An Automated Framework for Multi-label Brain Tumor Segmentation based on Kernel Sparse Representation

Xuan Chen[1], Binh P. Nguyen[3], Chee-Kong Chui[2], Sim-Heng Ong[1]

[1]Department of Electrical and Computer Engineering, National University of Singapore, 4 Engineering Drive 3, 117583, Singapore
[2]Department of Mechanical Engineering, National University of Singapore, 9 Engineering Drive 1, 117575, Singapore
[3]Centre for Computational Biology, Duke-NUS Medical School, 8 College Road, 169857, Singapore
xuan.chen@u.nus.edu, phubinh@ieee.org,
mpecck@nus.edu.sg, eleongsh@nus.edu.sg

Abstract: A novel automated framework is proposed in this paper to address the significant but challenging task of multi-label brain tumor segmentation. Kernel sparse representation, which produces discriminative sparse codes to represent features in a high-dimensional feature space, is the key component of the proposed framework. The graph-cut method is integrated into the framework to make a segmentation decision based on both the kernel sparse representation and the topological information of brain structures. A splitting technique based on principal component analysis (PCA) is adopted as an initialization component for the dictionary learning procedure, which significantly reduces the processing time without sacrificing performance. The proposed framework is evaluated on the multi-label Brain Tumor Segmentation (BRATS) Benchmark. The evaluation results demonstrate that the proposed framework is able to achieve compatible performance and better generalization ability compared to the state-of-the-art approaches.

Keywords: Brain tumor segmentation, kernel methods, superpixels, PCA, sparse coding, dictionary learning, graph-cuts

1 Introduction

Brain tumor refers to uncontrollable cell proliferation in the brain. Even though brain tumor is not a common disease, with prevalence of less than 0.1% in the western population, it results in high mortality [1]. The topic of brain tumor segmentation has long attracted researchers' attention because of its value in medical diagnosis and treatment planning. Brain tumor segmentation intends to separate tumors from non-tumor regions and classify brain tumor tissues according to predefined criteria [2]. Manual segmentation done by experts is possible but impractical, since it is tedious and time-consuming. Hence, semi-automated and automated approaches, which require less or even no human intervention, are practical alternatives.

Magnetic resonance (MR) imaging is preferable in brain imaging due its advantages of safety, better tissue contrast and fewer artifacts than computed tomography (CT). This emphasizes the significance of efficient and effective frameworks for brain tumor segmentation based on MR images. However, brain tumors exhibit a wide range in shape, size as well as location, and share intensities with normal brain regions in MR images. Besides, the structure of the tumor is usually complex. Therefore, much effort has been expended in the development of semi-automated or automated frameworks for brain tumor segmentation, especially multi-label brain tumor segmentation.

The past few decades have witnessed significant advances in the field of brain tumor segmentation. The approaches to brain tumor segmentation can be roughly classified into two categories: generative methods and discriminative methods. In generative methods, the anatomy and statistics of different brain tissues are explicitly modeled, while the features of task-relevant brain tissues are directly learned from training sets in discriminative methods [3]. Generative methods, although they have to deal with difficulties in modeling the prior knowledge of brain tissues and elaborate non-rigid registration, usually have better generalization ability on unseen images. Discriminative methods, which avoid the difficulties in modeling and registration, are sensitive to the amount and quality of training data.

The expectation-maximization (EM) algorithm usually plays an important role in the generative methods. Based on the statistics of the healthy brain, an outlier detection framework is proposed by Prastawa et al. [4] which treats brain tumor as outlier and generates model of tumors for subsequent EM segmentation. Menze et al. [5] incorporate multi-channel priors to augment the traditional atlas-based EM segmentation. Khotanlou et al. [6] introduce a two-step segmentation procedure, which includes tumor detection and initial segmentation refinement by fuzzy classification. Gooya et al. [7] describe a glioma growth model that is integrated with the inference of patient specific atlas to guides the EM-based segmentation.

Much research has been done in advancing discriminative methods. The classic level-set method [8, 9] is utilized due to its strength in following the change of object topology. The success of the random forest algorithm, which is essentially an ensemble classifier, in the multi-label Brain Tumor Segmentation (BRATS) challenge 2012 has boosted its popularity in the following years [10, 11].

The fact that sparse or compressible representations for signals and images are employed in some predefined or learned representation systems, also known as dictionaries, is the core of the well-known sparse coding algorithm. Compared to predefined dictionaries, learned dictionaries usually provide better sparse representations and hence more satisfying results [12]. Therefore, sparse coding and dictionary learning are commonly used together. Applications based on sparse representation using sparse coding and dictionary learning can be found in various tasks, e.g., image classification [13]. Instead of the explicit raw representation of data, kernel extension of sparse coding and dictionary learning work in an implicit, high-dimensional feature space to achieve more discriminative sparse representation. Kernel sparse representation has been utilized in the brain tumor segmentation task and its effectiveness in distinguishing tumor from normal brain regions has been demonstrated [14, 15]. However, multi-label brain tumor segmentation, which is more challenging compared to binary brain tumor segmentation, is not considered in their frameworks.

In this paper, we propose a fully automated framework based on kernel sparse representation for multi-label brain tumor segmentation. In the proposed framework, superpixels are used as basic processing units instead of traditional pixels [14] or patches [15]. A pixel-based framework involves much repeated effort in encoding similar pixels. In contrast, patches usually exhibit obvious inhomogeneity, though patch-based frameworks may be more efficient than their pixel-based counterparts. In the proposed framework, the sparse representation of each superpixel is generated in a high-dimensional feature space, where the nonlinear similarity among superpixels is more discriminative. Kernel dictionary learning is applied to learn class-specific dictionaries based on superpixel-level features including histogram and spatial location, while kernel sparse coding uses the learned dictionaries and features to generate a sparse representation for a given superpixel. The graph-cut method, which naturally take topological information into consideration, is employed in the framework. Kernel sparse representation, together with the topological information of brain tumor structure, is utilized by the graph-cut method to make the segmentation decision. The proposed framework is an enhanced version of the one introduced in our previous work [16] by including a PCA-based splitting component, named PCA-Split, to significantly speed up the processing procedure without affecting the accuracy. Furthermore, the new framework has slightly improved results. The idea of PCA-Split is driven by the fact that manipulation of a large matrix is of high computational cost. PCA-Split replaces the original training features with more compact and representative representations. Therefore, dominant features can be efficiently preserved, though the size of the training matrix is significantly decreased and hence processing time is reduced. The proposed framework is evaluated on 20 high-grade glioma (HGG) cases provided by the multi-modal Brain Tumor Segmentation Challenges 2013 (BRATS2013). Results shows the enhanced framework achieves comparable performance compared to the state-of-the-art approaches. In addition, it generalizes better on unseen images even though less training data is required.

