batch Normalization	-	12×28×32	Offset: 32 Scale: 32
	ReLU	-	12×28×32	-
Conv3	Convolution	64, 3×3	10×26×64	Weight: 3×3×32×64 Bias: 64
	Batch Normalization	-	10×26×64	Offset: 64 Scale: 64
	ReLU	-	10×26×64	-
Conv4	Convolution	128, 3×3	8×24×128	Weight: 3×3×64×128 Bias: 128
	Batch Normalization	-	8×24×128	Offset: 128 Scale: 128
	ReLU	-	8×24×128	-
Conv5	Convolution	256, 3×3	6×22×256	Weight: 3×3×128×256 Bias: 256
	Batch Normalization	-	6×22×256	Offset: 256 Scale: 256
	ReLU	-	6×22×256	-
Pooling	Max Pooling	-	3×11×256	-
FC1	Dropout	-	8448	-
	Fully-connected	-	256	Weight: 8448×256 Bias: 256
FC2	Fully-connected	-	2	Weight: 256×2 Bias: 2
Outputs	Softmax	-	1	

where N_l is the number of filters at the lth convolutional layer, L is the total number of convolutional layers (e.g., $L = 5$), and k is a scale parameter.

(b) *Batch normalization:* After convolution operation, batch normalization is used to speed up the training of CNN and reduce the effect of initialization Ioffe and Szegedy (2015). It normalizes each input channel across a mini-batch by subtracting mean of the mini-batch and dividing by its standard deviation. The normalized values are then scaled and shifted as follows:

$$y_i = \sigma \hat{x}_i + \beta \tag{3}$$

where y_i is output value, $\hat{x}_i$ is the normalized input value, σ and β are the scale and offset factors that are learnable during the network training.

(c) *ReLU:* Nonlinear activation functions are applied element-wise to increase the non-linear properties of the network. In CNN, rectified linear unit (ReLU) is the most commonly used nonlinear activation function Nair and Hinton (2010), and it replaces all negative pixel values by zeros using the following equation:

$$f = \max(0, y) \tag{4}$$

where y is the input value to ReLU, and f is the output. Compared with *tanh* and *sigmoid* activations, ReLU has the advantages of speeding up training Krizhevsky et al. (2012) and inducing the sparsity in hidden units Glorot et al. (2011).

Step (a)-(c) are repeated five times (as shown in Table 1). After the ReLU module, the feature map f is used as the input (I) of the next convolutional layer (if appropriate). After the five convolutional layers in the proposed architecture, 256 features maps, each of dimension 6×22, are obtained, as shown in Table 1. These are then down-sampled in a pooling layer.

2) *Pooling:* In a pooling layer, the feature maps are down-sampled using an average or a max operation to reduce the number of parameters and control overfitting in a network. In this study, a max pooling layer with filter of size 2×2 with a stride of 2 is used. After the pooling, there are 256 feature maps of size 3×11. After the pooling, the feature maps are flattened and concatenated to form a 8448×1 feature vector.

3) *Fully-connected (FC) layers:* The proposed CNN has two FC layers as shown in Fig. 6. After pooling and flattening, the feature vector of size 8448×1 is fully connected to 256 nodes at the first FC layer. The second FC layer has 2 nodes, and its output scores (noted as z_i, $i = 1, 2$) are fed to a softmax layer where the classification probability is calculated.

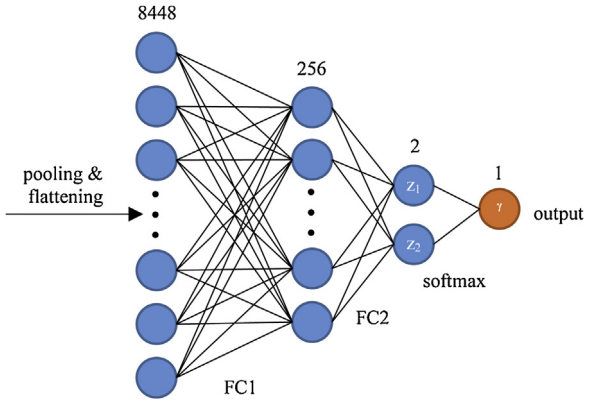

Fig. 6. Fully-connected layers.

