Supervoxel Clustering with a Novel 3D Descriptor for Brain Tissue Segmentation

Yongfan Liu, Sen Du and Youyong Kong

Abstract—**Accurate segmentation of brain tissues from magnetic resonance imaging (MRI) is of significant importance for clinical application and scientific research. Traditional strategies to handle the 2D images have the limitation of 3D data. In this paper, to overcome these issues, a tissue segmentation approach with supervoxel clustering and the novel 3D texture extraction method are proposed. At first, the simple linear iterative clustering in three-dimension is applied, to reduce the number of calculation objects. Then, a novel local binary pattern in three-dimension is proposed for better discriminate the supervoxels with different tissues. A clustering approach is also developed to classify supervoxels with features into different types of tissues. The labels of supervoxel are finally mapped back to original data to have the tissue type of voxels. The performance of the proposed method is evaluated on the commonly utilized Internet Brain Segmentation Repository 18 dataset. The experiment showed promising results with insufficient trainset.**

Index Terms—**Magnetic resonance imaging, brain tissue, supervoxel, clustering, texture extraction, k-nearest neighbor.**

I. INTRODUCTION

Magnetic resonance images (MRI) provide abundant information associated to anatomical structures [1]. The powerful technique has been widely applied for examinations of human brain in clinical applications and scientific research [2]. Brain tissue segmentation is essential in the aforementioned applications [3]. Accurate tissue segmentation can be challenging due to partial volume effect, intensity non-uniformity and noise [4]. What's worse, because of the privacy and collection cost, the training data for MRI is very limited.

Plenty of methods have been proposed to obtain accurate tissue segmentation. Most of these methods utilize the voxel-wise strategy, which have a limitation of the relatively high computational burden for the high-resolution 3D MRI volume. Recently, the supervoxel technique has been employed to improve the efficiency with a sound performance [5]-[8]. This technique groups the similar voxel in the local region into a meaningful supervoxel, and the tissue segmentation is thus performed by processing a small number of supervoxel [6]. The supervoxel based strategy has

two advantages. At the first, supervoxel representation has strong spatial adherence to the tissue boundaries, which can prevent distorting the important anatomical details in MR images [7]. Secondly, supervoxel can enable low complexity and significantly simplify the subsequent tasks.

However, accurate tissue segmentation depends on developing discriminative features for describing the supervoxel. In the past two decades, great efforts have been made to develop a variety of descriptors [9]-[11]. Local binary pattern (LBP) is a commonly used texture feature extraction method [12]. However, these LBP and some of its improved algorithms are only suitable for 2D images. Therefore, it is very important to design an algorithm suitable for 3D data based on the LBP algorithm.

In this paper, we proposed a brain tissue segmentation approach by clustering supervoxels with novel three-dimensional local binary pattern operator and descriptor. At first, 3D supervoxels were generated for each MRI volume. Secondly, a novel 3D local binary pattern operator was proposed to characterize the supervoxel. Then a clustering approach is developed to classify supervoxels with features into different types of tissues. The labels of supervoxel are finally mapped back to original data to have the tissue type of voxels. With the help of this system, LBP_3D will give full play to its advantages and make up for its shortcomings. The performance of the proposed method is evaluated on the commonly utilized Internet Brain Segmentation Repository 18 dataset. The experimental results show that the MLBP-based algorithm system can train a good and robust kNN model with insufficient data.

II. METHODS

The objective of our algorithm system is to classify brain tissues from MRI. We proposed LBP algorithms in three-dimension (LBP_3D) at first, then we use supervoxel segmentation, SLIC_3D, as pre-processing, kNN as the predictive model, and construct an algorithmic system to achieve goals. The reason we don't use some more complex predictive model is that we want LBP_3D to make as more contributions as possible in this system, to reflect the performance of itself. If the more complex algorithms are used, it may introduce unnecessary factors. What's more, the SLIC_3D has finished the work of clustering, so it's not necessary to involve the cluster function in the predictive model. kNN is qualified to accomplish the discrimination task. The main framework of the proposed method is illustrated as Fig. 1 [13].

