HHS Public Access

Author manuscript

IEEE Trans Biomed Eng. Author manuscript; available in PMC 2018 August 01.

Published in final edited form as:

IEEE Trans Biomed Eng. 2018 August ; 65(8): 1871–1884. doi:10.1109/TBME.2017.2783305.

Learning Based Segmentation of CT Brain Images: Application to Post-Operative Hydrocephalic Scans

Venkateswararao Cherukuri[1,2], **Peter Ssenyonga**[4], **Benjamin C. Warf**[4,5], **Abhaya V. Kulkarni**[6], **Vishal Monga**[1,†], and **Steven J. Schiff**[2,3,†]

[1]Dept. of Electrical Engineering, The Pennsylvania State University, University Park, USA [2]Center for Neural Engineering, The Pennsylvania State University, University Park, USA [3]Dept. Neurosurgery, Engineering Science and Mechanics, and Physics, The Pennsylvania State University, University Park, USA [4]CURE Children's Hospital of Uganda, Mbale, Uganda [5]Department of Neurosurgery, Boston Children's Hospital and Department of Global Health and Social Medicine, Harvard Medical School, Boston, Massachusetts [6]Division of Neurosurgery, Hospital for Sick Children, University of Toronto, Toronto, ON, Canada

Abstract

Objective—Hydrocephalus is a medical condition in which there is an abnormal accumulation of cerebrospinal fluid (CSF) in the brain. Segmentation of brain imagery into brain tissue and CSF (before and after surgery, i.e. pre-op vs. post-op) plays a crucial role in evaluating surgical treatment. Segmentation of pre-op images is often a relatively straightforward problem and has been well researched. However, segmenting post-operative (post-op) computational tomographic (CT)-scans becomes more challenging due to distorted anatomy and subdural hematoma collections pressing on the brain. Most intensity and feature based segmentation methods fail to separate subdurals from brain and CSF as subdural geometry varies greatly across different patients and their intensity varies with time. We combat this problem by a learning approach that treats segmentation as supervised classification at the pixel level, i.e. a training set of CT scans with labeled pixel identities is employed.

Methods—Our contributions include: 1.) a dictionary learning framework that learns class (segment) specific dictionaries that can efficiently represent test samples from the same class while poorly represent corresponding samples from other classes, 2.) quantification of associated computation and memory footprint, and 3.) a customized training and test procedure for segmenting post-op hydrocephalic CT images.

Results—Experiments performed on infant CT brain images acquired from the CURE Children's Hospital of Uganda reveal the success of our method against the state-of-the-art alternatives. We also demonstrate that the proposed algorithm is computationally less burdensome and exhibits a graceful degradation against number of training samples, enhancing its deployment potential.

[†]Contributed equally.

[*]This work is supported by NIH Grant number R01HD085853

Index Terms

CT Image Segmentation; Dictionary Learning; neurosurgery; hydrocephalus; subdural hematoma; volume

I. Introduction

A. Introduction to the Problem

Hydrocephalus is a medical condition in which there is an abnormal accumulation of cerebrospinal fluid (CSF) in the brain. This causes increased intracranial pressure inside the skull and may cause progressive enlargement of the head if it occurs in childhood, potentially causing neurological dysfunction, mental disability and death [1]. The typical surgical solution to this problem is insertion of a ventriculoperitoneal shunt which drains CSF from cerebral ventricles into abdominal cavity. This procedure for pediatric hydrocephalus has failure rates as high as 40 percent in the first 2 years with ongoing failures thereafter [2]. In developed countries, these failures can be treated in a timely manner. However, in developing nations, these failures can often lead to severe complications and even death. To overcome these challenges, a procedure has been developed which avoids shunts known as endoscopic third ventriculostomy and choroid plexus cauterization [3]. However, the long-term outcome comparison of these methods has not been fully quantified. One way of achieving quantitative comparison is to compare the volumes of brain and CSF before and after surgery. These volumes can be estimated by segmenting brain imagery (MR and/or CT) into CSF and brain tissue. Manual segmentation and volume estimation have been carried out but this is tedious and not scalable across a large number of patients. Therefore, automated/semi-automated brain image segmentation methods are desired and have been pursued actively in recent research.

Substantial previous work has been done in the past for segmentation of pre-operative (pre-op) CT-scans of hydrocephalic patients [4]–[7]. It has been noted that the volume of the brain appears to correlate with neurocognitive outcome after treatment of hydrocephalus [5]. Figure 1A) shows pre-op CT images and Figure 1B) shows corresponding segmented images using the method from [4] for a hydrocephalic patient. The top row of Figure 1A) shows the slices near base of the skull, second row shows the middle slices and bottom row shows the slices near top of the skull. As we observe from Figure 1, segmentation of pre-op images can be a relatively simple problem as the intensities of CSF and brain tissue are clearly distinguishable. However, post-op images can be complicated by addition of further geometric distortions and the introduction of subdural hematoma and fluid collections (subdurals) pressing on the brain. These subdural collections have to be separated from brain and CSF before volume calculations are made. Therefore, the images have to be segmented into 3 classes (brain, CSF and subdurals) and subdurals must be removed from the volume determination. Figure 2 shows sample post-operative (post-op) images of 3 patients having subdurals. Note that the subdurals in patient-1 are very small compared to the subdurals in other two patients. Further, large subdurals are observed in patient-3 on both sides of the brain as opposed to patient-2. The other observation we can make is that the intensity of subdurals in patient-2 is close to the intensity of CSF, whereas the intensity of subdurals in

other two patients is close to intensity of brain tissue. The histogram of the pixel intensity of the images remains bi-modal making it further challenging to separate subdurals from brain and CSF.

B. Closely Related Recent Work

Many methods have been proposed in the past for segmentation of brain images [4], [8]–[13]. Most of these methods work on the principles of intensity based thresholding and model-based clustering techniques. However these traditional methods for segmentation fail to identify subdurals effectively as they are hard to characterize by a specific model, and subdurals pose different range of intensities for different patients. For example, Figure 3 illustrates the performance of [11] on the images of 3 different patients with subdurals. We can observe that the accuracy in segmenting these images is very poor. Apart from these general methods for brain image segmentation, relatively limited work has been done to identify subdurals [14]–[18]. These methods work on the assumption that the images have to be segmented into only 2 classes which are brain and subdurals. Therefore, these methods are unlikely to succeed for images acquired from hydrocephalic patients where CSF volume is significant. Because intensity or other features that can help characterize a pixel into one of three segments (brain, CSF and subdurals) are not apparent; they must be discovered via a learning framework.

Recently, sparsity constrained learning methods have been developed for image classification [19] and found to be widely successful in medical imaging problems [20]–[23]. The essence of the aforementioned sparse representation based classification (SRC) is to write a test image (or patch) as a linear combination of training images collected in a matrix (dictionary), such that the coefficient vector is determined under a sparsity constraint. SRC has seen significant recent application to image segmentation [24]–[28] wherein a pixel level classification problem is essentially solved.

In the works just described, the dictionary matrix simply includes training image patches from each class (segment). Because each pixel must be classified, in segmentation problems training dictionaries can often grow to be prohibitively large. Learning compact dictionaries [29]–[31] continues to be an important problem. In particular, the Label Consistent K-SVD (LC-KSVD) [30] dictionary learning method, which has demonstrated success in image classification has been re-purposed and successfully applied to medical image segmentation [32]–[36].

Motivation and Contributions—In most existing work on sparsity based segmentation, a dictionary is used for each voxel/pixel that creates large computational as well as memory footprint. Further, the objective function for learning dictionaries described in the above literature (based invariably on LC-KSVD) is focused on extracting features that characterize each class (segment) well. We contend that the dictionary corresponding to a given class (segment) must additionally be designed to poorly represent out-of-class samples. We develop a new objective function that incorporates an out-of-class penalty term for learning dictionaries that accomplish this task. This leads to a new but harder optimization problem, for which we develop a tractable solution. We also propose the use of a new feature that

incorporates the distance of a candidate pixel from the edge of the brain computed via a distance transform. This is based on the observation that subdurals are almost always attached to the boundary of the brain. Both intensity patches as well as the distance features are used in the learning framework. The main contributions of this paper are summarized as follows:

1. **A new objective function to learn dictionaries for segmentation under a sparsity constraint:** Because discriminating features are automatically discovered, we call our method feature learning for image segmentation (FLIS). A tractable algorithmic solution is developed for the dictionary learning problem.
2. **A new feature that captures pixel distance from the boundary of brain** is used to identify subdurals effectively as subdurals are mostly attached to the boundary of the brain. This feature also enables the dictionary learning framework to use a single dictionary for all the pixels in an image as opposed to the existing methods that use a separate dictionary for each pixel type. Incorporating this additional "distance based feature" helps significantly reduce the computation and memory footprint of FLIS.
3. **Experimental validation:** Validation on challenging real data acquired from CURE Children's Hospital of Uganda is performed. FLIS results are compared against manually labeled segmentation as provided by an expert neurosurgen. Comparisons are also made against recent and state of the art sparsity based methods for medical image segmentation.
4. **Complexity analysis and memory requirements:** We analytically quantify the computational complexity and memory requirements of our method against competing methods. The experimental run time on typical implementation platforms is also reported.
5. **Reproducibility:** The experimental results presented in the paper are fully reproducible and the code for segmentation and learning FLIS dictionaries is made publicly available at: https://scholarsphere.psu.edu/concern/generic_works/bvq27zn031.

A preliminary version of this work was presented as a short conference paper at the 2017 IEEE Int. Conference on Neural Engineering [37]. Extensions to the conference paper include a detailed analytical solution to the objective function in Eq. (7). Further, extensive experiments are performed by changing various parameters of our algorithm and new statistical insights are provided. Additionally, a detailed complexity analysis is performed and memory requirements of FLIS along with competing methods is presented.

The remainder of the paper is organized as follows. A review of sparsity based segmentation and detailed description of the proposed FLIS is provided in Section II. Experimental results are reported in Section III including comparisons against state of the art. The appendix contains an analysis of the computation and memory requirements of our method and selected competing methods. Concluding remarks are provided in Section IV.