As observed in Table 1, the proposed network has a total of 2556914 parameters (numbers of parameters listed in the last column) that need to be updated during the training process, and the first FC layer takes approximately 85% of this number. To speed up training Srivastava et al. (2014), dropout is used in the first FC layer removing some nodes with a probability ($p = 0.4$) during training. In other words, 40% of the 8448×256 weights (chosen randomly) are not updated at each training iteration.

4) *Softmax:* The softmax function is applied at the last layer to generate the probability distribution of the classification results. The predicted probability is computed using the following equation.

$$\gamma = \frac{e^{z_1}}{e^{z_1} + e^{z_2}} \tag{5}$$

where γ is the probability that the input belongs to the FCD class.

Training: To train the CNN, the loss function and the optimization algorithm must be defined. In this study, we use the cross

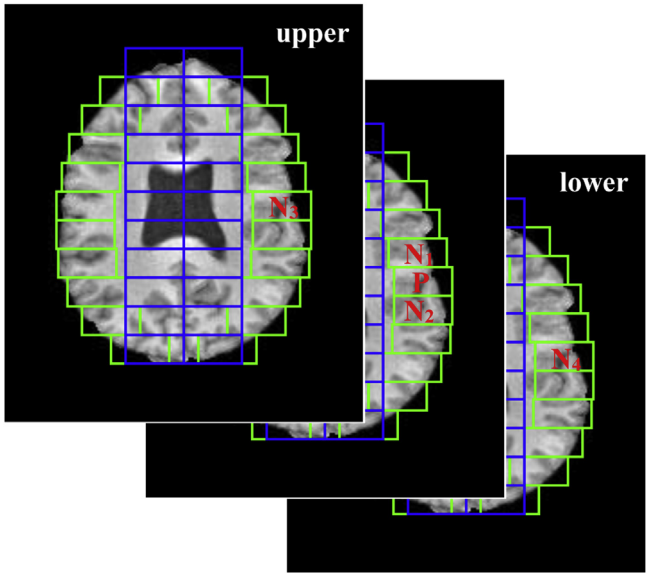

Fig. 7. A patch (P) and its four neighbors (N_1-N_4).

entropy loss function for multi-class classification, which is defined as follows:

$$E = -\sum_{s=1}^{S} t_s \ln \gamma_s \tag{6}$$

where S is the number of training samples (i.e., image patch pairs), t_s is the ground truth value that the sth sample belongs to the FCD class, and γ_s is the output probability of the network. To minimize the loss function E, the stochastic gradient descent with momentum (SGDM) Robert (2014) optimizer is used in the proposed technique. The SGDM accelerates stochastic gradient descent (SGD) and reduces the oscillation problem of SGD by adding the contribution from the previous iteration to the current iteration. The update to the network parameters is performed using the following equation:

$$\boldsymbol{\theta}_{n+1} = \boldsymbol{\theta}_n - \alpha \nabla E(\boldsymbol{\theta}_n) + \mu(\boldsymbol{\theta}_n - \boldsymbol{\theta}_{n-1}) \tag{7}$$

where n is the iteration number, $\boldsymbol{\theta}$ is the parameter vector, $\nabla E(\boldsymbol{\theta})$ is the gradient of the loss function, α is the learning rate, and μ is momentum. In this work, the momentum is set to 0.9. The learning rate is initialized to 0.01, and drops every 10 epochs by a factor of 0.2. The mini-batch size is set to 128 and the maximum number of epochs is set to 40.

3.4. Post-processing

A patch is considered as FCD if $\gamma > \tau$, where τ is a threshold that maximizes the validation accuracy. In the post-processing stage, the classification results of both a patch and its four neighbors are considered. As shown in Fig. 7, the four neighbors of a patch P include two patches (N_1, N_2) from the same slice, one patch (N_3) from the upper slice, and one (N_4) from the lower slice. The following two steps are performed to remove noises and correct misclassifications. 1) If P is classified as FCD and N_1-N_4 are classified as non-FCD, P is treated as a noise and re-labeled as non-FCD. 2) If P is a non-FCD patch and N_1-N_4 are FCD patches, P is treated as a misclassification and re-labeled as FCD. After post-processing, an image with FCD is corrected identified if the detected FCD patches and the ground-truth FCD regions have overlap.