The remainder of this paper is organized as follows. Section 2 provides an overview of the proposed framework for automated multi-label brain tumor segmentation. PCA-Split, kernel sparse representation and the graph-cut method, which are the three main components of the proposed framework, are discussed in Section 3-5. Evaluation results and comparison with the state-of-the-art approaches are reported in Section 6. The paper is concluded in Section 7.

2 Overview

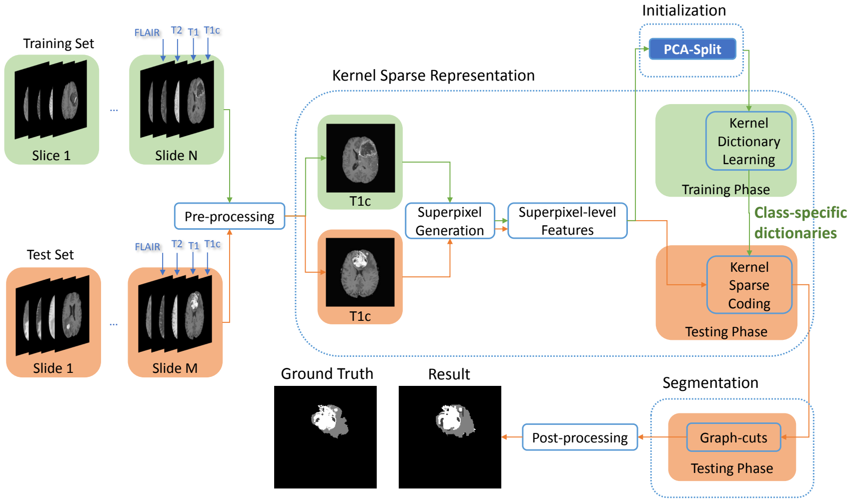

Figure 1
Overview of the proposed automated framework for multi-label brain tumor segmentation.

An overview of the proposed automated framework for multi-label brain tumor segmentation is shown in Figure 1. The proposed framework contains three main components: initialization with PCA-Split, kernel sparse representation and segmentation using graph-cuts. Given a set of training samples, PCA-Split initialization finds more compact and representative representations by splitting the set into a given number of subsets and replacing the raw representations with the centroids of each subsets. Kernel sparse representation consists of kernel dictionary learning and kernel sparse coding. In the training phase, kernel dictionary learning learns class-specific dictionaries based on superpixel-level features of brain tissues, which are used as representation systems for each task-relevant class. In the testing phase, kernel sparse coding generates optimal sparse codes for unseen testing samples according to the learned dictionaries and their superpixel-level features. The kernel sparse representation is then utilized in the graph-cut method to make pixel-wise segmentation decisions.

3 PCA-Split Initialization

Adequate and representative training samples are critical to the performance of learning-based approaches. However, manipulation of a large matrix is of high computational cost and the quality of the selected training samples is not guaranteed.

Algorithm 1 PCA-Split

```
Input: A input set $\mathbf{W} = [\mathbf{w}_i]_{i=1}^{N}$ and a desired number of subsets $Q$.
Task: Split a subset of the given input set with regard to its variance until the desired number of subsets is reached.
Initialize: Number of subsets $q = 1$, subsets $\mathbf{V} = [\mathbf{V}_1, ..., \mathbf{V}_i, ..., \mathbf{V}_q]$ and $\mathbf{V}_1 = \mathbf{W}$.
Procedure:
while $q \neq Q$ do
    for $\forall\ \mathbf{V}_i \subseteq \mathbf{V}$ do
        $\delta_i = \sum_{\{\forall j | \mathbf{w}_j \in \mathbf{V}_i\}} (\mathbf{w}_j - \boldsymbol{\mu}_i)^2$.
    end for
    Sort all subsets in descending order according to $\delta_i$.
    Calculate covariance matrix $\mathbf{\Sigma}_1 = \sum_{\{\forall j | \mathbf{w}_j \in \mathbf{V}_1\}} (\mathbf{w}_j - \boldsymbol{\mu}_1)(\mathbf{w}_j - \boldsymbol{\mu}_1)^T$
    Find out eigenvector $\mathbf{eig}_{max}$ which corresponds to the largest eigenvalue.
    for all $j \in \{\forall j | \mathbf{w}_j \in \mathbf{V}_1\}$ do
        if $\langle (\mathbf{w}_j - \boldsymbol{\mu}_1), \mathbf{eig}_{max} \rangle < 0$ then
            $\mathbf{w}_j \in \mathbf{V}_{left}$
        else if $\langle (\mathbf{w}_j - \boldsymbol{\mu}_1), \mathbf{eig}_{max} \rangle \geq 0$ then
            $\mathbf{w}_j \in \mathbf{V}_{right}$
        end if
    end for
    $q \leftarrow q + 1$
    $\mathbf{V}_{q-1} \leftarrow \mathbf{V}_{left}$
    $\mathbf{V}_q \leftarrow \mathbf{V}_{right}$
    for $\forall\ \mathbf{V}_i \subseteq \mathbf{V}$ do
        $\boldsymbol{\mu}_i = \frac{\sum_{\{\forall j | \mathbf{w}_j \in \mathbf{V}_i\}} \mathbf{w}_j}{|\mathbf{V}_i|}$
    end for
end while
Output: subsets $\mathbf{V} = [\mathbf{V}_1, ..., \mathbf{V}_i, ..., \mathbf{V}_Q]$ and centroids $\mathbf{U} = [\boldsymbol{\mu}_1, ..., \boldsymbol{\mu}_Q]$
```

To address this problem, a principal-component-analysis-based (PCA-based) splitting technique is applied, which is named PCA-Split. The PCA-based splitting technique has been utilized in various applications, like codebook initialization for vector quantization [17] and hierarchical clustering [18]. The purpose of PCA-Split is, in each iteration, to find an optimal splitting plane with respect to the variance of a subset of the given data [17]. Splitting continues until the desired number of subsets is achieved. The centroid of each subset is used to represent all data samples that lie in the subset. The main properties of the subset are preserved by the centroid, while "outliers" are eliminated. In this way, more compact and representative representations of the dataset can be obtained.

The procedure of performing PCA-Split is described as follows. Given an input set $\mathbf{W} = [\mathbf{w}_i]_{i=1}^{N}$, PCA-Split starts with only one subset $\mathbf{V}_1$ which contains the entire input set. In each iteration, all subsets are sorted in descending order according to their representation distortions calculated by the formulation $\delta_q = \sum_{\forall j|\mathbf{w}_j \in \mathbf{V}_q} (\mathbf{w}_i - \boldsymbol{\mu}_q)^2$ with respect to the centroid $\boldsymbol{\mu}_q$. The subset with the largest representation distortion is then selected to be split. The optimal splitting plane is the eigenvector corresponding to the largest eigenvalue, which splits the subset into “left” and “right” groups. Hence the number of subsets is increased by one in each iteration until the preset number of subsets is reached.