II. Feature Learning For Image Segmentation (FLIS)

A. Review of Sparse Representation Based Segmentation

To segment a given image into C classes/segments, every pixel z in the image has to be classified into one of these classes/segments. The general idea is to collect intensity values from a patch of size $w \times w$ (in case of 3D images a patch of size $w \times w \times w$ is considered) around each pixel and to represent this patch as a sparse linear combination of training patches that are already manually labeled. This idea is mathematically represented by Eq. (1). $m(z) \in \mathbb{R}^{(w^2)\times 1}$ represents a vector of intensity values for a square patch around pixel z. $Y(z) \in \mathbb{R}^{(w^2)\times N}$ represents the collection of N training patches for pixel z in a matrix form. $\alpha \in \mathbb{R}^{N\times 1}$ is the vector obtained by solving Eq. (1). $\|\bullet\|_0$ represents l_0 pseudo-norm of a vector which is the number of non-zero elements in a vector. $\|\bullet\|_2$ represents the l_2 Euclidean norm. The intuition behind this idea is to minimize the reconstruction error between $m(z)$ and the linear combination $Y(z)\alpha$ with the number of non-zero elements in α less than L. The constraint on l_0 pseudo-norm hence enforces sparsity. Often the l_0 pseudo-norm is relaxed to an l_1 norm [25] to obtain fast and unique global solutions. Once the sparse code α is obtained, pixel likelihood probabilities for each class (segment) $j \in \{1, \ldots, C\}$ are obtained using Eq. (2) and Eq. (3). The probability likelihood maps are normalized to 1 and a candidate pixel z is assigned to the most likely class (segment) as determined by its sparse code.

$$\arg \min_{\|\alpha\|_0 < L} \|m(z) - Y(z)\alpha\|_2^2 \quad (1)$$

$$P_j(z) = \frac{\sum_{i=1}^{N} \alpha_i \delta_j(V_i)}{\sum_{i=1}^{N} \alpha_i} \quad (2)$$

where V_i is the i^{th} column vector in the pre-defined dictionary $Y(z)$, and $\delta_j(V_i)$ is an indicator defined as

$$\delta_j(V_i) = \begin{cases} 1, V_i \in \text{class } j \\ 0, \text{otherwise} \end{cases} \quad (3)$$

Note that training dictionaries $Y(z)$ could grow to be prohibitively large, which motivates the design of compact dictionaries that can lead to high accuracy segmentation. Tong [32] *et al.* adapted the well-known LC-KSVD method [30] for segmentation by minimizing reconstruction error along with enforcing a label-consistency criteria. The idea is formally quantified in Eq. (4). For a given pixel z, $Y(z) \in \mathbb{R}^{(w^2)\times N}$ represents all the training patches for pixel z. N is the number of training patches. $D(z) \in \mathbb{R}^{(w^2)\times K}$ is the compact dictionary that is obtained with K being the size of the compact dictionary. $\|X\|_0 < L$, a sparsity

constraint means that each column of X has no more than L non-zero elements. $H(z) \in \mathbb{R}^{C\times N}$ represents the label matrix for the training patches with C being the number of classes/segments to which a given pixel can be classified. For example in our case $C = 3$ (Brain, CSF and Subdurals) and the label matrix for a patch around a pixel which has its ground truth as CSF will be $[0\ 1\ 0]^T$. $W(z) \in \mathbb{R}^{C\times K}$ is the linear classifier which is obtained along with $D(z)$ to represent $H(z)$. $\|\bullet\|_F$ represents the Frobenius (squared error) norm. The terms in black minimize reconstruction error while the term in red represents the label-consistency criteria. When a new test image is analyzed for segmentation, for each pixel z, $D(z)$ and $W(z)$ are invoked and the sparse code $\alpha \in \mathbb{R}^{K\times 1}$ is obtained by solving Eq. (5) which is an l_1 relaxation form of Eq. (1). Unlike the classification strategy used in Eq. (2), we use the linear classifier $W(z)$ on sparse code α to classify/segment the pixel which is shown in Eq. (6). Note that β is a positive regularization parameter that controls the relative regularization between reconstruction error and label consistency.

$$\arg \min_{D(z),W(z),X} \{ \min_{\|X\|_0 < L} \{ \|Y(z) - D(z)X\|_F^2 + \beta \|H(z) - W(z)X\|_F^2 \} \} \quad (4)$$

$$\arg \min_{\alpha > 0} \|m(z) - D(z)\alpha\|_2^2 + \lambda \|\alpha\|_1 \quad (5)$$

$$H_z = W(z)\alpha,\ label(z) = \arg \max_j (H_z(j)), \quad (6)$$

where H_Z is the class label vector for the tested pixel z, and the arg max reveals the best labelling achieved through applying α to the linear classifier $W(z)$.

Tong [32] *et al.*'s work is promising for segmentation but we identify two key open problems: 1.) learned dictionaries for each pixel lead to a high computational and memory footprint, and 2.) the label consistency criterion enhances segmentation by encouraging intra- or within-class similarity but inter-class differences must be maximized as well. Our proposed FLIS addresses both these issues.

B. FLIS Framework

We introduce a new feature that captures the pixel distance from the boundary of the brain. This serves two purposes. First, as we observe from Figure 2, subdurals are mostly attached to the boundary of the brain. Adding this feature along with the vectorized patch intensity intuitively helps enhance the recognition of subdurals. Secondly, we no longer need to design pixel specific dictionaries because the aforementioned "distance vector" (for a patch centered around a pixel) provides enough discriminatory nuance.

Notation—For a given patient, we have a stack of T CT slice images starting from base of the skull to top of the skull which can be observed from Figure 1. The goal is to segment

each image of the stack into three categories: brain, CSF and subdurals. Let $Y_B \in \mathbb{R}^{d \times N_B}$, $Y_F \in \mathbb{R}^{d \times N_F}$ and $Y_S \in \mathbb{R}^{d \times N_S}$ represent the training samples of brain, CSF and subdurals respectively. Each column of Y_i, $i \in B, F, S$ represents intensity of the elements in a patch of size $w \times w$ around a training pixel concatenated with the distances from boundary of brain for each pixel in the patch (described in detail in Section II-E). N_i represents the number of training patches for each class/segment. They are chosen to be same for all the 3 classes/segments. We denote the dictionaries learned as $D_i \in \mathbb{R}^{d \times K}$. K is the size of each dictionary. $X_i \in \mathbb{R}^{K \times N_i}$ represents the matrix that contains the sparse code for each training sample in Y_i. $H_i \in \mathbb{R}^{3 \times N_i}$ represents the label matrices of the corresponding training elements Y_i. For example, a column vector of H_B looks like $[1\ 0\ 0]^T$ and finally, W_i denotes the linear classifier that is learned to represent H_i.

C. Problem Formulation

The dictionary D_i should be designed such that it represents in-class samples effectively and poorly represent complementary samples along with achieving the label consistency criteria. To ensure this, we propose the following problem:

$$\arg \min_{D_i, W_i} \left\{ \frac{1}{N_i} \min_{\|X_i\|_0 < L} \{ \|Y_i - D_i X_i\|_F^2 + \beta \|H_i - W_i X_i\|_F^2 \} - \frac{\rho}{\hat{N}_i} \min_{\|\hat{X}_i\|_0 < L} \{ \|\hat{Y}_i - D_i \hat{X}_i\|_F^2 + \beta \|\tilde{H}_i - W_i \hat{X}_i\|_F^2 \} \right\} \quad (7)$$

The terms with ($\hat{\bullet}$) represent the complementary samples of a given class, $\|\bullet\|_F$ represents Frobenius norm and $\|X\|_0 < L$ implies that each column of $\|X\|$ has non-zero elements not more than L. The label matrices are concatenated, $\tilde{H}_i = [H_i\ H_i]$, to maintain consistency with the dimension of $W_i \hat{X}_i$, because there are two complimentary samples. β and ρ are positive regularization parameters. ρ is an important parameter to obtain a solution for the objective function that we discuss in subsequent sections.

Intuition behind the objective function—The term in black makes sure that intra-class difference is small and the term in red enforces label-consistency. These two terms make sure that in-class samples are well represented. To represent the complementary samples poorly, the reconstruction error between the complementary samples and the sparse linear combination of in-class dictionary samples should be large. This is achieved through the term in blue. Further, a ”label-inconsistency term” is added (in brown) utilizing the sparse code for out of class samples, which again encourages interclass differences. Essentially, the combination of terms in blue and brown enables us to discover discriminative features that differentiate one class (segment) from another effectively. Note that the objective functions described in [32]–[36] are special cases of Eq. (7) since they *do not* include terms that emphasizes inter-class differences. The visual representation of our idea in comparison with the objective function defined in [32] (known as discriminative dictionary learning and sparse coding (DDLS)) is shown in Figure 4. The problem in Eq. (7) is non-convex with

respect to its optimization variables; we develop a new tractable solution which is reported next.

D. Proposed Solution

For simplifying notation in Eq. (7), we replace Y_i, $\hat{Y}_i$, X_i, $\hat{X}_i$, H_i, $\tilde{H}_i$, W_i, $\hat{W}_i$, N_i, $\hat{N}_i$ with Y, $\hat{Y}$, X, $\hat{X}$, $\hat{X}$, H, $\tilde{H}$, W, $\hat{W}$, N, $\hat{N}$ respectively. Therefore, the cost function becomes

$$\arg \min_{D,W} \left\{ \frac{1}{N} \min_{\|X\|_0 < L} \{\|Y - DX\|_F^2 + \beta \|H - WX\|_F^2\} - \frac{\rho}{\hat{N}} \min_{\|\hat{X}\|_0 < L} \{\|\hat{Y} - D\hat{X}\|_F^2 + \beta \|\tilde{H} - W\hat{X}\|_F^2\} \right\} \quad (8)$$

First, an appropriate L should be determined. We begin by learning an "initialization dictionary" using the well-known online dictionary learning (ODL) [38] given by:

$$(D^{(0)}, X^{(0)}) = \arg \min_{D,X} \{\|Y - DX\|_F^2 + \lambda \|X\|_1\} \quad (9)$$

where λ is a positive regularization parameter. An estimate for L can then be obtained by:

$$L \approx \frac{1}{N} \sum_{i=1}^{N} \|x_i^{(0)}\|_0 \quad (10)$$

where $x_i^{(0)}$ represents the i^{th} column of $X^{(0)}$.