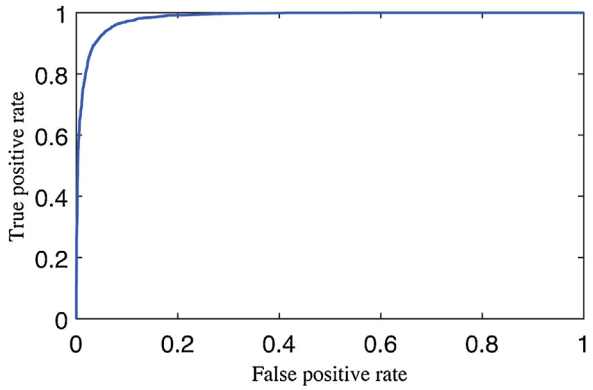

Fig. 8. The ROC curve for the proposed CNN.

4. Experimental results

In this section, we first present the implementation of the proposed technique. The evaluation results of the proposed CNN model are then provided. Finally, the system-level evaluation is presented.

4.1. Implementation

The proposed technique is implemented using MATLAB R2018a with the neural network toolbox. All experiments are performed on a Windows 10 computer with Intel i7-6700K CPU and 16 GB RAM. The CNN architectures are trained using a NVIDIA GeForce GTX 970 graphic card.

4.2. Evaluation of the CNN

In this section, the classification performances of multiple CNN architectures are evaluated and compared with the proposed network. These architectures are constructed based on the variation of the hyperparameters (i.e., number of layers, number of filters, type of activation function, and dropout probability). Each architecture is trained on the training dataset, and fine tunned on the validation dataset. The performance of each architecture is evaluated on the testing patch dataset using accuracy, which is calculated using the following equation.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{8}$$

where TP, FP, TN and FN are the numbers of true-positive, false-positive, true-negative, and false-negative patches, respectively.

Table 2 compares multiple CNN architectures in the patch classification performance. In this table, the last row in bold shows the testing accuracy (0.941) of the proposed architecture. The CNNs using 4 and 6 convolutional layers obtain a classification accuracy of 0.892 and 0.905, respectively, which are roughly a 5% drop in accuracy. Using smaller (2×2) filters reduces the performance by about 5%, and using larger (4×4) filters by another 1%. It illustrates that the kernel size 3×3 achieves the best accuracy in this experiment. To determine the parameter k, values of 3 and 5 are tested, and it is observed that neither decreasing nor increasing the number of filters could increase the performance. LReLU activations with scale factors 0.01 and 0.1 are also evaluated with the proposed CNN, and they are inferior to the use of the ReLU activation function. The use of a dropout layer with a probability of 0.4 could increase the classification accuracy from 0.897 to 0.941 comparing with the CNN without dropout. Using either a lower (0.3) or a higher (0.5) dropout probability results in a performance drop.

Fig. 8 shows the receiver operating characteristic (ROC) curve for the proposed CNN. It illustrates the true positive rate against

Table 2
Performance of different CNN architectures. Acc.: accuracy.

# layers	Filter size	# filters (k)	Activation function	Dropout (p)	Testing acc.
4	3×3	4	ReLU	0.4	0.892
6	3×3	4	ReLU	0.4	0.905
5	2×2	4	ReLU	0.4	0.897
5	4×4	4	ReLU	0.4	0.881
5	3×3	3	ReLU	0.4	0.903
5	3×3	5	ReLU	0.4	0.914
5	3×3	4	LReLU(0.01)	0.4	0.930
5	3×3	4	LReLU(0.1)	0.4	0.918
5	3×3	4	ReLU	0	0.897
5	3×3	4	ReLU	0.3	0.911
5	3×3	4	ReLU	0.5	0.930
5	**3×3**	**4**	**ReLU**	**0.4**	**0.941**