The pseudo-code for PCA-Split is given in Algorithm 1, where $\langle \cdot, \cdot \rangle$ denotes the inner product, $\mathbf{X}^T$ the transpose of $\mathbf{X}$, $|\mathbf{X}|$ the number of elements in $\mathbf{X}$.

4 Kernel Sparse Representation

4.1 Extraction and Fusion of Superpixel-Level Features

Superpixels that contain pixels with similar perceptual meaning are the basic processing units in the proposed framework. The compact grouping of pixels is beneficial to the achievement of better kernel sparse representation and faster segmentation. The contour relaxed superpixel (CRS) algorithm [19] is utilized for superixel generation due to its flexibility in controlling the adaption to a complicated contour with a single parameter κ. MR imaging provides multi-modal information, like T1-weighted (T1), T2-weighted (T2), contrast-enhanced T1-weighted (T1c) and FLAIR, which help to enrich our understanding of brain tumors. Due to their higher spatial resolution and clearer display of brain tumor structure compared to other modalities, T1c images are used as the reference in the generation of superpixels. Superpixel generation is restricted to the brain area only to avoid unnecessary processing to the background area. CRS ($\kappa = 0.01$) partitions an input image into a set of superpixels $\mathbf{S} = [s_1, ..., s_t, ...s_T]$. In order to fully utilize the multi-modal information, the generated superpixel regions are applied to T1, T2 and FLAIR modalities.

Superpixel-level features are extracted based on the generated superpixel regions (Figure 2). For a superpixel s_t, 64-bin histograms from all four modalities are calculated, which are denoted as $\mathbf{h}_{t(c)}$ $(c \in \{\text{T1}, \text{T2}, \text{T1c}, \text{FLAIR}\})$. All histograms are normalized to have $\sum_{j=1}^{r} \mathbf{h}_{t(c)}(j) = 1$, where r is the number of pixels located in superpixel s_t, to prevent bias induced by the difference in number of pixels. In addition to histograms, spatial locations of superpixels are taken into consideration. The spatial location of superpixel s_t is defined as its centroid $\mathbf{l}_t = (x_t, y_t)$. The mean values of positions of all pixels in superpixel s_t in the x-axis normalized by the width of the image and y-axis normalized by the height of the image correspond to the values of x_t and y_t respectively. Therefore, the learned dictionaries are able to simultaneously model both features including histogram and spatial location.

The proposed framework, instead of working on the raw representation of data, generates kernel sparse representation in a high-dimensional, implicit feature space $\mathscr{F}$.

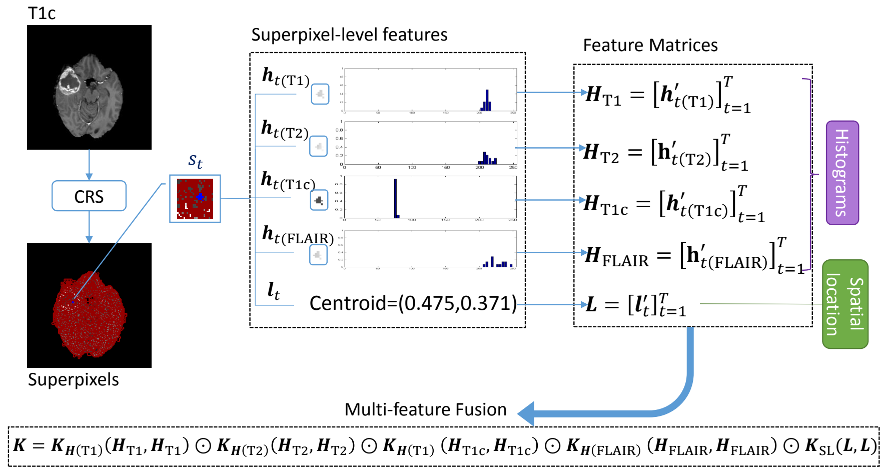
T1c
CRS
Superpixels
s_t
Superpixel-level features
$h_{t(\mathrm{T1})}$
$h_{t(\mathrm{T2})}$
$h_{t(\mathrm{T1c})}$
$h_{t(\mathrm{FLAIR})}$
l_t
Centroid=(0.475,0.371)
Feature Matrices
$H_{\mathrm{T1}} = [h'_{t(\mathrm{T1})}]_{t=1}^{T}$
$H_{\mathrm{T2}} = [\mathbf{h}'_{t(\mathrm{T2})}]_{t=1}^{T}$
$H_{\mathrm{T1c}} = [h'_{t(\mathrm{T1c})}]_{t=1}^{T}$
$H_{\mathrm{FLAIR}} = [\mathbf{h}'_{t(\mathrm{FLAIR})}]_{t=1}^{T}$
$L = [l'_t]_{t=1}^{T}$
Histograms
Spatial location
Multi-feature Fusion
$K = K_{H(\mathrm{T1})}(H_{\mathrm{T1}}, H_{\mathrm{T1}}) \odot K_{H(\mathrm{T2})}(H_{\mathrm{T2}}, H_{\mathrm{T2}}) \odot K_{H(\mathrm{T1})}\,(H_{\mathrm{T1c}}, H_{\mathrm{T1c}}) \odot K_{H(\mathrm{FLAIR})}\,(H_{\mathrm{FLAIR}}, H_{\mathrm{FLAIR}}) \odot K_{\mathrm{SL}}(L, L)$

Figure 2
Extraction and fusion of superpixel-level features.

Nonlinear similarities in $\mathscr{F}$ between samples are considered, which are more discriminative compared to the linear similarity in the original space. In order to map the raw representation to the feature space $\mathscr{F}$, a nonlinear transformation $\Phi(\cdot)$ is applied. Hence, nonlinear similarity between two samples $\mathbf{x}$ and $\mathbf{x}'$ can be measured by the inner product $\Phi(\mathbf{x})^T\Phi(\mathbf{x}')$. Nevertheless, $\Phi(\cdot)$ can be intractable in the high-dimensional, even infinite-dimensional, feature space $\mathscr{F}$ [14]. To address this problem, the kernel trick is adopted, which replaces the intractable inner product $\Phi(\mathbf{x})^T\Phi(\mathbf{x}')$ with a known kernel function $\mathscr{K}$. With the knowledge of the kernel and the samples, nonlinear similarity can always be calculated even though the explicit formulation of $\Phi(\cdot)$ is not known. To proceed with the replacement, the chosen kernel function should satisfy Mercer's theorem [20]. The well-known radial basis function (RBF) kernel is selected in our framework. The definition of the RBF Kernel is $\mathscr{K}(\mathbf{x},\mathbf{y}) = \exp(-\|\mathbf{x}-\mathbf{y}\|^2/2\sigma^2)(\sigma = 1.5)$.