We develop an iterative method to solve Eq. (8). The idea is to find X, $\hat{X}$ with a fixed values of D, W and then obtain D, W with the updated values of X, $\hat{X}$. This process is repeated until D, W converge. Since, we have already obtained an initial value for D from Eq. (9), we need to find an initial value for W. To find an initial value for W, we obtain the sparse codes X and $\hat{X}$ by solving the following equations:

$$\arg \min_{\|X\|_0 \leq L} \|Y - DX\|_F^2; \ \arg \min_{\|\hat{X}\|_0 \leq L} \|\hat{Y} - D\hat{X}\|_F^2$$

The above can be combined to find $\bar{X}$ in Eq. (11) using orthogonal matching pursuit (OMP) [39].

$$\arg \min_{\|\bar{X}\|_0 \leq L} \|\bar{Y} - D\bar{X}\|_F^2 \quad (11)$$

where, $\bar{Y} = [Y\ \hat{Y}]$, $\bar{X} = [X\ \hat{X}]$. Then, to obtain the initial value for W, we use the method proposed in [30] which is given by:

$$W = \bar{H}\bar{X}^t(\bar{X}\bar{X}^t + \lambda_1 I)^{-1} \quad (12)$$

where $\bar{H} = [H\ \tilde{H}]$. λ_1 is a positive regularizer parameter. Once the initial value of W is obtained, we construct the following vectors:

$$Y_{new} = \begin{pmatrix} Y \\ \sqrt{\beta}H \end{pmatrix}, \hat{Y}_{new} = \begin{pmatrix} \hat{Y} \\ \sqrt{\beta}\tilde{H} \end{pmatrix}, D_{new} = \begin{pmatrix} D \\ \sqrt{\beta}W \end{pmatrix}$$

As we have the initial values of D, W, we obtain the values of X, $\hat{X}$ by solving the following equation:

$$\arg \min_{\|\bar{X}\|_0 \le L} \|\bar{Y}_{new} - D_{new}\bar{X}\|_F^2 \quad (13)$$

where $\bar{Y}_{new} = [Y_{new}\ \hat{Y}_{new}]$, $\bar{X} = [X\ \hat{X}]$.

With these values of X and $\hat{X}$, we find D_{new} by solving the problem in Eq. (14) which automatically gives the values for D, W.

$$\arg \min_{D_{new}} \left\{ \frac{1}{N}\|Y_{new} - D_{new}X\|_F^2 - \frac{\rho}{\hat{N}}\|\hat{Y}_{new} - D_{new}\hat{X}\|_F^2 \right\} \quad (14)$$

Using the definition of Frobenius norm, the above equation expands to:

$$\arg \min_{D_{new}} \left\{ \frac{1}{N}(Y_{new} - D_{new}X)(Y_{new} - D_{new}X)^T - \frac{\rho}{\hat{N}}(\hat{Y}_{new} - D_{new}\hat{X})(\hat{Y}_{new} - D_{new}\hat{X})^T \right\} \quad (15)$$

Applying the properties of trace and neglecting the constant terms in Eq. (15), solution to the problem in Eq. (14) is equivalent to

$$\arg \min_{D_{new}} \{ -2\text{trace}(ED_{new}^T) + \text{trace}(D_{new}FD_{new}^T)) \} \quad (16)$$

where, $E = \frac{1}{N}Y_{new}X^T - \frac{\rho}{\hat{N}}\hat{Y}_{new}\hat{X}^T$; $F = \frac{1}{N}XX^T - \frac{\rho}{\hat{N}}\hat{X}\hat{X}^T$. The problem in Eq. (16) is convex if F is positive semidefinite. However, F is not guaranteed to be positive semidefinite. To make

F a positive semidefinite matrix, ρ should be chosen in a way such that the following condition is met:

$$\frac{1}{N}\lambda_{min}(XX^T) - \frac{\rho}{\hat{N}}\lambda_{max}(\hat{X}\hat{X}^T) > 0 \quad (17)$$

where $\lambda_{min}(\bullet)$ and $\lambda_{max}(\bullet)$ represent the minimum and maximum eigenvalues of the corresponding matrices. Once an appropriate ρ is chosen, Eq. (16) can be solved using dictionary update step in [38]. After we obtain D_{new}, Eq. (13) is solved again to obtain new values for X and $\hat{X}$ and we keep iterating between these two steps to obtain the final D_{new}. The entire procedure is formally described in Algorithm 1, which is used on a per-class basis to learn 3 class/segment specific dictionaries corresponding to brain, CSF and subdurals.

After we obtain class specific dictionaries and linear classifiers, we concatenate them to obtain $D = [D_B\ D_F\ D_S]$ and $W = [W_B\ W_F\ W_S]$.

Assignment of a test pixel to a class (segment)—Once the dictionaries are learned, to classify a new pixel z, we extract a patch of size $w \times w$ around it to collect the intensity values and distance values from the boundary of the brain for the elements in the patch to form column vector $m(z)$. Then we find the sparse code α in Eq. (18) using the learned dictionary D. Once α is obtained, we classify the pixel using Eq. (19).

$$\arg \min_{\alpha > 0} \|m(z) - D\alpha\|_2^2 + \lambda\|\alpha\|_1 \quad (18)$$

$$H_z = W\alpha,\ label = \arg \max_j (H_z(j)) \quad (19)$$

Algorithm 1

FLIS algorithm

Input: Y, $\hat{Y}$, H, ρ, β, dictionary size K

Output: D, W

procedure FLIS

Find L and an initial value for D using Eq. (9) and Eq. (10)

Find X and $\hat{X}$ using Eq. (11)

Initialize W using Eq. (12)

Update $Y_{new} = \begin{pmatrix} Y \\ \sqrt{\beta}H \end{pmatrix}, \hat{Y}_{new} = \begin{pmatrix} \hat{Y} \\ \sqrt{\beta}\tilde{H} \end{pmatrix}, D_{new} = \begin{pmatrix} D \\ \sqrt{\beta}W \end{pmatrix}$

Update X, $\hat{X}$ using Eq. (13)

while not converged do

10: Fix X, $\hat{X}$ and calculate $E = \frac{1}{N} Y_{new} X^T - \frac{\rho}{\hat{N}} \hat{Y}_{new} \hat{X}^T; F = \frac{1}{N} X X^T - \frac{\rho}{\hat{N}} \hat{X} \hat{X}^T$

11: Update D_{new} by solving

$$\arg \min_{D_{new}} \{ -2\text{trace}(ED_{new}^T) + \text{trace}(D_{new} F D_{new}^T) \}$$

12: Fix D_{new}, find X and $\hat{X}$ using Eq. (13)

13: **end while**

14: **end procedure**

15: **RETURN:** D_{new}

E. Training and Test Procedure Design for Hydrocephalic Image Segmentation

Training Set-Up—In selecting training image patches for segmentation, it is infeasible to extract patches for all the pixels in each training image because that would require a lot of memory. Further, it is desired that patches used from training images should be in correspondence with the patches from test images. For example, training patches collected from the slices in the middle of the CT stack cannot be used for segmenting a slice that belongs to top or bottom. To address this problem, we divide the entire CT-stack of any patient into P partitions such that images belonging to a given partition are anatomically similar. For each image in a partition (i.e a sub collection of CT image stack), we must carefully extract patches to have enough representation from the 3 classes (segments) and likewise have enough diversity in the range of distances from the boundary of the brain.

Patch Selection Strategy for each class/segment—First we find a *candidate* region for each image in the CT-stack by using an optical flow approach as mentioned in [4]. The candidate region is a binary image which labels the region of an image that is to be segmented into brain, CSF and subdurals as 1. Then, the distance value for each pixel z is given by $DT(z) = \min(d(z,q)) : CR(q) = 0$, where $d(z,q)$ is the Euclidean distance between pixel z and pixel q and CR is the candidate region. For a pixel z, it is essentially the minimum distance calculated from all the pixels that are not part of the candidate region. The candidate region of a sample image and its distance transform is shown in Fig. 5. A subset of “these distances” should be used in our training feature vectors. For this purpose, we propose a simple strategy wherein first we calculate the maximum and minimum distance of a given label/class in a CT image and pick patches randomly such that the distance range is uniformly sampled from min to max values. The pseudo-code for this strategy and more implementation details can be found in [40].

Once training patches for each partition are extracted, we learn dictionaries and linear classifiers for each partition using the objective function described in Section II-C. The entire training setup and segmentation of a new test CT stack is summarized as a flow chart in Figure 6.

III. Experimental Results

We report results on a challenging real world data set of CT images acquired from the CURE Children's Hospital of Uganda. Each patient (on an average) is represented by a stack of 28 CT images. We choose the number of partitions of such a stack P to be 12 based on neurosurgeon feedback. The size of each slice is 512×512. Slice thickness of the scans varied from 3mm to 10mm. The test set includes 15 patients while the number of training patients ranged from 9–17 and were non-overlapping with the test set. To validate our results, we used the dice-overlap coefficient, which for regions A and B is defined as

$$DO(A,B) = \frac{2 \mid A \cap B \mid}{\mid A \mid + \mid B \mid} \quad (20)$$

Note, $DO(A,B)$ evaluates to 1, only when $A = B$. The dice-overlap is computed for each method by using carefully obtained manually segmented results under the supervision of an expert neurosurgeon - (SJS). The proposed FLIS is compared against the following state of the art methods:

- SRC [19] based segmentation was implemented in [25] by using pre-defined dictionaries for each voxel/pixel in the scans. The objective function and classification procedure proposed in their work is implemented on our data set.
- LC-KSVD [30] based dictionary learning method was used to segment MR brain images in [32] for hippocampus labeling. Two types of implementations were proposed in their paper which are named as DDLS and F-DDLS. In Fixed-DDLS (F-DDLS) dictionaries are learned offline and segmentation is performed online to improve speed of segmentation whereas in DDLS both operations are performed simultaneously. In this paper, we compare with the DDLS approach, as storing a dictionary for each pixel offline requires a very large memory.

Apart from these two methods, there are few others that use dictionary learning and a sparsity based framework for medical image segmentation [26]–[28], [33]–[36]. The objective function used in these aforementioned methods is similar to the above two methods with the application being different. We chose to compare against [25] and [32] because they are widely cited and were also applied to brain image segmentation.