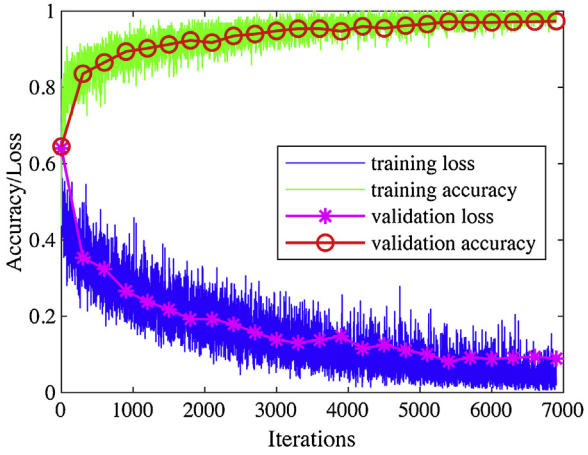

Fig. 9. Loss and accuracy plots during the network training process.

false positive rate based on the classification results on the testing dataset. The area under the ROC curve is 0.985, which indicates a good classification performance Mandrekar (2010) of the proposed CNN.

In Fig. 9, it shows the loss and performance curves during the training of the proposed CNN. Each iteration is corresponding to an update to the network parameters based on the mini-batch gradients. The training accuracy and loss are calculated on each individual mini-batch. The validation accuracy and loss are calculated on the validation dataset at the end of every 300 iterations. The network training stops when the validation accuracy is not increasing in the previous 5 validations, and it takes approximately 19 minutes. In Fig. 9, the training and validation accuracy curves are generally consistent during the network training process. The small divergence between the two at the end of the training, however, may be related to the emerging of overfitting. Similar trend is also observed from the training and validation loss curves.

The 16 filters of the first convolutional layer are shown in Fig. 10. In Fig. 10, row 1-2 shows the two channels of the first 8 filters, and row 3-4 are the two channels of the other 8 filters. It is observed that both high and low frequency patterns are captured by using this group of filters. The similarity of the two filter channels may indicate that these filters could capture the symmetric properties of the input patch pair.

Fig. 11 shows the testing accuracy of the proposed CNN versus the patch size. The patch size changes from 14×28 pixels to 18×36 pixels, and it is observed that the accuracy varies between 0.82 and 0.94. Therefore, the patch size of 16×32 pixels is the most reasonable of all of the tested patches.

Table 3
Confusion matrix for FCD detection with 30 MR images.

		Prediction		
		FCD	Non-FCD	Total
Actual	FCD	9	1	10
	Non-FCD	3	17	20
	Total	12	18	30

4.3. Evaluation of the system

To evaluate the FCD detection performance of the proposed system, we use leave-one-out cross validation strategy that all images but one are used for training the CNN model, and the left image is tested. This procedure is repeated until all images are evaluated. The FCD detection performance is measured using sensitivity, specificity, and accuracy, which are defined using the following equations:

$$Sensitivity = \frac{TP}{TP + FN} \tag{9}$$

$$Specificity = \frac{TN}{TN + FP} \tag{10}$$

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{11}$$

where TP and TN are the numbers of correctly detected FCD and non-FCD images, respectively. FP and FN are the numbers of the false detections - FCD and non-FCD, respectively.

To evaluate the FCD segmentation performance of the proposed system, the detected FCD patches are mapped to the inflated cortical surfaces (obtained using the FreeSurfer software Laboratory for Computational (Neuroimaging)) and the dice coefficient is calculated using the following equation.

$$Dice = \frac{2\,(A \cap B)}{A \cup B} \tag{12}$$

where $A \cap B$ is the number of vertices common to the detected (A) and the ground-truth (B) FCD surface, and $A \cup B$ is the number of vertices in either A or B.

Table 3 shows the confusion matrix of the FCD detection. Nine of the ten FCD images are detected correctly using the proposed system. For non-FCD images, 85% of them are correctly identified, and all of the three false detections are from the HFH dataset that has a larger slice thickness than the IXI dataset. It indicates the cortical structures may have some unnoticeable distortions due to either the effect of the disease (TLE) or the interpolation artifacts in image preprocessing. It is possible to increase the specificity by using higher resolution images.