A. Supervoxel Generation

Supervoxel has been increasingly utilized for 3D brain MRI volumes. A large number of voxels can be transferred

for a small number of supervoxels, which can highly improve computational efficiency. Supervoxels essentially cluster voxels with similarities in features such as space and color. A number of supervoxel generation methods has been proposed in the past decade. Among these methods, simple linear iterative clustering (SLIC) algorithm is one of the most efficient and effective methods [7]. However, this algorithm is designed for 2D images, and it needs to be upgraded to 3D situation.

Oringinal Data

Supervoxel Clustering

Textrue Map Obtained by LBPa Operator

Textrue Map Obtained by LBPa Operator

LBP with Supervoxel Delineation

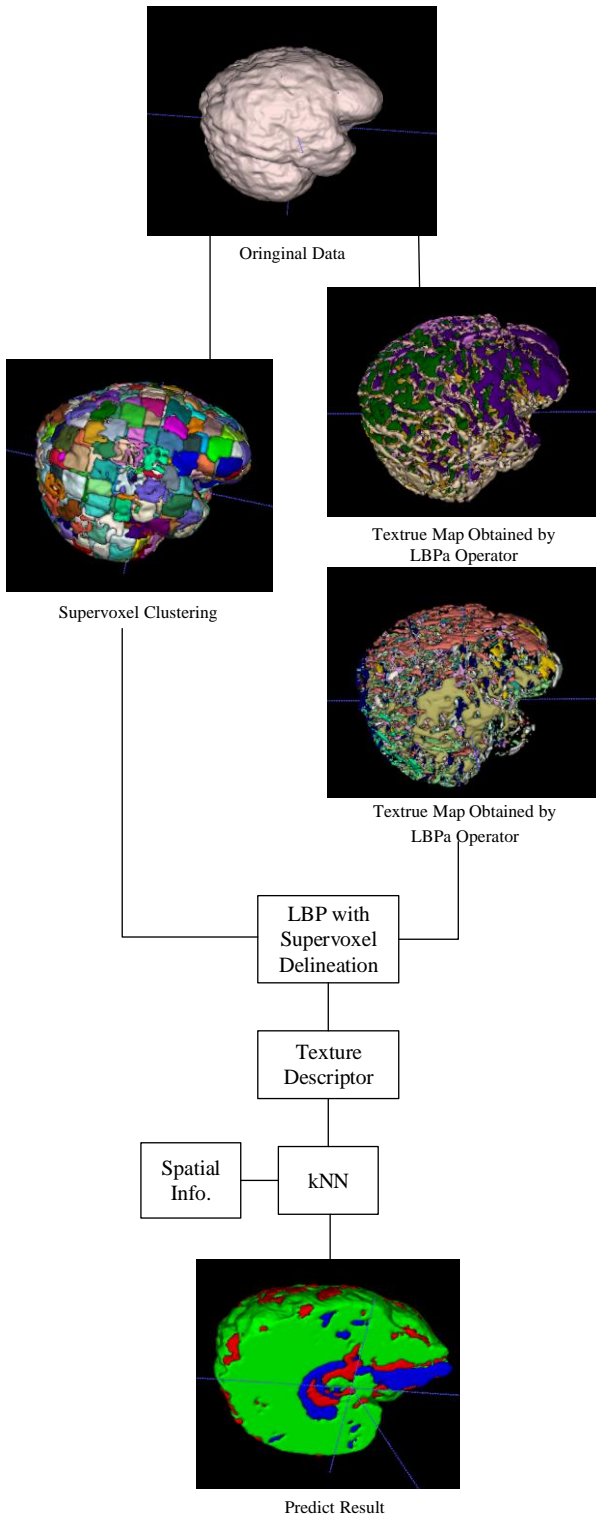Texture Descriptor

Spatial Info. — kNN

Predict Result

Fig. 1. The framework of the proposed brain tissue segmentation method.