A. The need for a learning framework

Before we compare our method against the state of the art in learning based segmentation, we demonstrate the superiority of the learning based approaches in comparison to the traditional intensity based methods. It was illustrated visually in Fig. 3 in Section I that intensity based methods find it difficult to differentiate subdurals from brain and CSF. To validate this quantitatively, we compare dice-overlap coefficients obtained by using the segmentation results of [11][1] which is one of the best known intensity based methods and addressed as Brain Intensity Segmentation (BIS). The comparisons are reported in Table I.

[1]Note that the method in [11] was implemented for MR brain images. We adapted their strategy for segmenting our CT images.

The learning based methods use a patch size of 11×11 with number of training patients set to 15 and the sizes of individual class specific dictionaries set to 80.

The results in Table I confirm that learning based methods clearly outperform the traditional intensity based method, esp. in terms of the accuracy of identifying subdurals. Note that the dice overlap values in Table I for each class/segment are averaged over the 15 test patients. This will be the norm for the remainder of this Section unless otherwise stated. We performed a balanced two-way Analysis of Variance (ANOVA)[2] [42] on the dice overlap values across patients for all 3 classes (Brain, CSF and Subdural). Fig. 7 illustrates these comparisons using posthoc Tukey range test [42] and confirms that SRC, DDLS and FLIS (learning based methods) are significantly separated from BIS. p values of BIS compared with other methods are observed to be much less than .01 which emphasizes the fact that learning based methods are more effective.

B. Parameter Selection

In our method, several parameters have to be chosen carefully before we start implementation. Some of the important parameters are patch size, dictionary size, number of training patients and regularization parameters ρ and β. ρ and β are picked by a cross-validation procedure [43], [44] such that ρ is in compliance with Eq. (17). The best values are found to be $\rho = .5$ and $\beta = 2$. Our algorithm is fairly robust to other parameters such as patch size, number of training patients and length of dictionaries which is discussed in the subsequent sub-sections.

C. Influence of Patch Size

If the patch size is very small, namely a single pixel in the extreme case, the necessary spatial information to accurately determine its class/segment is unavailable. On the other hand, a very large patch size might include pixels from different classes. For the experiment performed, the dictionary size of each class/segment and number of training patients for performing experiments are set to 120 and 17 respectively. Experiments are reported for square patch windows with size varying from 5 to 25. The mean dice overlap values for all the 15 patients that are shown in Fig. 8 reveal that the results are quite stable for patch size in the range 11 to 17, indicating that while patch size should be chosen carefully, FLIS is robust against small departures from the optimal choice.

D. Influence of Dictionary Size

Dictionary size is another important parameter in our method. Similar to patch size, very small dictionaries are incomplete and can not represent the data accurately. However, large dictionaries can represent the data more accurately, but at the cost of increased run-time and memory requirements.

In the results presented next, varying dictionary sizes of 20, 80, 120 and 150 are chosen. Note that these dictionary sizes are for each individual class. However, DDLS does not use class specific dictionaries. Therefore, to maintain consistency in both the methods, the

[2]Prior to application of ANOVA, we rigourously verified that the observations (dice overlap values) satisfy ANOVA assumptions [41].

overall dictionary size for DDLS is fixed to be 3 times the size of each individual dictionary in our method. Table II compares FLIS with DDLS for different dictionary sizes. We did not compare with [25] as dictionary learning in not used in their approach. Experiments are conducted with a patch size of 13×13 and with data from 17 patients used for training.

From Table II, we observe that FLIS remains fairly stable with the change in size of dictionary whereas the DDLS method performed better in identifying subdurals as the size of dictionary is increased. For a fairly small dictionary size of 20, the performance of both methods drops but FLIS is still relatively better. Further, to compare both the methods statistically, a 3-way balanced ANOVA is performed for all the 3 classes as shown in Fig. 9. We observe that FLIS exhibits superior segmentation accuracy compared to DDLS although there is significant overlap between confidence intervals of FLIS and DDLS. This can be primarily attributed to the discriminative capability of the FLIS objective function which automatically discovers features that are crucial for separating segments. Visual comparisons are available in Figure 10 when size of dictionary is set to 120. Visual results from Figure 10 show that both the methods performed similarly in detecting large subdurals, but FLIS identifies subdurals more accurately in Patient 3 (3rd column of Fig. 10) where the subdurals have a smaller spatial footprint.

E. Performance variation against training

For the following experiment, we vary the number of training and test samples by dividing the total 32 patients CT stacks into 9–23, 11–21, 13–19, 15–17, 17–15, 19–13 and 21-11 configurations (to be read as training-test). Figure 12 compares our method with DDLS and patch based SRC [25] for all these configurations. Note that, the results reported for each configuration are averaged over 10 random combinations of a given training-test configuration to remove selection bias. The per-class dictionary size was fixed to 80 for our method and DDLS, whereas for [25], the dictionary size is determined automatically for a given training selection. The patch size is set to 13×13.

A plot of dice overlap vs. training size is shown in Fig. 12. Unsurprisingly, each of the three methods shows a drop in performance as the number of training image patches (proportional to the number of training patients) decreases. However, note that FLIS exhibits the most graceful degradation.

Fig. 13 represents the gaussian fit for the histogram (for all 10 realizations combined) of dice-overlap coefficients for the configuration 13–19. Two trends may be observed: 1.) FLIS histogram has a mean higher than competing methods, indicating higher accuracy, 2.) the variance is smallest for FLIS confirming robustness to choice of training-test selection.

Comparisons are visually shown in Figure 11. A similar trend is also observed here where patch based SRC and DDLS improve as the number of training patients increase. We observe that DDLS and SRC based methods performed poorly in identifying the subdurals for Patient 3 (column 3) in Figure 11. We also observe that both DDLS and FLIS outperform SRC implying that dictionary learning improves accuracy significantly.

F. Discriminative Capability of FLIS

To illustrate the discriminative property of FLIS, we plot the sparse codes that are obtained from the classification stage for our method and DDLS for a single random pixel with a dictionary size of 150 in Fig. 14. The two red lines in the figure act as a boundary for the 3 classes. For each of the three segments, i.e. brain, CSF and subdurals, we note that the active coefficients in the sparse code are concentrated more accurately in the correct class/segment for FLIS vs. DDLS.

To summarize the quantitative results, FLIS stands out particularly in its ability to correctly segment subdurals. The overall accuracy of brain and fluid segmentation is better than the accuracy of subdural segmentation for all the 3 methods. This is to be expected because the amount of subdurals present throughout in the images is relatively small compared to brain and fluid volumes.

G. Computational Complexity

We compare the computational complexity of our FLIS with DDLS method. We do not compare with [25] as it does not learn dictionaries. Complexity of dictionary learning methods is estimated by calculating the approximate number of operations required for learning dictionaries for each pixel. Detailed derivation of complexity is presented in Appendix A. The run-time and derived complexity per pixel are shown in Table III. The run-time and computational complexity are derived per pixel. The values of parameters are defined as follows: The number of training patches $N = 4700$ for each class and the patch size is 11×11. Sparsity level L is chosen to be 5. The run time numbers are consistent with the estimated number of operations shown in Table III obtained by plugging in the values of above parameters in to the derived complexity formulas. FLIS is substantially less expensive from a computational standpoint. This is to be expected because DDLS uses pixel specific dictionaries, whereas FLIS dictionaries are class or segment specific but do *not* vary with the pixel location.

H. Memory requirements

Memory requirements are derived in Appendix B. The memory required for storing dictionaries for all the 3 methods are reported in Table IV. These numbers are obtained assuming each element requires 16 bytes, and the following parameter choices: Number of training patients, $N_t = 15$, patch size = 11×11, $K = 80$ and $I_x = I_y = 512$. Consistent with Section III-G, the memory requirements of FLIS are also modest.

I. Comparison with deep learning architectures

A significant recent advance has been the development of deep learning methods, which have recently been applied to medical image segmentation [45], [46]. We implement the technique in [45] which designs a convolutional neural network (CNN) for segmenting MR images. This method extracts 2D patches of different sizes centered around the pixel to be classified and a separate network is designed for each patch size. The output of each network is then connected to a single softmax layer to classify the pixel. Three different patch sizes were used in their work and the network configuration for each patch size is mentioned in Table V. We reproduced the design in [45] but with CT scans for training. We address this

method as Deep Network for Image Segmentation (DNIS). Results in terms of comparisons with FLIS are shown in Table VI. Note that the training-test configuration of this experiment is the same as the one performed in subsection III-E. Unsurprisingly, FLIS performed better than DNIS for low training scenarios and DNIS performed slightly better than FLIS with an increase in number of training samples. Further, to confirm this statistically, a 3-way balanced ANOVA is performed for all the 3 classes as shown in Fig. 15. It may be inferred from Fig. 15 that FLIS outperforms DNIS in the low to realistic training regime, while DNIS is competitive or mildly better than FLIS when training is generous. An example visual illustration of the results is shown for 3 patients in Fig. 16 where the benefits of FLIS are readily apparent. Also, note that the cost of training DNIS is in hours vs. the training time of FLIS which takes seconds – see Table VI.

IV. Discussion and Conclusion

In this paper, we address the problem of segmentation of post-op CT brain images of hydrocephalic patients from the viewpoint of dictionary learning and discriminative feature discovery. This is very challenging problem from the distorted anatomy and subdural hematoma collections on these scans. This makes subdurals hard to differentiate from brain and CSF. Our solution involves a sparsity constrained learning framework wherein a dictionary (matrix of basis vectors) is learned from pre-labeled training images. The learned dictionaries under a new criterion are shown capable of yielding superior results to state of the art methods. A key aspect of our method is that only class or segment specific dictionaries are necessary (as opposed to pixel specific dictionaries), substantially reducing the memory and computational requirements.

Our method was tested on real patient images collected from CURE Children's Hospital of Uganda and the results outperformed well-known methods in sparsity based segmentation.

Acknowledgments

We thank Tiep Huu Vu for providing his valuable inputs to this work supported by NIH grant R01HD085853.