Given two matrices $\mathbf{X} = [\mathbf{x}_i]_{i=1}^{N}$ and $\mathbf{X}' = [\mathbf{x}'_i]_{i=1}^{M}$, a Gramian matrix $\mathbf{K}(\mathbf{X},\mathbf{X}') \in \mathbb{R}^{N\times M}$ is defined such that its (n,m)-entry $\mathbf{K}_{n,m}$ corresponds to the nonlinear similarity $\mathscr{K}(\mathbf{x}_n,\mathbf{x}'_m)$ between the n^{th} element of $\mathbf{X}$ and the m^{th} element of $\mathbf{X}'$. All extracted superpixel-level features are arranged in column vector manner into their corresponding feature matrices (Figure 2). Specifically, in the training phase, the raw representations of all features are substituted by the centroids of subsets obtained by applying PCA-Splits to their corresponding feature matrices with a specified number of substes Q. For histogram feature matrices, Gramian matrices $\mathbf{K}_{H(c)}$ $(c \in \{\mathrm{T1},\mathrm{T2},\mathrm{T1c},\mathrm{FLAIR}\})$ are obtained to represent the nonlinear similarities in a specific modality, while a Gramian matrix $\mathbf{K}_{SL}$ is calculated for that of spatial location. $\mathbf{K}_{H(c)}$ $(c \in \{\mathrm{T1},\mathrm{T2},\mathrm{T1c},\mathrm{FLAIR}\})$ and $\mathbf{K}_{SL}$ are denoted as the following formulations:

$$\begin{aligned} \mathbf{K}_{H(c)}(i,j) &= \exp\left(-\frac{||\mathbf{h}_{i(c)} - \mathbf{h}_{j(c)}||_2^2}{2\sigma^2} \right) \\ \mathbf{K}_{\text{SL}}(i,j) &= \exp\left(-\frac{||\mathbf{l}_i - \mathbf{l}_j||_2^2}{2\sigma^2} \right) \end{aligned} \tag{1}$$

Not only the sparse representation benefits from the kernel trick, the use of the the kernel trick also facilitate the fusion of multi-features such that all the Gramian matrices can be combined in an elegant way by simple Hadamard product. The combination yields an ensemble matrix $\mathbf{K}$, i.e., $\mathbf{K} = \mathbf{K}_{H(\text{T1})} \odot \mathbf{K}_{H(\text{T2})} \odot \mathbf{K}_{H(\text{T1c})} \odot \mathbf{K}_{H(\text{FLAIR})}$. Learning of dictionary based on the ensemble Gramian matrix is more efficient and effective since all five features are captured at one time. For simplicity, the rest of the paper only focuses on the ensemble Gramian matrix for the generation of kernel sparse representation, rather than the five Gramian matrices individually.

4.2 Kernel Sparse Coding and Kernel Dictionary Learning

Given a set of input data $\mathbf{Y} = [\mathbf{y}_i]_{i=1}^N, \mathbf{y}_i \in \mathbb{R}^M$, the goal of dictionary learning is to obtain an optimal overcomplete dictionary $\mathbf{D} \in \mathbb{R}^{M \times K}$ to well model the given data $\mathbf{Y}$, so that each element $\mathbf{y}_i \in \mathbf{Y}$ can be approximated by a linear combination of only a few dictionary atoms $\mathbf{d}_k, (k = 1, 2, ..., K)$ via a code $\mathbf{x}_i \in \mathbb{R}^K$. The code $\mathbf{x}_i$ is sparse since only a few entries are non-zero. The objective function of dictionary learning is given by:

$$(\hat{\mathbf{X}}, \hat{\mathbf{D}}) = \arg\min_{\mathbf{X,D}} ||\mathbf{Y} - \mathbf{D}\mathbf{X}||_F^2 \ s.t. \ ||\mathbf{x}_i||_0 \leq T_0, \forall i \tag{2}$$

where $\mathbf{X} = [\mathbf{x}_i]_{i=1}^N$, $||.||_F$ is the Frobenius norm, $||.||_0$ denotes the ℓ_0 norm and T_0 the sparsity level, which indicates the maximum number of non-zero entries in a sparse code $\mathbf{x}_i$.

Upon obtaining the dictionary, $\mathbf{D}$ is fixed and sparse coding finds the optimal sparse representation $\mathbf{X}'$ for the testing data $\mathbf{Y}'$ based on the learned dictionary $\mathbf{D}$. The optimization problem of sparse coding is expressed as:

$$(\hat{\mathbf{X}}') = \arg\min_{\mathbf{X}'} ||\mathbf{Y}' - \mathbf{D}\mathbf{X}'||_F^2 \ s.t. \ ||\mathbf{x}_i'||_0 \leq T_0, \forall i \tag{3}$$

To adapt the original optimization problem of sparse coding and dictionary learning into feature space $\mathscr{F}$, a nonlinear transformation $\Phi(\cdot)$ is applied to both the data matrix. Therefore, the kernel extensions of dictionary learning and sparse coding are formulated as equations (4) and (5) respectively:

$$(\hat{\mathbf{X}}, \hat{\mathbf{D}}) = \arg\min_{\mathbf{X,D}} ||\Phi(\mathbf{Y}) - \Phi(\mathbf{D})\mathbf{X}||_F^2 \ s.t. \ ||\mathbf{x}_i||_0 \leq T_0, \forall i \tag{4}$$

$$(\hat{\mathbf{X}}') = \arg\min_{\mathbf{X}'} ||\Phi(\mathbf{Y}') - \Phi(\mathbf{D})\mathbf{X}'||_F^2 \ s.t. \ ||\mathbf{x}_i'||_0 \leq T_0, \forall i \tag{5}$$

where $\Phi(\mathbf{Y}) = [\Phi(\mathbf{y}_i)]_{i=1}^N$, $\Phi(\mathbf{Y}') = [\Phi(\mathbf{y}'_i)]_{i=1}^P$ and $\Phi(\mathbf{D}) = [\Phi(\mathbf{d}_i)]_{i=1}^K$.

The dictionary in $\mathscr{F}$ can be represented by the linear combination of the input data (i.e., $\Phi(\mathbf{D}) = \Phi(\mathbf{Y})\mathbf{A}$), since all dictionary atoms lie in the linear span of the input data [12]. $\mathbf{A} \in \mathbb{R}^{N\times K}$ is an atom representation dictionary and the optimal $\mathbf{A}$ is directly related to the best dictionary $\mathbf{D}$ that can be achieved. The formulation of kernel dictionary learning and kernel sparse coding can be re-written as equations (6) and (7) respectively:

$$(\hat{\mathbf{X}}, \hat{\mathbf{A}}) = \arg\min_{\mathbf{X,A}} \|\Phi(\mathbf{Y}) - \Phi(\mathbf{Y})\mathbf{A}\mathbf{X}\|_F^2 \ s.t. \ \|\mathbf{x}_i\|_0 \le T_0, \forall i \tag{6}$$

$$(\hat{\mathbf{X}'}) = \arg\min_{\mathbf{X}'} \|\Phi(\mathbf{Y}') - \Phi(\mathbf{Y})\mathbf{A}\mathbf{X}'\|_F^2 \ s.t. \ \|\mathbf{x}'_i\|_0 \le T_0, \forall i \tag{7}$$

A kernel extension of the K-SVD type dictionary learning algorithm [12] is adopted in our framework. Since learning of dictionary iteratively alternates between kernel sparse coding and kernel dictionary learning until predefined criteria are met or maximum iteration number is reached, we only focus on the optimization of kernel dictionary learning (i.e., equation (6)) for simplicity.