Since there is only one channel per voxel in the MRI image, the calculations in the SLIC need to be modified. d_c is the

distance between the voxels (v_i, v_j) in the color space, and we have made a simplification; d_s is the distance between the voxels in the geometric space, and all three dimensions of it $(x_i, y_i, z_i$ or $x_j, y_j, z_j)$ need to be counted. m is to adjust the weight of two distances, and its value is determined according to the actual situation. D is the final distance.

$$d_c = v_j - v_i$$
$$d_s = \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2 + (z_j - z_i)^2} \qquad (1)$$
$$D = \sqrt{d_c^2 + \left(\frac{d_s}{S}\right)^2 m^2}$$

When we access the final distance D, the work about clustering will become easy. There is no much difference between 3D situation and 2D situation in the field of clustering method.

B. Local Binary Pattern in Three-Dimension

LBP (Local Binary Pattern) is a commonly used operator for extracting two-dimensional image features, which has excellent robustness in pattern recognition [12]. The original LBP operator is defined as a 3×3 window. In this window, the value of the center pixel is considered as the threshold, and the value of the adjacent 8 pixels need to compare with the threshold. If the value of a pixel is greater than the threshold, the position of this pixel is marked as 1, otherwise 0. In this way, 8 labels in the 3×3 window can produce an 8-bit binary number. Then it will be converted to a decimal number, called texture units, from 0 to 255. The texture units will be filled in the center pixel in the window, which is used to reflect the texture information for this area. Finally, a texture map is obtained, as the same size as the original image.

To better capture the 3D supervoxels, this paper proposed a novel LBP operator in three-dimension (**LBP_3D**). In three- dimensional space, 1 voxel has 26 neighbors. If all 26 neighbors are taken into consideration, there will be 2^{26} possible texture units, which is hard to process. Therefore, it's better to only select a part of the neighbors for calculation. Here, we have developed two types of operators. One is named **Local Binary Pattern of Adjoin Voxel (LBPa)**, where 1 voxel has 6 adjoin neighbors. Another one is named **Local Binary Pattern of Diagonal Voxel (LBPd)**, where the center has 8 diagonal neighbors. When we only consider the relationship between these specific neighbor voxels and the central voxel. LBPa only generates 2^6 possible texture units, while LBPd generates 2^8 possible texture units. The method of calculating the texture units by the LBPd operator is as following steps. The method of LBPa is similar.

<1> Select Voxel: Select a 3×3×3 window in the original 3D data. Take the voxel in the center of the window as the original point, and then take its eight diagonal voxels;

<2>Thresholding: Take the value X of the central voxel as a threshold. Compare the value xi of the 8 voxels obtained in step (1) with the threshold value. The ci is the output result.

$$c_i = \begin{cases} 1 & x_i \geq X \\ 0 & x_i < X \end{cases} \qquad (2)$$

Taking the value in Fig. 2 as an example, the central voxel

value is 16, and its eight diagonal neighbor values are 7, 33, 23, 24, 8, 11, 14, 12. If the value is greater than 16, take 1; or else take 0. The output result can be obtained: 0,1,1,1,0,0,0,0;

<3> Coding: Change the output result ci to the binary serial, then convert to decimal number.

$$\text{texture} = \sum_{i=1}^{8} 2^{c_i} \qquad (3)$$

Continue the example in Fig. 2, the texture unit of the center pixel is 112, which convert from 01110000;

<4>Move the 3×3×3 window, and the moving stride is 1 voxel. Traverse the entire three-dimensional data, so that all voxels, except the voxels at the edge of the data, have texture units. If specific texture units are given to the edge-voxels, a texture map as the same size as the original 3D data can be obtained. The specific texture unit is determined by the actual situation and it's 0 in this paper.
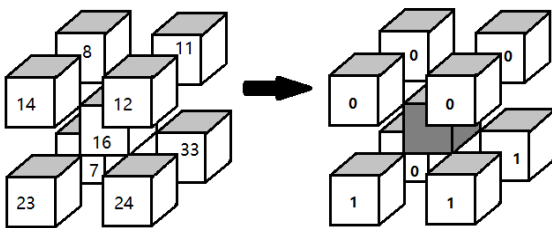

Fig. 2. Example of LBPd operator.