References

1. Adams R, et al. Symptomatic occult hydrocephalus with normal cerebrospinal-fluid pressure: a treatable syndrome. New England Journal of Medicine. 1965; 273(3):117–126. [PubMed: 14303656]
2. Drake JM, et al. Randomized trial of cerebrospinal fluid shunt valve design in pediatric hydrocephalus. Neurosurgery. 1998; 43(2):294–303. [PubMed: 9696082]
3. Warf BC. Endoscopic third ventriculostomy and choroid plexus cauterization for pediatric hydrocephalus. Clinical neurosurgery. 2007; 54:78. [PubMed: 18504900]
4. Mandell JG, et al. Volumetric brain analysis in neurosurgery: Part 1. particle filter segmentation of brain and cerebrospinal fluid growth dynamics from MRI and CT images. Journal of Neurosurgery: Pediatrics. 2015; 15(2):113–124. [PubMed: 25431902]
5. Mandell JG, et al. Volumetric brain analysis in neurosurgery: part 2. Brain and CSF volumes discriminate neurocognitive outcomes in hydrocephalus. Journal of Neurosurgery: Pediatrics. 2015; 15(2):125–132. [PubMed: 25431901]
6. Luo F, et al. Wavelet-based image registration and segmentation framework for the quantitative evaluation of hydrocephalus. Journal of Biomedical Imaging. 2010; 2010:2.

7. Brandt ME, et al. Estimation of CSF, white and gray matter volumes in hydrocephalic children using fuzzy clustering of MR images. Computerized Medical Imaging and Graphics. 1994; 18(1):25–34. [PubMed: 8156534]
8. Mayer A, Greenspan H. An adaptive mean-shift framework for MRI brain segmentation. IEEE Transactions on Medical Imaging. 2009; 28(8):1238–1250. [PubMed: 19211339]
9. Li C, Goldgof DB, Hall LO. Knowledge-based classification and tissue labeling of MR images of human brain. IEEE transactions on Medical Imaging. 1993; 12(4):740–750. [PubMed: 18218469]
10. Weisenfeld NI, Warfield SK. Automatic segmentation of newborn brain MRI. Neuroimage. 2009; 47(2):564–572. [PubMed: 19409502]
11. Makropoulos A, et al. Automatic whole brain MRI segmentation of the developing neonatal brain. IEEE Transactions on Medical Imaging. 2014; 33(9):9.
12. Ribbens A, et al. Unsupervised segmentation, clustering, and group-wise registration of heterogeneous populations of brain MR images. IEEE transactions on medical imaging. 2014; 33(2):201–224. [PubMed: 23797244]
13. Greenspan H, Ruf A, Goldberger J. Constrained gaussian mixture model framework for automatic segmentation of MR brain images. IEEE transactions on medical imaging. 2006; 25(9):1233–1245. [PubMed: 16967808]
14. LiuB, , et al. Automatic segmentation of intracranial hematoma and volume measurement. 2008 30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society; IEEE; 200812141217
15. Liao CC, et al. A multiresolution binary level set method and its application to intracranial hematoma segmentation. Computerized Medical Imaging and Graphics. 2009; 33(6):423–430. [PubMed: 19428217]
16. SharmaB, , VenugopalanK. Classification of hematomas in brain CT images using neural network. Issues and Challenges in Intelligent Computing Techniques (ICICT), 2014 International Conference on; IEEE; 20144146
17. GongT, , et al. Finding distinctive shape features for automatic hematoma classification in head CT images from traumatic brain injuries. Tools with Artificial Intelligence (ICTAI), 2013 IEEE 25th International Conference on; IEEE; 2013242249
18. Soltaninejad M, et al. A hybrid method for haemorrhage segmentation in trauma brain CT. MIUA. 2014:99–104.
19. Wright J, et al. Robust face recognition via sparse representation. IEEE transactions on pattern analysis and machine intelligence. 2009; 31(2):210–227. [PubMed: 19110489]
20. Srinivas U, et al. Simultaneous sparsity model for histopathological image representation and classification. IEEE transactions on medical imaging. 2014; 33(5):1163–1179. [PubMed: 24770920]
21. Vu TH, et al. Histopathological image classification using discriminative feature-oriented dictionary learning. IEEE transactions on medical imaging. 2016; 35(3):738–751. [PubMed: 26513781]
22. Mousavi HS, et al. Automated discrimination of lower and higher grade gliomas based on histopathological image analysis. Journal of pathology informatics. 2015; 6
23. YuY, , et al. Group sparsity based classification for cervigram segmentation. Biomedical Imaging: From Nano to Macro, 2011 IEEE International Symposium on; IEEE; 201114251429
24. Wang L, et al. Segmentation of neonatal brain MR images using patch-driven level sets. NeuroImage. 2014; 84:141–158. [PubMed: 23968736]
25. Wang L, et al. Integration of sparse multi-modality representation and anatomical constraint for isointense infant brain MR image segmentation. NeuroImage. 2014; 89:152–164. [PubMed: 24291615]
26. Wu Y, et al. Prostate segmentation based on variant scale patch and local independent projection. IEEE transactions on medical imaging. 2014; 33(6):1290–1303. [PubMed: 24893258]
27. ZhouY, , et al. Nuclei segmentation via sparsity constrained convolutional regression. 2015 IEEE 12th International Symposium on Biomedical Imaging (ISBI); IEEE; 201512841287
28. Liao S, et al. Sparse patch-based label propagation for accurate prostate localization in CT images. IEEE transactions on medical imaging. 2013; 32(2):419–434. [PubMed: 23204280]

29. YangM, , et al. Fisher discrimination dictionary learning for sparse representation. 2011 International Conference on Computer Vision; IEEE; 2011543550
30. Jiang Z, Lin Z, Davis LS. Label consistent K-SVD: Learning a discriminative dictionary for recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2013; 35(11):2651–2664. [PubMed: 24051726]
31. MongaV. Handbook of Convex Optimization Methods in Imaging ScienceSpringer; 2017
32. Tong T, et al. Segmentation of MR images via discriminative dictionary learning and sparse coding: Application to hippocampus labeling. NeuroImage. 2013; 76:11–23. [PubMed: 23523774]
33. LeeJ, , et al. Brain tumor image segmentation using kernel dictionary learning. 2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC); IEEE; 2015658661
34. Roy S, et al. Subject-specific sparse dictionary learning for atlas-based brain MRI segmentation. IEEE journal of biomedical and health informatics. 2015; 19(5):1598–1609. [PubMed: 26340685]
35. Bevilacqua M, Dharmakumar R, Tsaftaris SA. Dictionary-driven ischemia detection from cardiac phase-resolved myocardial BOLD MRI at rest. IEEE transactions on medical imaging. 2016; 35(1):282–293. [PubMed: 26292338]
36. Nouranian S, et al. Learning-based multi-label segmentation of transrectal ultrasound images for prostate brachytherapy. IEEE transactions on medical imaging. 2016; 35(3):921–932. [PubMed: 26599701]
37. CherukuriV, , et al. Learning based image segmentation of post-operative ct-images: A hydrocephalus case study. Neural Engineering (NER), 2017 8th International IEEE/EMBS Conference on; IEEE; 20171316
38. Mairal J, et al. Online learning for matrix factorization and sparse coding. Journal of Machine Learning Research. Jan.2010 11:19–60.
39. Tropp JA, Gilbert AC. Signal recovery from random measurements via orthogonal matching pursuit. IEEE Transactions on information theory. 2007; 53(12):4655–4666.
40. CherukuriV, , et al. Tech RepThe Pennsylvania State University; 2017Implementation details of FLIS. [Online]. Available: https://scholarsphere.psu.edu/concern/generic_works/bvq27zn031
41. McDonaldJH. Handbook of biological statisticsVol. 2. Sparky House Publishing; Baltimore, MD: 2009
42. WuCJ, , HamadaMS. Experiments: planning, analysis, and optimizationVol. 552. John Wiley & Sons; 2011
43. KohaviR, , et al. IjcaiVol. 14. Stanford, CA: 1995A study of cross-validation and bootstrap for accuracy estimation and model selection; 11371145
44. LeeMS, , et al. Efficient algorithms for minimizing cross validation error. Machine Learning Proceedings 1994: Proceedings of the Eighth International Conference; Morgan Kaufmann; 2014190
45. Moeskops P, et al. Automatic segmentation of MR brain images with a convolutional neural network. IEEE transactions on medical imaging. 2016; 35(5):1252–1261. [PubMed: 27046893]
46. Pereira S, et al. Brain tumor segmentation using convolutional neural networks in MRI images. IEEE transactions on medical imaging. 2016; 35(5):1240–1251. [PubMed: 26960222]
47. Aharon M, Elad M, Bruckstein A. *rmk*-svd: An algorithm for designing overcomplete dictionaries for sparse representation. IEEE Transactions on signal processing. 2006; 54(11):4311–4322.
48. Rubinstein R, Zibulevsky M, Elad M. Efficient implementation of the K-SVD algorithm using batch orthogonal matching pursuit. Cs Technion. 2008; 40(8):1–15.

Appendix A. Complexity analysis

We derive the computational complexity of our FLIS and compare it with DDLS [32]. Computational complexity for each method is derived by finding the approximate number of operations required per pixel in learning the dictionaries. To simplify the derivation, let us assume that number of training samples and size of dictionary be same for all the 3 classes.

Let they be represented as N and K. Let us also assume that sparsity constraint L remains the same for all the classes. Let the training samples be represented as Y and the sparse code be represented as X.

Two major steps in most of the dictionary learning methods are the dictionary update and sparse coding steps, which in our case are l_0 minimization. The dictionary update step is solved either by using block coordinate descent [38] or the singular value decomposition [47]. The second step which involves solving an Orthogonal Matching Pursuit [39] is the most expensive step. Therefore, to derive the computational complexities, we find the approximate number of operations required to solve the sparse coding step in each iteration.