In the kernel sparse coding step, the atom representation dictionary $\mathbf{A}$ is assumed to be known and fixed. The sparse codes matrix $\mathbf{X}$ can be found by minimizing the approximation error $\|\Phi(\mathbf{Y}) - \Phi(\mathbf{Y})\mathbf{A}\mathbf{X}\|_F^2$ subject to the sparsity constraint $\|\mathbf{x}_i\|_0 \le T_0, \forall i$. The penalty term can be decomposed and written as:

$$\|\Phi(\mathbf{Y}) - \Phi(\mathbf{Y})\mathbf{A}\mathbf{X}\|_F^2 = \sum_{i=1}^N \|\Phi(\mathbf{y}_i) - \Phi(\mathbf{Y})\mathbf{A}\mathbf{x}_i\|_2^2 \tag{8}$$

Now, the "big" problem is separated into N "small" optimization problems:

$$\min_{\mathbf{x}_i} \|\Phi(\mathbf{y}_i) - \Phi(\mathbf{Y})\mathbf{A}\mathbf{x}_i\|_2^2 \ s.t. \ \|\mathbf{x}_i\|_0 \le T_0 \tag{9}$$

To facilitate optimization, the objective function is reconstructed with kernel function $\mathscr{K}$ to avoid the unknown nonlinear transformation $\Phi(\cdot)$:

$$\min_{\mathbf{x}_i} \mathscr{K}(\mathbf{y}_i, \mathbf{y}_i) - 2\mathbf{K}(\mathbf{y}_i, \mathbf{Y})\mathbf{A}\mathbf{x}_i + \mathbf{x}_i^T\mathbf{A}^T\mathbf{K}(\mathbf{Y,Y})\mathbf{A}\mathbf{x}_i \ s.t. \ \|\mathbf{x}_i\|_0 \le T_0 \tag{10}$$

With the help of the kernel trick, this optimization problem can be solved by the classic orthogonal matching pursuit (OMP) algorithm [21].

Once the sparse codes matrix is calculated, we update the all dictionary atoms according to the projection error. In other words, kernel dictionary learning, with the fixed $\mathbf{X}$, searches for a new atom representation dictionary $\mathbf{A}$ to minimize $\|\Phi(\mathbf{Y}) - \Phi(\mathbf{Y})\mathbf{A}\mathbf{X}\|_F^2$.

First, the penalty term is rewritten as:

$$\|\Phi(\mathbf{Y}) - \Phi(\mathbf{Y})\sum_{j=1}^K \mathbf{a}_j\mathbf{x}_j^R\|_F^2 = \|\Phi(\mathbf{Y})(\mathbf{I} - \sum_{j\neq k} \mathbf{a}_j\mathbf{x}_j^R) - \Phi(\mathbf{Y})(\mathbf{a}_k\mathbf{x}_k^R)\|_F^2 \tag{11}$$

where $\mathbf{a}_k$ and $\mathbf{x}_k^R$ correspond to the k^{th} column of $\mathbf{A}$ and the k^{th} row of $\mathbf{X}$ respectively. Contribution made by the k^{th} dictionary atom to the estimated sample can be obtained from $\mathbf{a}_k\mathbf{x}_k^R$. For simplicity, we denote $\mathbf{E}_k = \mathbf{I} - \sum_{j\neq k}\mathbf{a}_j\mathbf{x}_j^R$, which represents the approximation error between the estimated and original samples when the k^{th} dictionary atom is removed.

As can be seen in equation (11), the pair of unknown variables $(\mathbf{a}_k, \mathbf{x}_k^R)$ is expected to be found to minimize the approximation error. This can be solve by the best rank-1 approximation. Due to their trivial contribution to the optimization problem, columns related to zero entries of $\mathbf{x}_k^R$ in $\mathbf{E}_k$ and $\mathbf{a}_k\mathbf{x}_k$ are removed, which yields $\mathbf{E}_k^{Re}$ and $\mathbf{a}_k\mathbf{x}_k^{Re}$ respectively ($\mathbf{x}_k^{Re}$ containing only non-zero weights of $\mathbf{x}_k^R$). Singular value decomposition (SVD) is applied to $\mathbf{E}_k^{Re}$ and $\mathbf{a}_k\mathbf{x}_k^{Re}$ instead of $\mathbf{E}_k$ and $\mathbf{a}_k\mathbf{x}_k^R$ to preserve the specified sparsity level and reduce computational cost.

The SVD decomposes $\Phi(\mathbf{Y})\mathbf{E}_k^{Re}$ into three parts:

$$\Phi(\mathbf{Y})\mathbf{E}_k^{Re} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T \tag{12}$$

Equating $\Phi(\mathbf{Y})\mathbf{a}_k\mathbf{x}_k^{Re}$ to the rank-1 matrix, which corresponds to the largest singular value $\sigma_1 = \mathbf{\Sigma}(1,1)$ of $\Phi(\mathbf{Y})\mathbf{E}_k^{Re}$, gives the solution to the best rank-1 approximation.

$$\Phi(\mathbf{Y})\mathbf{a}_k\mathbf{x}_k^{Re} = \mathbf{u}_1\sigma_1\mathbf{v}_1^T \tag{13}$$

where $\mathbf{u}_1$ and $\mathbf{v}_1$ are the first columns of $\mathbf{U}$ and $\mathbf{V}$ corresponding to σ_1 respectively. Thus, the solution can be calculated from the equations below:

$$\begin{aligned} \Phi(\mathbf{Y})\mathbf{a}_k &= \mathbf{u}_1 \\ \mathbf{x}_k^{Re} &= \sigma_1\mathbf{v}_1^T \end{aligned} \tag{14}$$

However, it is impractical to perform SVD on $\Phi(\mathbf{Y})\mathbf{E}_k^{Re}$ since the explicit formulation of $\Phi(\cdot)$ is unknown. Consequently, the kernel trick should be used again such that the eigen decomposition of $\mathbf{E}_k^{ReT}\Phi(\mathbf{Y})^T\Phi(\mathbf{Y})\mathbf{E}_k^{Re}$, which is $\mathbf{V}\mathbf{\Delta}\mathbf{V}^T$, is calculated to infer the unknown variables. As a result, $\mathbf{V}$ is obtained and σ_1 can be deduced by $\sigma_1 = \sqrt{\mathbf{\Delta}(1,1)}$. An analytical solution is possible when the term for σ_1 is substituted into equation (14):

$$\mathbf{a}_k = \sigma_1^{-1}\mathbf{E}_k^{Re}\mathbf{v}_1 \tag{15}$$

In each iteration, all the atoms of $\mathbf{A}$ are updated according to the manner stated above followed by the search for new sparse codes based on the new dictionary. This process alternates between kernel dictionary learning and kernel dictionary learning till some preset conditions are satisfied.