LBD_3D also faces a problem, that is, how to determine the order of the binary code. In the LBP algorithm, the binary codes are arranged in a clockwise order, so that the texture difference between two pixels, who have smaller texture value differences, is also small. That is to say, the texture units of LBP have a metric meaning. In the step <3>, the voxels around the central voxel are in 3D space, and there are no such concepts as "clockwise", "sequential", etc., therefore it's hard to confirm how to arrange the binary code. Still in the step <3>, if the binary serial is 00000111, it's totally fine.

C. kNN and Data Set Division

kNN (k-Nearest Neighbor) is a classic training-predictive model [14], which is a non-parametric method used for classification and regression. The input consists of the k closest training examples in feature space, and the output is the prediction of the label to the test set. As mentioned above, the characteristics of the MRI data can be summarized as follows: the amount of voxel in a single data is large, while the amount of data in the training set is small. The combination of supervoxel and kNN can make up for these two problems and finally solve them. This is why the kNN algorithm is chosen.

In the case of limited data sets, the composition of the training set needs to be carefully considered. If one three-dimensional data is used as one sample, as shown in (4), and $DATA_i$ is each MRI 3D data in the data set. It will be obviously not enough to train any machine learning model.

$$\textbf{\textit{TrainSet1}} = \{DATA_k\} \qquad (4)$$

Fortunately, because the 3D data contains tens of thousands of voxels, the number of supervoxels in a set of data is also very abundant (usually an MRI data can generate thousands of supervoxels). If one supervoxel can be used as

one training sample, as shown in (5), and *SV* is supervoxel, k is the number of MRI data in the data set, l is the number of supervoxels in one MRI data. It will become a very large training set.

$$\textbf{\textit{TrainSet2}} = \{SV_{kl}\} \qquad (5)$$

Now a new problem comes out. $DATA_i$ is similar to each other, but SV_{kl} is not. When k is a constant, SV_{kl} represents all supervoxels of a certain MRI data; when l is a constant, there aren't many correlations between every SV_{kl}. Therefore, if we want to use SV_{kl} as an element of the training set, we need to do some identification to prevent the prediction model from being confused with these supervoxels.

Thanks to the object we are dealing with is brain MRI data, which means most of the data are similar. In other words, in different MRI data, the data performances of the same area are approximately the same. So, in the first step of training, the spatial characteristic of supervoxel clustering center is very important. The machine will use spatial features to determine which super voxels should have similar characteristics and avoid global confusion.

The SLIC_3D has finished the work of clustering, so it's not necessary to involve the cluster function in the predictive model. kNN is qualified to accomplish the discrimination task.

Only spatial features are not enough. We need more features to help the machine judge. It should be noted that kNN accepts only one vector for one training data. Thus, the next task is to characterize each supervoxel feature as a one-dimensional vector.

D. Texture Descriptor and Feature Matching

When it comes to feature matching, the most easily thought of is the grayscale feature of each supervoxel. The grayscale histogram is used as one-dimensional data and is suitable for being sent to the kNN model for training. The method is: selecting supervoxels one by one, counting gray information of all voxels in supervoxels, and listing statistical histograms. After the grayscales of all supervoxels are counted, combine them with the spatial information of each supervoxel, as (6), and send it to the kNN model for training. Finally, the prediction result can be obtained.

$$\boldsymbol{v}^T = [\, x, y, z, \boldsymbol{H}^T \,] \qquad (6)$$

$\boldsymbol{v}$ is the eigenvector provided to kNN, $\boldsymbol{H}$ is the grayscale histogram of the voxel contained in the current supervoxel, and x, y, z are the spatial coordinates of this supervoxel. The same symbol has the same meaning in (7).