A. Complexity of FLIS

As discussed above, we find the approximate number of operations required to solve the sparse coding step in our algorithm. To do that, first we find the complexity of the major sparse coding step which is given by Eq. (21).

$$\arg \min_{\|X\|_0 \leq L} \|Y - DX\|_F^2 \quad (21)$$

where the dimension of Y is equal to $\mathbb{R}^{d \times N}$ and dimension of D is equal to $\mathbb{R}^{d \times K}$. For a batch-OMP problem with the above dimensions, the computational complexity is derived in [48] and it is equal to $N(2dK+L^2K+3LK+L^3)+dK^2$. Assuming $L \ll K \approx d \ll N$, it approximately simplifies to

$$NK(2d + L^2). \quad (22)$$

The sparse coding step in our FLIS algorithm requires us to solve $\arg \min_{\|\bar{X}\|_0 \leq L} \|\bar{Y}_{new} - D_{new}\bar{X}\|_F^2$ where $\bar{Y}_{new} \in \mathbb{R}^{(d+3)\times 3N}$ and $D_{new} \in \mathbb{R}^{(d+3)\times K}$ which can be solved from Eq. (13). Substituting these values into Eq. (22), we get the complexity of learning dictionary for a single class as $3NK(2(d+3)+L^2)$. Since we have 3 classes, the overall complexity of learning is multiplied by 3: $C_{FLIS} = 9NK(2(d+3)+L^2)$. As the same dictionary is used for all the pixels in an image I with dimension $I_x \times I_y$,

$$C_{FLIS} = \frac{9NK(2(d+3)+L^2)}{I_x \times I_y}.$$

B. Complexity of DDLS [32]

We already showed that by removing the discriminating term from FLIS in Eq. (7), it turns into the objective function described for DDLS in Section II-C. Therefore, the most complex step remains the same for DDLS as well. However, since DDLS does not include distance feature the size of d changes to $\frac{d}{2}$ and also it computes the dictionaries for all the classes at

once. Keeping these two differences in mind, the computational complexity of DDLS is: $C_{DDLS} = 9NK(2(\frac{d}{2}+3)+L^2)$. In addition, a separate dictionary is computed for each pixel in DDLS, which means the complexity scales with the size of the image.

Appendix B. Memory Requirements

We now calculate the memory required for our method and compare it with DDLS [32] and patch based SRC [25]. Memory requirement for all the methods is calculated by estimating the number of bytes required to store the dictionaries. In the case of FLIS and DDLS, the size of the dictionary plays an important role in calculating memory requirement whereas in SRC, the number of training images plays an important role as it uses pre-defined dictionaries. Another point to note is, as the entire CT stack is divided into P partitions and a dictionary is stored for each partition, we derive the memory required for storing dictionaries for each individual partition. To obtain the total memory required, the formulas derived in the subsequent sections have to be multiplied by P.

A. Memory required for FLIS

Suppose the length of each dictionary is K and the size of the column vector is d, then the size of the complete dictionary for all the 3 classes combined is $d\times 3K$. Further, we also store linear classifier W for classification which is of size $3\times 3K$. Therefore, the complete size of the dictionary is $(d+3)\times 3K$. Assuming each element in dictionary is represented by 16 bytes, the total memory in bytes required for storing FLIS dictionaries is $M_{FLIS} = (d+3)\times 3K\times 16$.

B. Memory required for DDLS [32]

One major difference between FLIS and DDLS is the size of the column vector in DDLS is approximately half of the size in FLIS's case as the distance values are not considered in DDLS. The other major difference is a dictionary is stored for each individual pixel. Keeping these two differences in mind and with the same dictionary length, the total memory in bytes required for storing DDLS dictionaries is $M_{DDLS} = (\frac{d}{2}+3)\times 3K\times 16\times I_x\times I_y$ where $I_x\times I_y$ is the image size.

C. Memory required for Patch based SRC [25]

In SRC method, predefined dictionaries for each pixel are stored instead of compact dictionaries. For a given pixel x in an image, a patch of size $w\times w$ is considered around the same pixel location in training images and then a patch of size $w\times w$ around new pixels form the dictionary of pixel x. Assuming there are N_t training images, the total size of the dictionary for a given pixel is $\frac{d}{2}\times\frac{d}{2}\times N$ as the size of the patch in this method is approximately half of the size of column vector in FLIS method. Therefore, the total memory in bytes required for this methods is $M_{SRC} = \frac{d}{2}\times\frac{d}{2}\times N_t\times I_x\times I_y\times 16$.

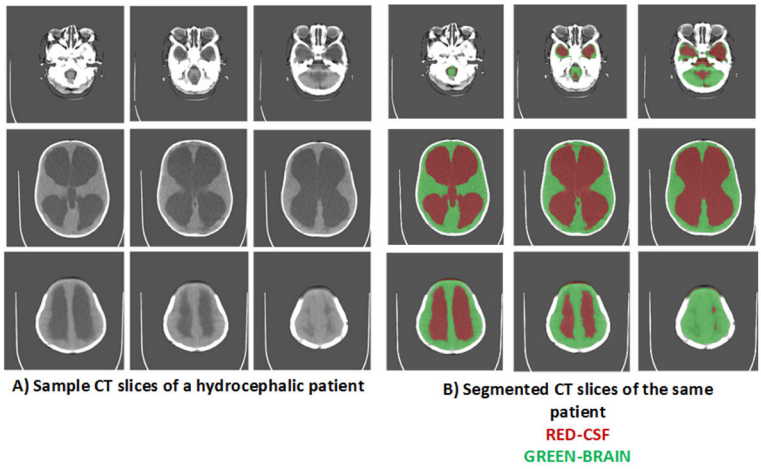

Fig. 1.

A) Sample Pre-operative (pre-op) CT scan slices of a hydrocephalic patient B) Segmented CT-slices of the same patient using [4]

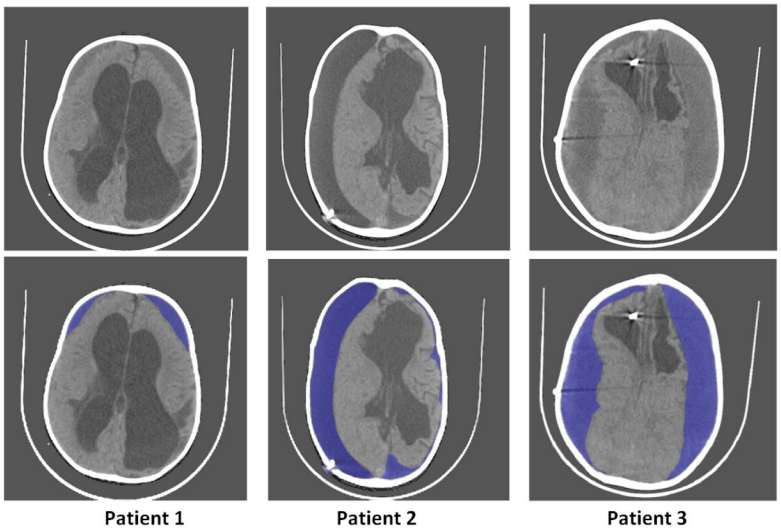

Fig. 2.

Sample post-op CT-images of 3 patients. Top row shows the original images. Bottom row shows subdurals marked in blue. A shunt catheter is visible in patients 2 and 3.

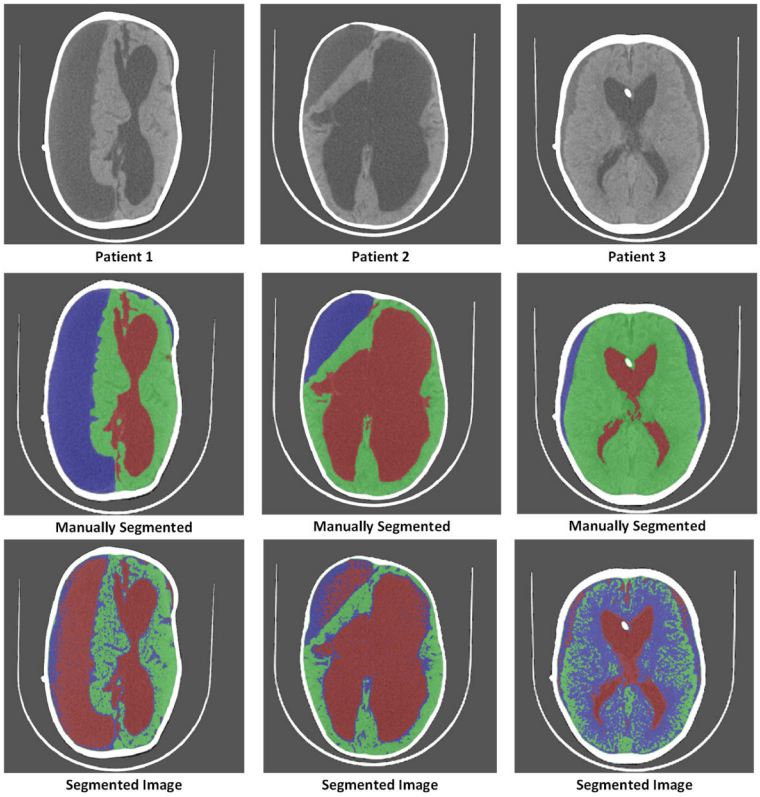

Fig. 3.

Demonstration of segmentation using a traditional intensity based method [11]. Top row represents original images of 3 patients. Second row represents manually segmented images. Third row represents the segmentation using [11]. Green-Brain, Red-CSF, Blue-Subdurals

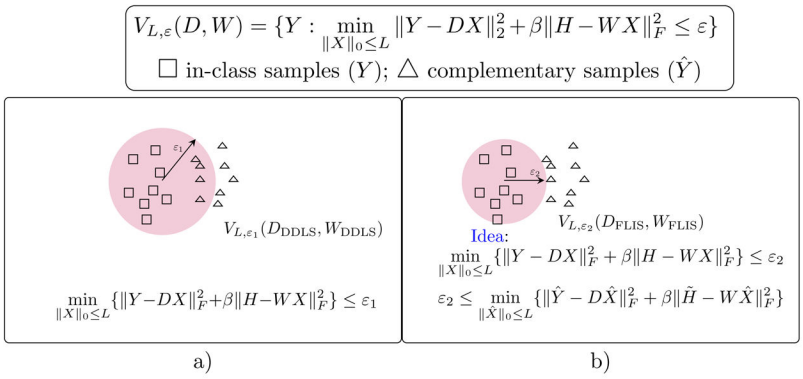

Fig. 4.

Visual representation of our FLIS in comparison with DDLS [32]. a) represents the idea of DDLS and b) represents a desirable outcome of our idea which is more capable of differentiating in-class and out of class samples.