Fig. 12 shows an example of correct detections with a 3D view of the lesion on the inflated cortical surface. In Fig. 12(a), it is an axial slice with marked FCD lesion along the red line. Patch extraction and classification results of this slice are shown in Fig. 12(b) and

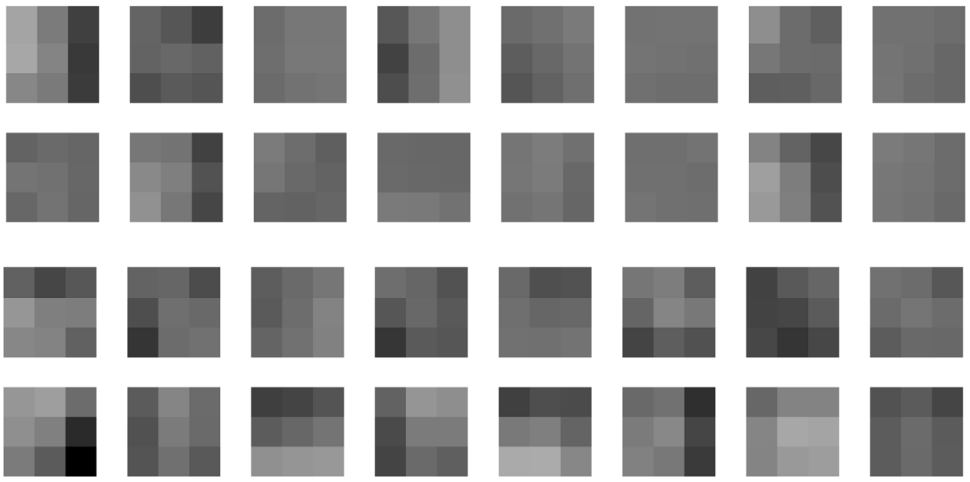

Fig. 10. The first layer filters.

Table 4
Performance comparison with existing techniques for FCD detection. Spe.: specificity, sen.: sensitivity, acc: accuracy

Paper	Method	Spe.	Sen.	Acc.
Pail, 2012 Pail et al. (2012)	VBM	0.60	0.70	0.66
Ahmed, 2015 Ahmed et al. (2015)	SBC	0.75	0.60	0.70
Jin, 2018 Jin et al. (2018)	SBC	0.75	0.70	0.73
Proposed	CNN	**0.85**	**0.90**	**0.88**

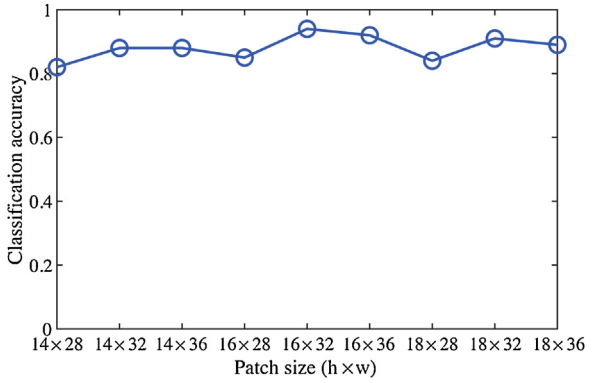

Fig. 11. The classification accuracy versus image patch size.

Fig. 12(c), respectively. The magenta-color numbers in Fig. 12(c) are the predicted probabilities of these patches being FCD, and the high values at FCD regions illustrate the effectiveness of the proposed classification model. After post-processing, the FCD patches are mapped to the inflated cortical surface (obtained using FreeSurfer) so that the predicted FCD region could be viewed in 3D as the yellow region in Fig. 12(d).

Fig. 13 shows the FCD image that is not successfully detected by the proposed technique. It is observed that although the lesion is subtle, the proposed CNN could classify the FCD patch with a high probability (0.99). But the post-processing stage mistakenly removed the FCD patch, resulting in the miss detection. To increase the sensitivity of the FCD detection, a smaller threshold (τ)) could be chosen, but it in turn will decrease the specificity.

Table 4 shows a performance comparison between the proposed technique with three existing techniques using the images in this work. For the VBM technique Pail et al. (2012), the optimal z-score threshold (of 3.5) is chosen to maximize the detection accuracy based on the GMC maps. The two SBC techniques Ahmed et al. (2015), Jin et al. (2018) are supervised classification techniques, and they are evaluated using the leave-one-out strategy. It is observed that the proposed technique provides a detection accuracy of 0.88, which is a 0.15 increase over the state of the art at 0.73. The specificity and sensitivity are also observed superior to the other three techniques. It indicates that the proposed deep learning technique may be a valuable tool for FCD detection and patient evaluation.