If a better predictive model is wanted, LBP_3D will be involved. As we know, the data returned by LBP_3D is a texture map, which is as the same size as the original 3D data. The texture map cannot be provided as a vector to kNN, so it needs to be converted into the eigenvector. Our approach is to measure the grayscale of the texture units in supervoxel range, called **Texture Descriptor**. Supervoxel range means the range of current supervoxel, the supervoxel segmentation is based on the original image. We combine the texture descriptor with the spatial information to get a new eigenvector, as (7). $\boldsymbol{T}$ is texture descriptor, the grayscale histogram of texture map.

TABLE I: PERFORMANCE OF BRAIN TISSUE SEGMENTATION ON IBSR18 DATASETS
THE UNITS OF ALL DATA IN THE TABLE ARE %, THE VAR (SA) ARE 10^{-4}

	DSC				Avg (SA)	Var (SA)
	CSF	WM	GM	Avg (DSC)		
HIST	51.35	78.02	58.27	62.55	71.46	7.69
LBPa	39.25	83.12	64.13	62.17	78.55	2.96
LBPd	39.37	83.79	79.45	67.54	79.57	**1.24**
MLBP	41.39	**84.10**	**79.66**	**68.38**	**79.94**	3.62
HIS_LBPa	50.98	79.49	61.86	64.11	73.26	2.67
HIS_LBPd	**51.41**	80.04	63.74	65.07	74.06	7.19
HIS_MLBP	49.67	80.58	65.24	65.15	74.68	4.34

$$v = \begin{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \times w \\ T \end{bmatrix} \qquad (7)$$

w is the weight of the spatial information. Because the spatial information only occupies three dimensions, it's weak for them to express themselves, the parameter w can solve it. w can be in the range [1, 10].

Since the data required by kNN is the texture descriptor, it's no longer important whether the texture units have the metric meaning. The texture descriptor counts the number of texture states in the current supervoxel. Thus, it's quite enough for the texture units have the ability to tell the difference from different texture states by their value.

The two methods LBPa and LBPd are independent of each other. They can work independently, or collaborate to extract features of MRI data, called **MLBP (Multi-Local Binary Pattern)**. MLBP is still a type of LBP_3D algorithm.

III. RESULTS AND DISCUSSION

A. Experimental Setup

The goal of this paper is to distinguish between tissues in the brain and segment them. The tissues can be divided into three parts, cerebrospinal fluid (CSF), white matter (WM), and gray matter (GM). All the proposed approaches have been evaluated on the widely utilized datasets: the internet brain segmentation repository 18 (IBSR18) dataset [15]. The IBSR dataset consists of 18 real MRI volumes derived from healthy subjects. Each volume consists of $256 \times 256 \times 128$ voxels with 2 mm³. 15 of the volumes form the training set, while 3 of them form the test set. The division is completely random. In order to ensure the certainty of the variables, all experiments are based on these division sets.

We have prepared 7 sets of experiments. As mentioned above, the LBPa, LBPd and the gray scale method are depending with each other. These 7 sets of experiments are the result of different combinations.

HIST: Only the grayscale features in each super voxel are used as training objects;

LBPa: Only the LBPa texture features and descriptor in each super voxel are used as training objects;

LBPd: Only the LBPd texture features and descriptor in each super voxel are used as training objects;

MLBP: Combine LBPa texture features and LBPd texture features in super voxels for training;

HIS_LBPa: Combine LBPa texture features and gray features in super voxels for training;

HIS_LBPd: Combine LBPd texture features and gray features in super voxels for training;

HIS_MLBP: Combine MLBP texture features and gray features in super voxels for training.

The number of partitions of SLIC_3D is 30000, m is 2; w is 2.5; the number of nearest neighbors in kNN is 5.