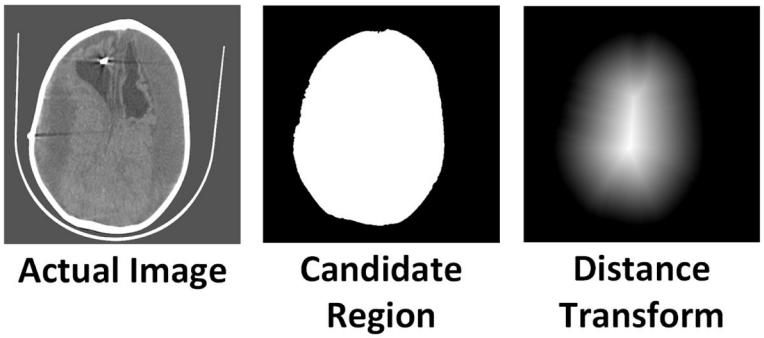

Fig. 5.
Visual representation of obtaining distance values from a CT-slice.

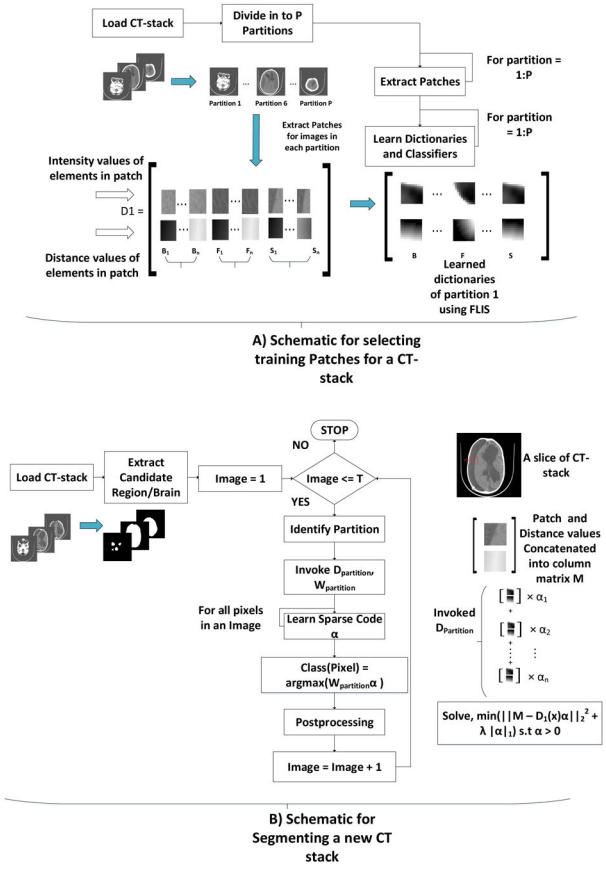

Fig. 6.

A) illustrates the procedure for selecting patches for training. B) illustrates the procedure for segmentation of a new CT- stack

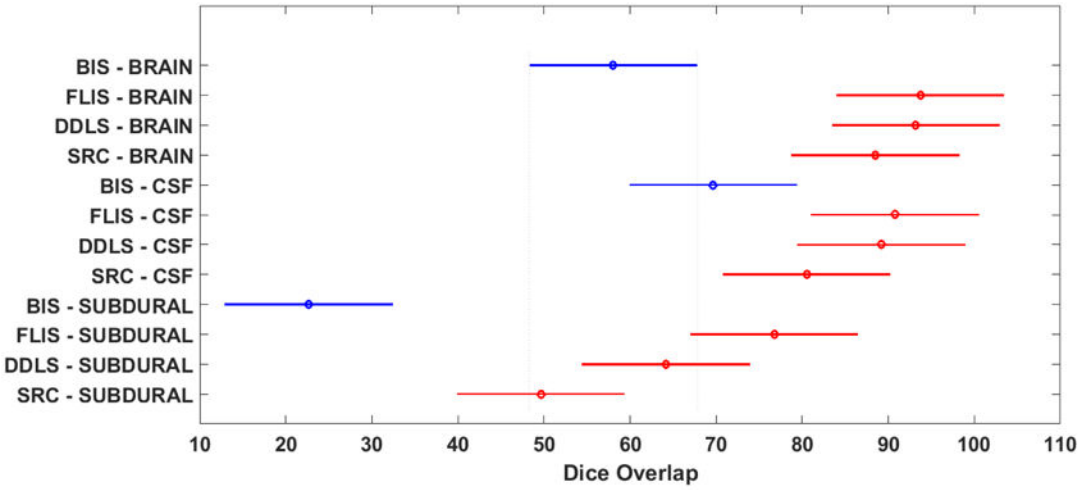

Fig. 7.

Comparison of traditional intensity based thresholding method with learning based approaches by a two-way ANOVA. Values reported by ANOVA across the method factor are $df=3$, $F=45.23$, $p \ll .01$, indicating that results of learning based approaches are significantly different and better than BIS. The intervals shown represent the 95 percent confidence intervals of the dice overlap values for the corresponding method-class configuration. Blue color represents BIS method and Red indicates the learning based approaches.

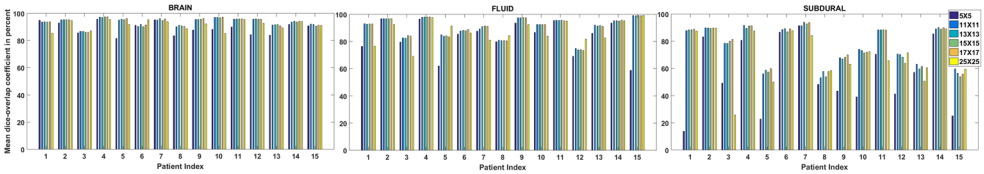

Fig. 8.

Mean dice overlap coefficients for all the 15 patients using our method are reported in this figure. Results for different square patch sizes varying from 5 to 25 are reported.

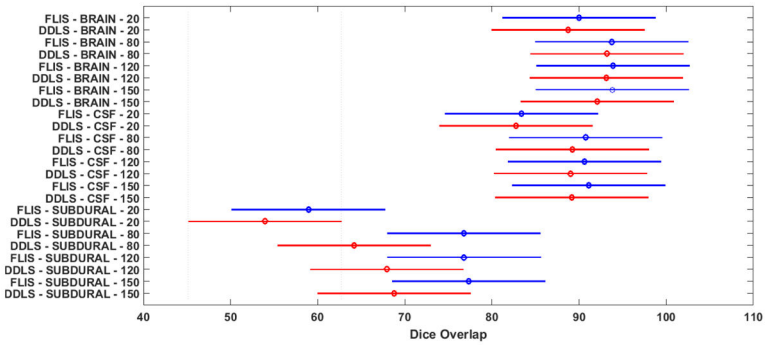

Fig. 9.

Comparison of FLIS with DDLS for different dictionary sizes by using a 3-way ANOVA. The intervals represent the 95 percent confidence intervals of dice overlap values for a given configuration of method-class-dictionary size. FLIS is represented in blue and DDLS in red. Values reported for ANOVA across the method factor are $df = 1$, $F = 7.22$, $p = .0075$. ANOVA values across dictionary length factor are $df = 3$, $F = 9.95$, $p \ll .01$. We also performed a repeated ANOVA across dictionary size factor for the two methods which reported a p–value$=1.73\times10^{-10}$, which confirms that dictionary size has a significant role.

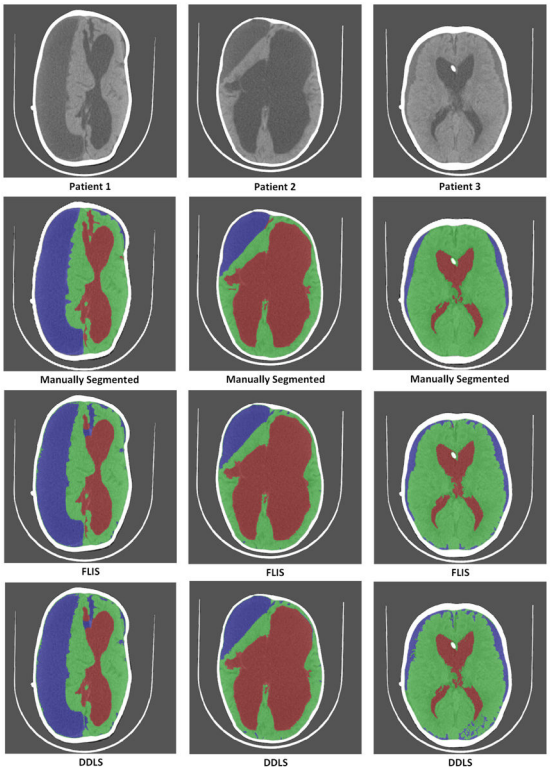

Fig. 10.

Comparison of results of the 2 methods for a dictionary size of 120 and training size of 17 patients. First row represents the original images of 3 patients. Second row represents their corresponding manually segmented image. Third row represents segmented images using FLIS. Fourth row represent segmented images using DDLS [32]. Green-Brain, Red-CSF, Blue-Subdurals.

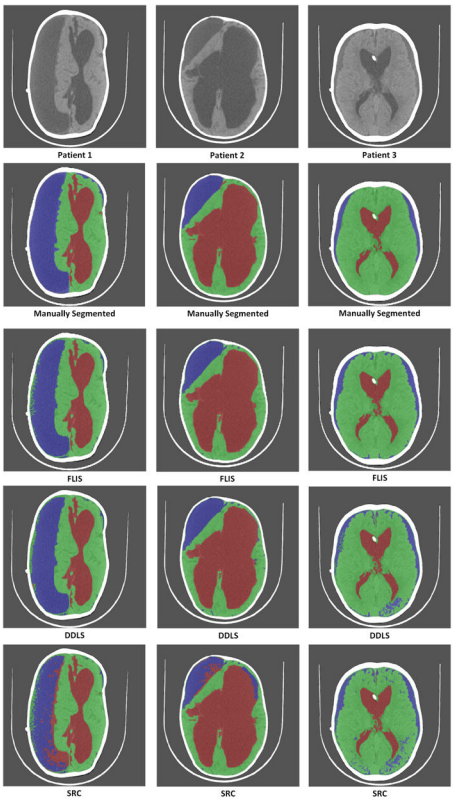

Fig. 11.