5 Graph-Cuts

The pixel-wise segmentation decision is made by the graph-cut method based on both kernel sparse representation and topological information of the brain structures. The task requires the proposed framework to classify pixels into five specific classes, which are non-tumor (label=0), necrotic core (label=1), edema (label=2), non-enhancing core (label=3) and enhancing core (label=4). For each class, a dictionary is learned by applying kernel dictionary learning to a set of training samples as described in Section 4. These dictionaries should be able to model their own classes well since they are optimized for the particular purpose, even though they fail to approximate well the rest of the classes.

For a test superpixel s_t, the proposed framework computes five sparse codes $\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3$ and $\mathbf{x}_4$ with respect to the five dictionaries. The approximation errors between the input sample s_t and the five approximations are denoted by $e_0^{s_t}, e_1^{s_t}, e_2^{s_t}, e_3^{s_t}$ and $e_4^{s_t}$, and measured by:

$$e_i^{s_t} = \|\Phi(\mathbf{y}^{s_t}) - \Phi(\mathbf{D}_i)\mathbf{x}_i^{s_i}\|_2^2,\ i = 0,1,2,3,4 \tag{16}$$

Segmentation based on kernel sparse representation does not take topological information of the brain structure into consideration. The graph-cut method, which naturally incorporates topological information, is a possible remedy. We propose a variant graph-cuts [22, 23] to better adapt to our application. A graph should be constructed to proceed with the variant graph-cuts. To facilitate graph construction, a superpixel is first ungrouped into a set of pixels which form the superpixel. Then these pixels are given the same approximation errors as the superpixel they belong. The image is represented by a array which contains all its pixels $\mathbf{z} = (z_1, ..., z_l ..., z_L)$, assuming there are L pixels in total. The approximation errors assigned to pixel z_l are denoted by $e_i^{z_l} (i = 0,1,2,3,4)$. These pixels, besides the approximation errors, contains extra information in terms of different gray-level intensities in multimodalities. For pixel z_l, gray-level intensities in the four modalities are defined as $g_{\mathrm{T1}}^{z_l}, g_{\mathrm{T2}}^{z_l}, g_{\mathrm{T1c}}^{z_l}$ and $g_{\mathrm{FLAIR}}^{z_l}$.

The energy function of graph-cuts is expressed by:

$$E(f) = \sum_{\{p,g\} \in \mathscr{N}} V_{p,q}(f_p, f_q) + \sum_{p \in \mathscr{P}} D_p(f_p) \tag{17}$$

where f is a label in a finite label set $\mathscr{L}$, $\{p,q\}$ a pair of pixels in the pixel set $\mathscr{P}$, and $\mathscr{N}$ a set of neighboring pixels. The first term in equation (17) is known as the smoothness term, which encourages pairwise smoothness while preserving label discontinuity on boundaries. The data term is the name given to the second term, which measures the fit of label f to the observed data p.

Typically, the data term is formulated with negative log-likelihood. According to the previous discussion, if a test sample belongs a specific class, the smallest approximation error can be achieve when the dictionary learned for this class is used in kernel sparse coding. Therefore, the kernel sparse representation generated in the previous step is utilized in the data term as the measurement of label appropriateness as shown below:

$$\sum_{l=1}^{L} D_{z_l}(f_{z_l}) = \sum_{l=1}^{L} log(e_{f_{z_l}}^{z_l}) \tag{18}$$

The smoothness term is defined as:

$$\sum_{\{z_l,z_q\}\in\mathcal{N}} V_{z_l,z_q}(f_{z_l},f_{z_q}) = \theta \sum_{\{z_l,z_q\}\in\mathcal{N}_4} [f_{z_l} \neq f_{z_q}] \exp -\beta\|z_l - z_q\|_2^2 \tag{19}$$

where θ is a constant controlling the degree of discontinuity preserving, $\mathcal{N}_4$ indicates 4-way connectivity and $[\cdot]$ is a indicator function taking value 1 for true prediction or 0 for false prediction. θ is empirically set to 50 according to the preliminary experiments. The Euclidean distance between pixel z_l and z_q is given by:

$$\|z_l - z_q\|_2^2 = \sum_c (g_c^{z_l} - g_c^{z_q})^2,\ c \in \{\mathrm{T1},\mathrm{T2},\mathrm{T1c},\mathrm{FLAIR}\} \tag{20}$$

Though θ only has the control on overall smoothness, we have another parameter β to prevent the tendency of being over-smooth on boundaries between different classes. β is computed by:

$$\beta = (2 < \|z_l - z_q\|_2 >)^{-1} \tag{21}$$

where $< \cdot >$ denotes expectation over $\mathcal{N}_4$ neighborhood.

The optimization of the variant graph-cuts, depending on nonlinear feature similarity and topological information, provides the best label configurations for all pixels. We use the GCMex - MATLAB wrapper to implement the proposed variant graph-cuts [23, 24, 25].

6 Experiment and Discussion

The proposed framework is evaluated on 20 real HGG cases in the training set of BRATS2013 with two-fold cross validation (CV). In the training phase, the superpixels collected from the training set for each of the five classes (i.e., non-tumor(0), necrotic core(1), edema(2), non-enhancing core(3) and enhancing-core(4)) are initialized for kernel dictionary learning by PCA-Split. The desired number of subsets Q is empirically set to 512 considering the trade-off between good segmentation result and less processing time. As a result, the dictionaries of the five task-relevant classes are learned from their corresponding 512 PCA-Split centroids. For kernel dictionary learning and kernel sparse coding, we fix the number of dictionary atoms to 200 and the sparsity level to 5. The framework is implemented on MATLAB using a computer with Intel processor (i7-3930K, 3.20GHz) and 32GB of RAM.

The following three regions are segmented and used for evaluation:

- Region 1: complete tumor (label 1+2+3+4)
- Region 2: tumor core (label 1+3+4)
- Region 3: enhancing core (label 4)

The performance of the proposed framework is reported via the Dice similarity coefficient, Jaccard index and sensitivity [3] on the aforementioned three regions.

Even though the BRATS2013 dataset has been pre-processed with skull-striping and co-registration, obvious intensity bias can still be observed. The intensity bias can significantly worsen the segmentation accuracy since superpixel-level histograms are intensively used in our framework. This requires further pre-processing steps including bias field correction and intensity inhomogeneity correction. T2 and FLAIR are exempted from bias field correction due to the fact that the correction decreases their contrast. N4ITK [26] tool in Slicer3D is used for bias field correction, while intensity inhomogeneity is adjusted by a learning-based two-step standardization [27].