B. Comparison

We bring into two evaluation indexes for an objective evaluate of the segmentation results. One is the segmentation accuracy (SA) another is dice similarity coefficient (DSC). They are calculated as follow,

$$SA = \frac{TP}{TP+FN} \times 100\% \qquad (8)$$

$$DSC = \frac{2 \times TP}{2 \times TP+FP+FN} \times 100\% \qquad (9)$$

where *TP*, *FP* and *FN* are true positive, false positive and false negative voxel. The DSC value reflects the similarity between the segmentation result and the ground truth while SA only record the proportion of the voxel with the correct label in the segmentation result to the corresponding voxel in ground truth.

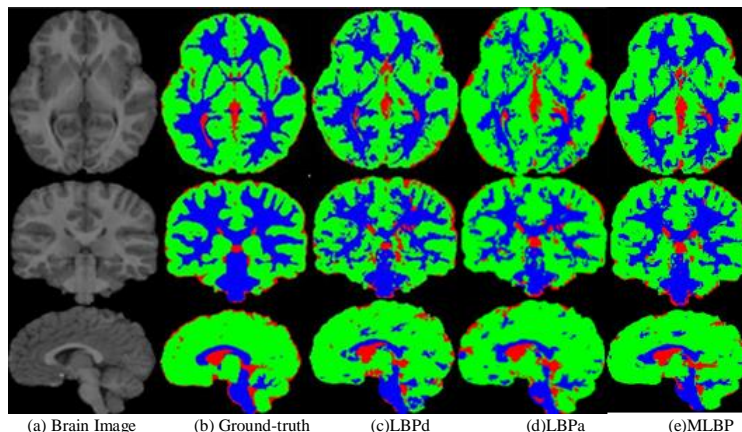(a) Brain Image   (b) Ground-truth   (c)LBPd   (d)LBPa   (e)MLBP

Fig. 3. The performance of our proposed segmentation algorithm on IBSR18. The color of red, green and blue represents CSF, GM and WM respectively. Column (a) and (b) are the original image and the ground truth of the segmentation. Column (c-e) are the segmentation results of the LBPd, LBPa and MLBP.

As three sets of data were selected as test sets, it is more objective to use the average of the three prediction results. In addition, the variance of the three prediction results should also be considered, as (10) shows, which is an important indicator to reflect the robustness of the algorithm system.

$$\sigma^2(SA) = \frac{\sum_{i=1}^{3}(\mu(SA) - SA_i)^2}{3} \qquad (10)$$

C. Result Analysis

Table I shows the result of the experiment, MLBP obtains better performance compared to other six methods. MLBP consists of LBPa and LBPd, so it's not surprises that MLBP obtains more precise segmentation result. The eigenvector of MLBP has more dimensions than LBPa and LBPd, which make more information be provided to the kNN for training and prediction.

The result in Table I suggests that the variance of LBPd is the least, which means LBPd is the most robust in this experiment. The truth is LBPd indeed has good performance as well. When the *w* is 1, LBPd will be the best algorithm in these 7 algorithm systems, both average and variance. Also, when the *w* more than 1, the MLBP has more advantages, and most of its parameters are better than others.

The success of MLBP doesn't mean that more information is more effective, cause the HIS-MLBP gives out the counter-example. According to the experiment, we can draw a conclusion, that is, grayscale histogram of the voxel is not an ideal method to extract the feature of MRIs and segment them. The average result of HIST is the lowest, and the variance of it is the highest. There must be some error information in HIST. When combining HIST and LBP_3D, the accuracy comes down totally, comparing with LBP_3D working alone.

D. Reliability

According to the results in Table I and the analysis in Section III.C, LBPd and MLBP have a sound performance. To further illustrate the reliability of the two methods in the prediction process, cross-validation is performed.

In the previous experiment, the calculated variances were just rough verification. It's reasonable to suspect that the variances will fluctuate as the training set changes. As mentioned above, the proposed algorithm is aimed at the general shortage of medical image data, and the experiment is also carried out with only 18 sets of data. For this situation, leave-one-out cross-validation is the best option [16].