Table 5 compares the segmentation accuracy of the correctly detected FCD regions between the proposed technique and two SBC-based techniques Ahmed et al. (2015), Jin et al. (2018) using surface mapping and dice coefficient. After FCD detection, we first reconstructed the cortical surface of each image using the FreeSurfer software, and then mapped the detection results to this surface. The dice coefficient is calculated using Eq. (12) for each

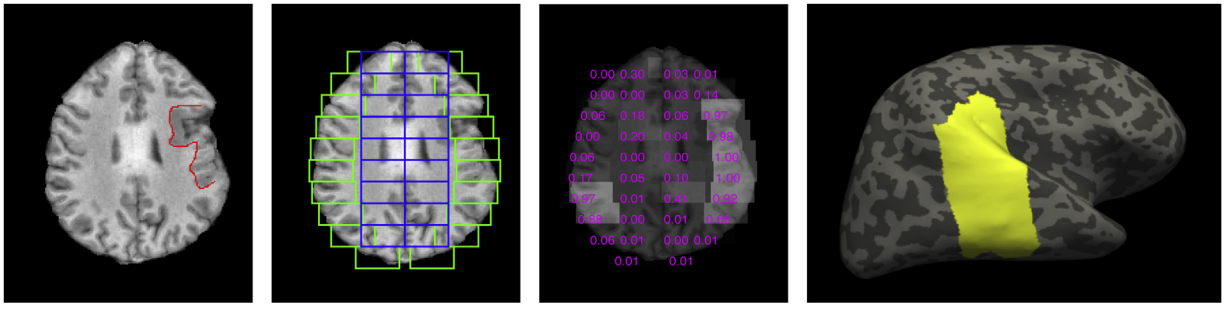

Fig. 12. An example of FCD detection. (a) an axial slice with FCD lesion labeled in red, (b) patch extraction results, (c) classification results and numbers stand for probabilities of being FCD patches, (d) detection results mapped onto the inflated cortical surface (shown in yellow).

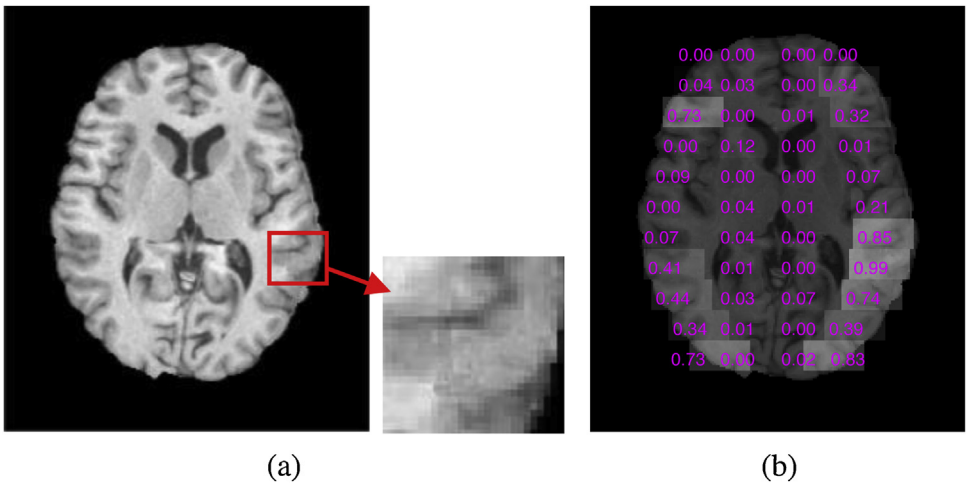

Fig. 13. The missed FCD detection. (a) an axial slice with the enlarged FCD lesion, (b) patch probabilities (in magenta) of being FCD.

Table 5
Performance comparison with existing techniques for FCD segmentation.

Technique	Dice
Ahmed, 2015 Ahmed et al. (2015)	0.43
Jin, 2018 Jin et al. (2018)	0.55
Proposed	**0.78**

FCD lesion. The mean dice coefficients are reported in Table 5. It is observed that the proposed technique can provide a higher segmentation accuracy.