The segmentation result output directly from graph-cuts can be noisy. Therefore, binary morphological processing and connected component analysis are applied as post-processing steps.

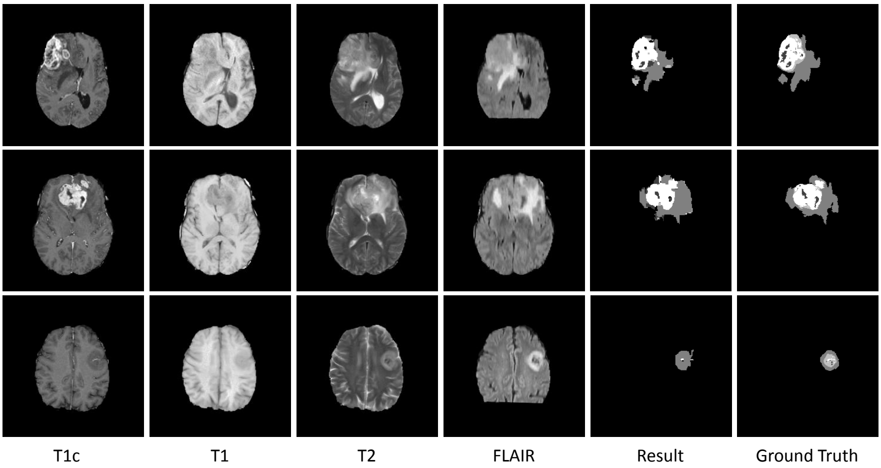

Figure 3
Three segmentation examples of the proposed framework. First column to sixth column correspond to T1c images, T1 images, T2 images, FLAIR images, segmentation results and ground truths respectively. The first row shows one slice of patient009, while the second row is a slice of patient015. The bottom row demonstrates the performance of the proposed framework on the worst case-patine012.

Several segmentation examples generated by the proposed approach are shown in Figure 3. In addition, we report the averages and standard deviations of the Dice similarity coefficient, Jaccard index and sensitivity that achieved by the proposed framework in Table 1. The performance of our previous method [16] is also concluded in Table 1. For Region 2 and Region 3, we report the performance twice, one including patient012 while the other excluding patient012, since the peculiarity of patient012 significantly worsens the overall performance as can be seen from Table 1. The reason why both our frameworks fail to give good segmentation results for patient012 is probably because of the similar intensities shared by the non-enhancing core and the edeme in all four modalites. Moreover, the tumor of patient012 mainly consists of non-enhancing core and edema, which makes it extremely difficult for our approaches to make good segmentation decision. Hence,

Table 1
Evaluation of performance on BRATS 2013 training cases (HGG)

	Dice				Jaccard				Sensitivity			
	previous		proposed		previous		proposed		previous		proposed	
	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std
Region 1	81.1	9.3	81.1	9.6	69.2	12.5	69.1	12.9	81.9	13.6	82.3	14.1
Region 2	62.9	17.6	63.3	22.1	48.0	17.3	49.5	21.0	69.3	22.4	71.1	25.2
Region 2(*)	65.3	14.2	66.5	17.1	50.0	15.2	52.0	18.2	72.4	18.2	74.8	19.6
Region 3	69.7	17.2	70.4	19.6	55.6	17.8	57.1	19.6	70.1	22.5	71.2	24.5
Region 3(*)	71.9	14.4	73.4	14.8	57.7	15.5	59.7	16.2	72.9	19.2	74.5	20.0

* denotes the scores are calculated excluding the result of patient0012.

the proposed framework easily mistakes the non-enhancing core for the edema and results in very low scores in both Region 2 and Region 3. The average processing time for one slice required by our previous framework and the proposed framework are 8 seconds and 30 seconds. The comparison between our previous framework and the proposed framework in terms of performance (Table 1) and processing time clearly reveals the advantages of the proposed framework over the previous one. The proposed framework, with exactly the same training and test set configuration, achieves comparable scores in Region 1 and slightly outperforms the previous one in both Region 2 and Region 3. The proposed framework (8 seconds) requires less than one third of the average execution time for one slice of the previous method (30 seconds).

We also show in Table 2 the performances of three state-of-the-art discriminative approaches [28, 29, 30] evaluated on the same dataset. Scores are directly extracted from their published papers. This table is for reference only due to the lack of their training and testing set configurations. Nevertheless, we can conclude that our proposed approaches achieves competitive performance compared to the state-of-the-art approaches. In addition, better generalization ability of the proposed framework is observed when we compare the CV type used in our framework to those in their approaches (Table 3). This means, the proposed framework achieves comparable performance with much less training cases, but still perform well on more unseen images.

Conclusions

A novel automated framework for multi-label brain tumor segmentation is proposed in this paper. As an enhanced version of our previous framework in [16], the proposed framework has advantages in both performance and processing time. PCA-Split initialization provides compact and representative training samples for kernel dictionary learning, which significantly reduce training and processing time without scarifying good models for related classes. Kernel sparse representation based on kernel dictionary learning and kernel sparse coding is utilized in the graph-cut method together with the topological information of brain structure to arrive at a segmentation decision. The results show that the proposed framework gives a comparable performance while better generalization ability is observed when compared to the state-of-the-art discriminative approaches.

We plan to include topological information in the generation of sparse representation as an extra regularization term, instead of optimizing sparse representation and graph-cuts separately, such that jointly optimization can be achieved and hence better sparse representation and result are expected.

Table 2
Performance of state-of-the-art methods

	Dice						Jaccard						Sensitivity					
	Approach 1		Approach 2		Approach 3		Approach 1		Approach 2		Approach 3		Approach 1		Approach 2		Approach 3	
	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std
Region 1	73.4	—	79.0	17.0	80.2	12.4	59.8	—	—	—	68.4	15.3	85.7	—	—	—	85.9	12.6
Region 2	60.8	—	60.0	26.0	69.1	22.0	48.5	—	—	—	56.1	21.2	67.8	—	—	—	71.9	26.2
Region 3	63.5	—	53.0	25.0	69.8	24.7	50.8	—	—	—	57.8	23.2	66.8	—	—	—	68.0	27.0

Table 3
CV Type used in the state-of-the-art approaches

	Approach 1	Approach 2	Approach 3
CV Type	20-fold	Leave-one-out	5-fold

Approach 1 is proposed by Buendia et al.[28], Approach 2 is proposed by Cordier et al.[29] and Approach 3 is proposed by Meier et al.[30]

References

[1] Stefan Bauer, Roland Wiest, Lutz-P Nolte, and Mauricio Reyes. A survey of MRI-based medical image analysis for brain tumor studies. *Physics in Medicine and Biology*, 58(13):R97, 2013.