In practice, leave-one-out cross-validation means 18 groups of experiments will be conducted for each algorithm. In each experiment, 17 sets of data will be taken as the train set and the rest one set of data will become the test set. 18 groups of experiments are to make sure that every data set has the chance to be the test set. Referring to the settings in III.B, each set of experiments will produce 4 results, they are SA, CSF, WM and GM. According to the output of the four types of results, draw grouping diagrams and calculate their variance, as shown in (11). *res* is the 4 types results and $\mu(\bullet)$ is the average of each type of the result.

$$\sigma^2(res) = \frac{\sum_{i=1}^{18}(\mu(res) - res_i)^2}{18} \qquad (11)$$

The performance of the two algorithm is shown as Fig. 4, and the statistical results are shown in Table II. As it can be seen, the averages of LBPd and MLBP are almost the same, the difference between them is nuance as well in Fig. 4. While the variances of MLBP are totally less than LBPd's, which means MLBP is more reliable. According to the Fig. 5, both algorithms have smooth circle, so they are both stable prediction models. The only flaw is the division of the CSF. Because cerebrospinal fluid is liquid and has a narrow distribution in the brain, its segmentation effect is relatively poor.

TABLE II: THE STATISTICAL RESULTS OF VALIDATION
THE UNITS OF ALL AVG DATA IN THE TABLE ARE %, THE VAR ARE 10^{-4}

		LPBd	MLBP
SA	Avg	86.99	87.12
	Var	12.11	10.12
CSF	Avg	65.09	65.27
	Var	57.41	53.06
WM	Avg	89.40	89.54
	Var	9.38	7.75
GM	Avg	87.66	87.68
	Var	12.82	12.16

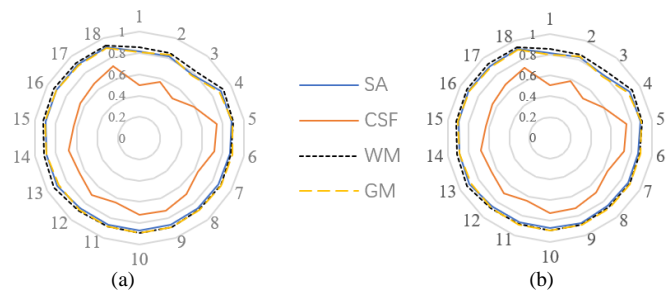

Fig. 4. The result of 18 groups of experiments, for each algorithm. Diagram (a) is for LBPd, and diagram (b) is for MLPB.

IV. CONCLUSIONS

In this paper, we have proposed a novel LBP_3D operator and texture descriptor, which bring us a new idea to handle with 3D MRI data. Moreover, we introduced a brain MRI image segmentation method to make good use of them, based on supervoxel and kNN. The proposed MLBP algorithm system performed better in the same series of algorithms.

On a regular computer, this method can process, train, and predict all data in less than five minutes, which means it has good universality and can be used on most computers. We are confident that when the number of training sets is larger, more accurate results can be achieved. In addition, the LBP_3D descriptor in this paper is a kind of compensation for which the coding order of LBP_3D is not fixed. In fact, most organs are symmetrical. If the coding order can correspond with these symmetries, or other novel method to handle the coding order is proposed, its ability to extract the feature of MRI data will be stronger.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

AUTHOR CONTRIBUTIONS

Yongfan Liu conducted the research, experiment, and edited part of this paper. Sen Du launched the research, contributed his ideas and edited part of this paper. Youyong

Kong supervised the research, provided vital guidance, and was responsible for the corresponding of this paper; all authors had approved the final version.

REFERENCES

[1] K. Kim, P. A. Habas, F. Rousseau *et al.*, "Intersection Based Motion Correction of Multislice MRI for 3-D in Utero Fetal Brain Image Formation," *IEEE Transactions on Medical Imaging*, vol. 29, no. 1, pp. 146-158, 2009.