5. Conclusion

In this paper, an automated technique is proposed for FCD detection in T1-weighted MR images using a deep convolutional neural network. The proposed technique first performs preprocessing to align the input image with a standard brain atlas. The cortical patches are then extracted on axial slices. Each patch is paired and fed to a CNN classifier with 5 convolutional layers, 1 max pooling layer, and 2 fully-connected layers. Finally, the post-processing stage removes noises and correct missed detections. Experimental results show a superior performance (90% sensitivity) of the proposed technique compared to the state of the art (70% sensitivity).

Acknowledgments

This work was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) (grant number: RGPIN-2014-05215). We also acknowledge the support of China Scholarship Council (CSC).

References

Kabat, J., Król, P., 2012. Focal cortical dysplasia-review. Polish journal of radiology 77 (2), 35.
Lerner, J.T., Salamon, N., Hauptman, J.S., Velasco, T.R., Hemb, M., Wu, J.Y., Sankar, R., Donald Shields, W., Engel Jr, J., Fried, I., et al., 2009. Assessment and surgical outcomes for mild type I and severe type II cortical dysplasia: a critical review and the UCLA experience. Epilepsia 50 (6), 1310–1335.
Rajan, J., Kannan, K., Kesavadas, C., Thomas, B., 2009. Focal cortical dysplasia (FCD) lesion analysis with complex diffusion approach. Computerized Medical Imaging and Graphics 33 (7), 553–558.
Salat, D.H., Buckner, R.L., Snyder, A.Z., Greve, D.N., Desikan, R.S., Busa, E., Morris, J.C., Dale, A.M., Fischl, B., 2004. Thinning of the cerebral cortex in aging. Cerebral cortex 14 (7), 721–730.
Mechelli, A., Price, C.J., Friston, K.J., Ashburner, J., 2005. Voxel-based morphometry of the human brain: methods and applications. Current medical imaging reviews 1 (2), 105–113.
Colliot, O., Bernasconi, N., Khalili, N., Antel, S.B., Naessens, V., Bernasconi, A., 2006. Individual voxel-based analysis of gray matter in focal cortical dysplasia. Neuroimage 29 (1), 162–171.
Kreyszig, E.O., 1979. Advanced Engineering Mathematics. Wiley.
Pail, M., Marecek, R., Hermanová, M., Slaná, B., Tyrlíková, I., Kuba, R., Brázdil, M., 2012. The role of voxel-based morphometry in the detection of cortical dysplasia within the temporal pole in patients with intractable mesial temporal lobe epilepsy. Epilepsia 53 (6), 1004–1012.
Wagner, J., Weber, B., Urbach, H., Elger, C.E., Huppertz, H.-J., 2011. Morphometric MRI analysis improves detection of focal cortical dysplasia type II. Brain 134 (10), 2844–2854.
Wong-Kisiel, L.C., Quiroga, D.F.T., et al., 2018. Morphometric analysis on T1-weighted MRI complements visual MRI review in focal cortical dysplasia. Epilepsy Res. 140, 184–191.
House, P.M., Lanz, M., Holst, B., Martens, T., Stodieck, S., Huppertz, H.-J., 2013. Comparison of morphometric analysis based on T1-and T2-weighted MRI data for visualization of focal cortical dysplasia. Epilepsy Res 106 (3), 403–409.
Laboratory for Computational Neuroimaging, FreeSurfer. [Online]. Available: https://surfer.nmr.mgh.harvard.edu/.
Besson, P., Bernasconi, N., Colliot, O., Evans, A., Bernasconi, A., 2008. Surface-based texture and morphological analysis detects subtle cortical dysplasia. International Conference on Medical Image Computing and Computer-Assisted Intervention, 645–652.
Hong, S.-J., Kim, H., Schrader, D., Bernasconi, N., Bernhardt, B.C., Bernasconi, A., 2014. Automated detection of cortical dysplasia type II in MRI-negative epilepsy. Neurology 83 (1), 48–55.
Ahmed, B., Brodley, C.E., Blackmon, K.E., Kuzniecky, R., Barash, G., Carlson, C., et al., 2015. Cortical feature analysis and machine learning improves detection of "MRI-negative" focal cortical dysplasia. Epilepsy & Behavior 48, 21–28.
Adler, S., Wagstyl, K., Gunny, R., Ronan, L., Carmichael, D., Cross, J.H., et al., 2017. Novel surface features for automated detection of focal cortical dysplasias in paediatric epilepsy. NeuroImage: Clinical 14, 18–27.
Tan, Y.-L., Kim, H., et al., 2018. Quantitative surface analysis of combined MRI and PET enhances detection of focal cortical dysplasias. Neuroimage 166, 10–18.
Jin, B., Krishnan, B., Adler, S., Wagstyl, K., Hu, W., Jones, S., Najm, I., Alexopoulos, A., Zhang, K., Zhang, J., et al., 2018. Automated detection of focal cortical dysplasia type II with surface-based magnetic resonance imaging postprocessing and machine learning. Epilepsia 59 (5), 982–992.
Ashburner, J., 2009. Computational anatomy with the SPM software. Magnetic Resonance Imaging 27 (8), 1163–1174.
Riccelli, R., Toschi, N., Nigro, S., Terracciano, A., Passamonti, L., 2017. Surface-based morphometry reveals the neuroanatomical basis of the five-factor model of personality. Social cognitive and affective neuroscience 12 (4), 671–684.
LeCun, Y., Bengio, Y., Hinton, G., 2015. Deep learning. nature 521 (7553), 436.
Anthimopoulos, M.M., Christodoulidis, S., Ebner, L., Christe, A., Mougiakakou, S., 2016. Lung pattern classification for interstitial lung diseases using a deep convolutional neural network. IEEE Transactions on Medical Imaging 35 (5 (May)), 1207–1216.
Pereira, S., Pinto, A., Alves, V., Silva, C.A., 2016. Brain tumor segmentation using convolutional neural networks in MRI images. IEEE Transactions on Medical Imaging 35 (5 (May)), 1240–1251.
Centre for the Developing Brain, Imperial College London, IXI dataset. [Online]. Available: http://brain-development.org/ixi-dataset/.
Hosseini, M.-P., Nazem-Zadeh, M.-R., Pompili, D., Jafari-Khouzani, K., Elisevich, K., Soltanian-Zadeh, H., 2016. Comparative performance evaluation of automated segmentation methods of hippocampus from magnetic resonance images of temporal lobe epilepsy patients. Med Phys 43 (1), 538–553.