[2] Nelly Gordillo, Eduard Montseny, and Pilar Sobrevilla. State of the art survey on MRI brain tumor segmentation. *Magnetic Resonance Imaging*, 31(8):1426–1438, 2013.

[3] Bjoern H Menze, Andras Jakab, Stefan Bauer, Jayashree Kalpathy-Cramer, Keyvan Farahani, Justin Kirby, Yuliya Burren, Nicole Porz, Johannes Slotboom, Roland Wiest, et al. The multimodal brain tumor image segmentation benchmark (BRATS). *IEEE Transactions on Medical Imaging*, 34(10):1993–2024, 2015.

[4] Marcel Prastawa, Elizabeth Bullitt, Sean Ho, and Guido Gerig. A brain tumor segmentation framework based on outlier detection. *Medical Image Analysis*, 8(3):275–283, 2004.

[5] Bjoern H Menze, Koen Van Leemput, Danial Lashkari, Marc-André Weber, Nicholas Ayache, and Polina Golland. A generative model for brain tumor segmentation in multi-modal images. In *Proceedings of International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI 2010)*, pages 151–159. 2010.

[6] Hassan Khotanlou, Olivier Colliot, Jamal Atif, and Isabelle Bloch. 3D brain tumor segmentation in MRI using fuzzy classification, symmetry analysis and spatially constrained deformable models. *Fuzzy Sets and Systems*, 160(10):1457–1473, 2009.

[7] Ali Gooya, Kilian M Pohl, Michel Bilello, Luigi Cirillo, George Biros, Elias R Melhem, and Christos Davatzikos. GLISTR: glioma image segmentation and registration. *IEEE Transactions on Medical Imaging*, 31(10):1941–1954, 2012.

[8] Aaron E Lefohn, Joshua E Cates, and Ross T Whitaker. Interactive, GPU-based level sets for 3D segmentation. In *Proceedings of International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI 2003)*, pages 564–572, 2003.

[9] Sean Ho, Lizabeth Bullitt, and Guido Gerig. Level-set evolution with region competition: automatic 3-D segmentation of brain tumors. In *Proceedings of IEEE International Conference on Pattern Recognition*, volume 1, pages 532–535, 2002.

[10] Ezequiel Geremia, Bjoern H Menze, Nicholas Ayache, et al. Spatial decision forests for glioma segmentation in multi-channel MR images. In *the MICCAI Challenge on Multimodal Brain Tumor Image Segmentation (BRATS 2012)*, volume 34, pages 14–17, 2012.

[11] S Reza and KM Iftekharuddin. Multi-class abnormal brain tissue segmentation using texture. In *the MICCAI Challenge on Multimodal Brain Tumor Image Segmentation (BRATS 2013)*, pages 38–42, 2013.

[12] Hien Nguyen, Vishal M Patel, Nasser M Nasrabadi, and Rama Chellappa. Kernel dictionary learning. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2012)*, pages 2021–2024, 2012.

[13] Jinjun Wang, Jianchao Yang, Kai Yu, Fengjun Lv, Thomas Huang, and Yihong Gong. Locality-constrained linear coding for image classification. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2012)*, pages 3360–3367, 2010.

[14] Jayaraman J Thiagarajan, Karthikeyan Natesan Ramamurthy, et al. Kernel sparse models for automated tumor segmentation. *International Journal on Artificial Intelligence Tools*, 23(3), 2014.

[15] Jeon Lee, Seung-Jun Kim, Rong Chen, and Edward H Herskovits. Brain tumor image segmentation using kernel dictionary learning. In *Proceedings of IEEE International Conference on Engineering in Medicine and Biology Society (EMBC 2015)*, pages 658–661. IEEE, 2015.

[16] Xuan Chen, Binh P. Nguyen, Chee-Kong Chui, and Sim-Heng Ong. Automated brain tumor segmentation using kernel dictionary learning and superpixel-level features. In *Proceedings IEEE International Conference on Systems, Man, and Cybernetics (SMC 2016)*, page [to appear], Budapest, Hungary, 9–12 Oct 2016. IEEE.

[17] Jens Schneider and Rüdiger Westermann. Compression domain volume rendering. In *Proceedings of IEEE International Conference on Visualization (VIS 2003)*, pages 293–300, 2003.

[18] Mark Pauly, Markus Gross, and Leif P Kobbelt. Efficient simplification of point-sampled surfaces. In *Proceedings of IEEE International Conference on Visualization (VIS 2002)*, pages 163–170, 2002.

[19] Christian Conrad, Matthias Mertz, and Rudolf Mester. Contour-relaxed superpixels. In *International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition*, volume 8081, pages 280–293, 2013.

[20] Nello Cristianini and John Shawe-Taylor. *An introduction to support vector machines and other kernel-based learning methods*. Cambridge University Press, 2000.

[21] Yagyensh Chandra Pati, Ramin Rezaiifar, and PS Krishnaprasad. Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition. In *Proceedings of The Twenty-Seventh Asilomar Conference on Signals, Systems and Computers*, pages 40–44, 1993.

[22] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. Grabcut: Interactive foreground extraction using iterated graph cuts. In *ACM Transactions on Graphics (TOG)*, volume 23, pages 309–314, 2004.

[23] Yuri Boykov, Olga Veksler, and Ramin Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(11):1222–1239, 2001.

[24] Vladimir Kolmogorov and Ramin Zabih. What energy functions can be minimized via graph cuts? *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(2):147–159, 2004.

[25] Yuri Boykov and Vladimir Kolmogorov. An experimental comparison of mincut/max-flow algorithms for energy minimization in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(9):1124–1137, 2004.

[26] Nicholas J Tustison, Brian B Avants, et al. N4ITK: improved N3 bias correction. *IEEE Transactions on Medical Imaging*, 29(6):1310–1320, 2010.

[27] László G Nyúl, Jayaram K Udupa, and Xuan Zhang. New variants of a method of MRI scale standardization. *IEEE Transactions on Medical Imaging*, 19(2):143–150, 2000.

[28] Patricia Buendia, Thomas Taylor, Michael Ryan, and Nigel John. A grouping artificial immune network for segmentation of tumor images. In *the MICCAI Challenge on Multimodal Brain Tumor Image Segmentation (BRATS 2013)*, pages 1–5, 2013.

[29] Nicolas Cordier, Bjoern Menze, Hervé Delingette, and Nicholas Ayache. Patch-based segmentation of brain tissues. In *the MICCAI Challenge on Multimodal Brain Tumor Image Segmentation (BRATS 2013)*, pages 6–17, 2013.

[30] Raphael Meier, Stefan Bauer, Johannes Slotboom, Roland Wiest, and Mauricio Reyes. A hybrid model for multimodal brain tumor segmentation. In *the MICCAI Challenge on Multimodal Brain Tumor Image Segmentation (BRATS 2013)*, pages 31–37, 2013.