Comparison of results of the 3 methods for a training size of 17 patients. First row represents the original images of 3 patients. Second row represents their corresponding manually segmented image. Third row represents segmented images using FLIS. Fourth and Fifth rows represent segmented images using DDLS [32] and patch-based SRC [25] respectively. Green-Brain, Red-CSF, Blue-Subdurals.

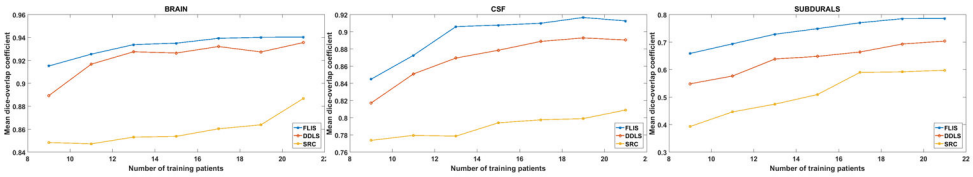

Fig. 12.

Comparing dice-overlap coefficients of FLIS with DDLS [32] and patch based SRC [25] for different sizes of training data.

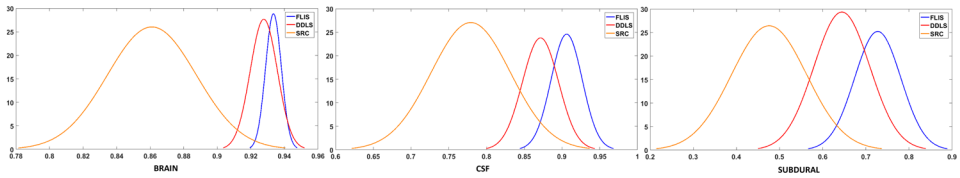

Fig. 13.

Gaussian fit for the histogram of dice overlap coefficients for ten random realizations of training data.

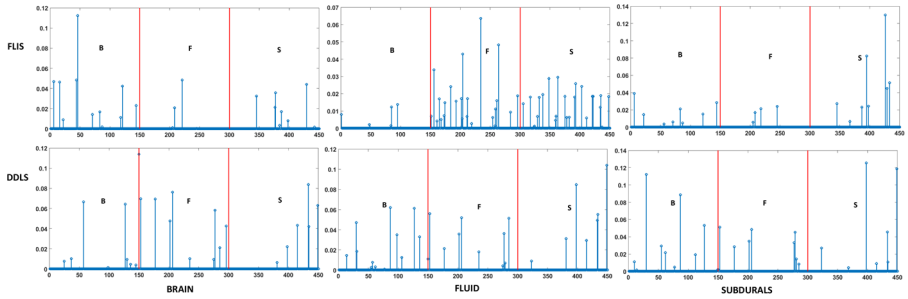

Fig. 14.

Comparing Sparse codes of a random pixel for brain (B), fluid (F) and subdurals (S). Row1: Sparse code for FLIS. Row2: Sparse code for DDLS. X axis indicates the dimension of the sparse codes. The left side of first red line correspond to brain, middle section corresponds to fluid and right side of second red line correspond to subdurals. Y axis indicate the values of the sparse codes.

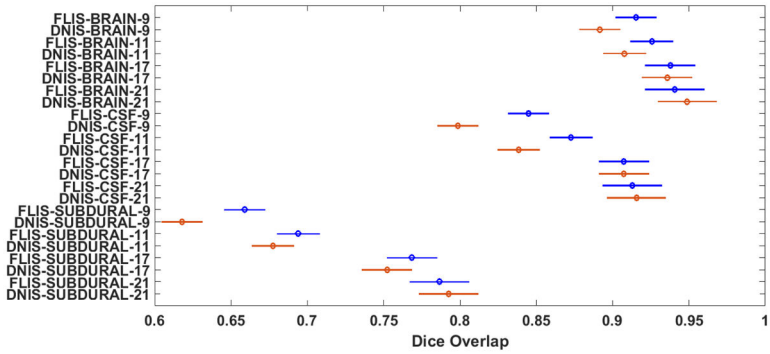

Fig. 15.

Comparison of FLIS with DNIS for different training configurations by using a 3-way ANOVA. The intervals represent the 95 percent confidence intervals of dice overlap values for a given configuration of method-class-training size. FLIS is represented in blue and DNIS in red. Values reported for ANOVA across the method factor are $df=1$, $F=35.54$, $p \ll .01$. ANOVA values across training size factor are $df=3$, $F=308.85$, $p \ll .01$.

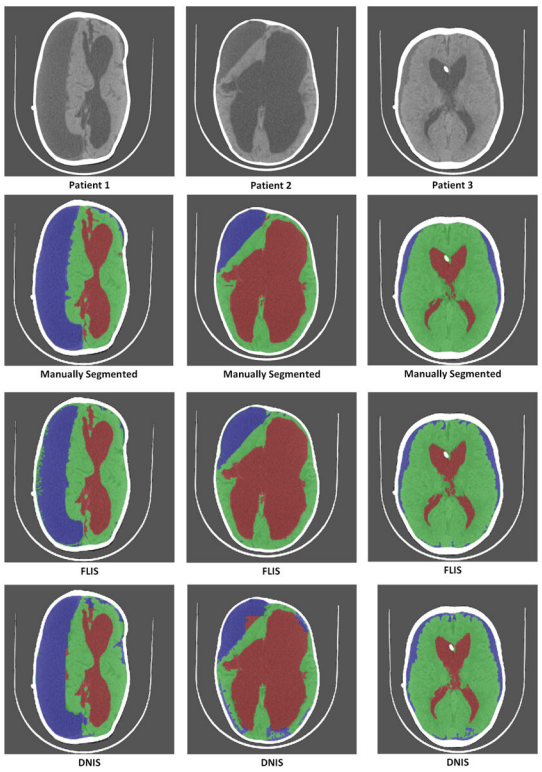

Fig. 16.

Comparison of results between DNIS and FLIS for training-test configuration of 17–15. First row represents the original images of 3 patients. Second row represents their corresponding manually segmented image. Third row represents segmented images using FLIS. Fourth row represent segmented images using DNIS. Green-Brain, Red-CSF, Blue-Subdurals.

TABLE I

Comparison of learning based method with traditional intensity based thresholding method. Values are reported in Mean±SD(standard deviation) FORMAT

Method	Brain	CSF	Subdural
BIS [11]	.580±0.21	.696±0.18	.226±0.14
Patch based SRC [25]	.885±0.15	.805±0.22	.496±0.28
DDLS [32]	.932±0.04	.892±0.08	.641±0.2
FLIS (our method)	.937±0.02	.908±0.07	.767±0.14

TABLE II

Performance of our method with different dictionary sizes. Values are reported in Mean±SD(standard deviation) FORMAT

Dictionary size	Method	Brain	CSF	Subdural
20	FLIS	.891±0.04	.833±0.12	.580±0.23
	DDLS [32]	.887±0.06	.827±0.12	.539±0.30
80	FLIS	.939±0.03	.907±0.07	.770±0.13
	DDLS [32]	.932±0.05	.892±0.08	.641±0.26
120	FLIS	.940±0.03	.906±0.07	.768±0.14
	DDLS [32]	.931±0.04	.890±0.07	.679±0.17
150	FLIS	.938±0.03	.911±0.07	.773±0.13
	DDLS [32]	.921±0.04	.891±0.08	.687±0.19

TABLE III

Complexity Analysis of methods

Method	Complexity	Run time	Est. Operations
DDLS	$\sim 9NK(2(\frac{d}{2}+3)+L^2)$	46.66 seconds	1.39×10^9
FLIS	$\sim \frac{9NK(2(d+3)+L^2)}{I_x \times I_y}$	.0003 seconds	1.005×10^4

TABLE IV

Memory requirements

Method	Memory(in bytes)	Approx Memory
SRC [25]	$\frac{d}{2} \times \frac{d}{2} \times N_t \times I_x \times I_y \times 16$	~ 9.2×10^{11} bytes
DDLS [32]	$(\frac{d}{2} + 3) \times 3K \times 16 \times I_x \times I_y$	~ 1.24×10^{11} bytes
FLIS (our method)	$(d + 3) \times 3K \times 16$	~ 4.8×10^5 bytes

TABLE V

Deep network configuration of DNIS. Note: Conv-Convolutional layer followed by a 2 × 2 Max pool Layer, FC-Fully connected layer

Patch Size	Layer1 (Conv)	Layer2 (Conv)	Layer3 (Conv)	Layer4 (FC)
25 × 25	24 5 × 5 × 1	32 3 × 3 × 24	48 3 × 3 × 32	256 nodes
50 × 50	24 7 × 7 × 1	32 5 × 5 × 24	48 3 × 3 × 32	256 nodes
75 × 75	24 9 × 9 × 1	32 7 × 7 × 24	48 5 × 5 × 32	256 nodes

TABLE VI

Performance of our method with DNIS. Values are reported in Mean±SD(standard deviation) format

Training samples	Method	Brain	CSF	Subdural	Training Time (in seconds)
9	FLIS	.915 ± 0.03	.845 ± 0.08	.660 ± 0.14	69.83
	DNIS	.890 ± 0.03	.80 ± 0.09	.632 ± 0.13	2860.66
11	FLIS	.926 ± 0.02	.873 ± 0.07	.694 ± 0.13	96.61
	DNIS	.910 ± 0.03	.834 ± 0.07	.671 ± 0.13	9464.34
13	FLIS	.934 ± 0.02	.906 ± 0.06	.729 ± 0.14	106.15
	DNIS	.919 ± 0.02	.880 ± 0.07	.690 ± 0.13	10443.57
15	FLIS	.935 ± 0.02	.908 ± 0.06	.750 ± 0.12	115.23
	DNIS	.934 ± 0.02	.897 ± 0.06	.728 ± 0.12	11823.99
17	FLIS	.939 ± 0.02	.910 ± 0.06	.770 ± 0.11	124.41
	DNIS	.939 ± 0.02	.908 ± 0.05	.752 ± 0.12	12940.41
19	FLIS	.940 ± 0.02	.917 ± 0.06	.786 ± 0.13	138.71
	DNIS	.943 ± 0.02	.914 ± 0.04	.786 ± 0.10	14669.76
21	FLIS	.940 ± 0.01	.913 ± 0.04	.786 ± 0.10	149.05
	DNIS	.950 ± 0.02	.919 ± 0.04	.792 ± 0.10	15846.87