Henry Ford Hospital, Temporal lobe epilepsy multiple modality data. [Online]. Available: https://www.nitrc.org/projects/hfh_t1_hp_seg1.
Jenkinson, M., Beckmann, C.F., Behrens, T.E., Woolrich, M.W., Smith, S.M., 2012. FSL. Neuroimage 62 (2), 782–790.
Zhang, Y., Brady, M., Smith, S., 2001. Segmentation of brain MR images through a hidden Markov random field model and the expectation-maximization algorithm. IEEE transactions on medical imaging 20 (1), 45–57.
Jenkinson, M., Bannister, P., Brady, M., Smith, S., 2002. Improved optimization for the robust and accurate linear registration and motion correction of brain images. Neuroimage 17 (2), 825–841.
Fonov, V.S., Evans, A.C., McKinstry, R.C., Almli, C., Collins, D., 2009. Unbiased nonlinear average age-appropriate brain templates from birth to adulthood. NeuroImage 47, S102.
Smith, S.M., 2002. Fast robust automated brain extraction. Human brain mapping 17 (3), 143–155.
Deans, S.R., 2007. The Radon transform and some of its applications. Courier Corporation.
S. Ioffe and C. Szegedy, Batch normalization: accelerating deep network training by reducing internal covariate shift, arXiv preprint arXiv:1502.03167, 2015.
Nair, V., Hinton, G.E., 2010. Rectified linear units improve restricted boltzmann machines. Proceedings of the 27th international conference on machine learning, 807–814.
Krizhevsky, A., Sutskever, I., Hinton, G.E., 2012. Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems., pp. 1097–1105.
Glorot, X., Bordes, A., Bengio, Y., 2011. Deep sparse rectifier neural networks. Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, 315–323.
Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R., 2014. Dropout: a simple way to prevent neural networks from overfitting. The Journal of Machine Learning Research 15 (1), 1929–1958.
Robert, C., 2014. Machine learning, a probabilistic perspective. CHANCE 27 (2), 62–63.
Mandrekar, J.N., 2010. Receiver operating characteristic curve in diagnostic test assessment. Journal of Thoracic Oncology 5 (9), 1315–1316.