[2] A. Mayer and H. Greenspan, "An adaptive mean-shift framework for MRI brain segmentation," *IEEE Transactions on Medical Imaging*, vol. 28, no. 8, pp. 1238-1250, 2009.

[3] K. Hassan, C. Olivier, A. Jamal, B. Isabelle, "3D brain tumor segmentation in MRI using fuzzy classification, symmetry analysis and spatially constrained deformable models," *Fuzzy Sets & Systems*, vol. 160, no. 10, pp. 1457-1473, 2009.

[4] L. P. Clarke, R. P. Velthuizen, M. A. Camacho, and J. J. Heine "Segmentation: Methods and applications," *Magnetic Resonance Imaging*, vol. 13, no. 3, pp. 343–368, 1995.

[5] H. Ma, R. Lei, J. Sun, and Y. Kong, "Multi-Session Parcellation of the Human Brain Using Resting-State fMRI," in *Proc. 2018 IEEE 22nd International Conference on Computer Supported Cooperative Work in Design ((CSCWD))*, Nanjing, 2018, pp. 336-340.

[6] N. Verma, M. C. Cowperthwaite, and M. K. Markey, "Superpixels in brain MR image analysis," *Engineering in Medicine & Biology Society*, 2013.

[7] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk "SLIC superpixels compared to state-of-the-art superpixel methods," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 34, no. 11, pp. 2274-2282, 2012.

[8] Y. Kong, Y. Deng, and Q. Dai, "Discriminative clustering and feature selection for brain MRI segmentation," *IEEE Signal Processing Letters*, vol. 22, no. 5, pp. 573-577, 2015.

[9] J. Reynolds and K. Murphy, "Figure-ground segmentation using a hierarchical conditional random field," in *Proc. Conference on Computer & Robot Vision*, 2007.

[10] W. Jie, "Image segmentation using situational DCT descriptors," in *Proc. International Conference on Image Processing*, 2001.

[11] R. Shanker, R. Singh, and M. Bhattacharya, "Segmentation of tumor and edema based on K-mean clustering and hierarchical centroid shape descriptor," in *Proc. IEEE International Conference on Bioinformatics & Biomedicine*, 2017.

[12] T. Ojala, M. Pietikainen, and D. Harwood, "Performance evaluation of texture measures with classification based on Kullback discrimination of distributions," *Pattern Recognition*, vol. 1, 1994.

[13] P. A. Yushkevich, J. Piven, H. C. Hazlett, R. G. Smith, S. Ho, J. C. Gee, and G. Gerig, "User-guided 3D active contour segmentation of anatomical structures: Significantly improved efficiency and reliability," *Neuro Image*, vol. 31, issue 3, pp. 1116-1128, July 2006.

[14] N. S. Altman, "An introduction to kernel and nearest-neighbor nonparametric regression," *The American Statistician*, vol. 46, no. 3, pp. 175-185, 1992.

[15] IBSR. [Online]. Available: https://www.nitrc.org/projects/ibsr/

[16] S. J. Sheather, "Density estimation," *Statist. Sci.*, vol. 19, no. 4, pp. 588-597, 2004.

Yongfan Liu is a student of Chien-shiung Wu College, Southeast University, which is the Honor College. He is going to receive his bachelor degree this academic year, as well as the honor degree. Now he is with the Division of Continuing Education, University of California, Irvine, and preparing for the further study. His research interest is computer vision and machine learning. In this project he worked under the supervision of Dr. Kong, who is the corresponding author of this article.

Sen Du is a student of School of Computer Science and Engineering, Southeast University, China. He is a senior majoring on computer science and technology. His research interest including computer network, distributed system and digital image.

Youyong Kong is currently an associate professor with the College of Computer Science and Engineering, Southeast University. He received the B.S. and M.S. degrees in computer science and engineering from Southeast University, China in 2004 and 2008, and the Ph.D. degree in imaging and diagnostic radiology from The Chinese University of Hong Kong, Hong Kong, China, in 2014, respectively. His research interests include image processing, machine learning and neuroscience.