Computers and Electrical Engineering 69 (2018) 768–781

Contents lists available at ScienceDirect

Computers and Electrical Engineering

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

Ensemble-classifiers-assisted detection of cerebral microbleeds in brain MRI☆

Tayyab Ateeq[a], Muhammad Nadeem Majeed[a], Syed Muhammad Anwar[a], Muazzam Maqsood[b], Zahoor-ur Rehman[b], Jong Weon Lee[c], Khan Muhammad[d], Shuihua Wang[e], Sung Wook Baik[d], Irfan Mehmood[c,*]

[a] Department of Software Engineering, University of Engineering and Technology, Pakistan
[b] Department of Computer Science, COMSATS Institute of Information and Technology, Attock Campus, Pakistan
[c] Department of Software, Sejong University, Seoul, Republic of Korea
[d] Intelligent Media Laboratory, Digital Contents Research Institute, Sejong University, Seoul, Republic of Korea
[e] Department of Electrical Engineering, The City College of New York, CUNY, New York, NY 10031, USA

ARTICLE INFO

Article history:
Received 9 October 2017
Revised 11 February 2018
Accepted 13 February 2018
Available online 7 March 2018

Keywords:
Cerebral Microbleeds
Support Vector Machine
Quadratic Discriminant Analysis
Ensemble classifier
Susceptibility-Weighted Imaging

ABSTRACT

Cerebral Microbleeds (CMBs) are considered as an essential indicator in the diagnosis of critical cerebrovascular diseases such as ischemic stroke and dementia. Manual detection of CMBs is prone to errors due to complex morphological nature of CMBs. In this paper, an efficient method is presented for CMB detection in Susceptibility-Weighted Imaging (SWI) scans. The proposed framework consists of three phases: i) brain extraction, ii) extraction of initial candidates based on threshold and size based filtering, and iii) feature extraction and classification of CMBs from other healthy tissues in order to remove false positives using Support Vector Machine, Quadratic Discriminant Analysis (QDA) and ensemble classifiers. The proposed technique is validated on a dataset of 20 subjects with CMBs that consists of 14 subjects for training and 6 subjects for testing. QDA classifier achieved the best sensitivity of 93.7% with 56 false positives per patient and 5.3 false positives per CMB.

1. Introduction

Cerebral Microbleeds (CMBs) are small brain hemorrhages that are caused by the leakage of blood from blood vessels [1]. These CMBs are also known as hemosiderin deposits [2]. The presence of CMBs and their patterns are considered as a crucial indicator of the cerebrovascular disorder. Recent studies show that CMBs are the root cause of many other brain disorders like intracerebral brain hemorrhage and stroke [34]. Aside from causing these vascular abnormalities, CMBs can also damage their neighboring tissues causing cognitive disabilities like dementia [2]. CMBs appear as small circular shaped spots [1]. They can be visualized through advanced Magnetic Resonance Imaging (MRI) techniques like Susceptibility-Weighted Imaging (SWI) and T2* weighted gradient-recalled echo. The SWI scans outperform T2* weighted sequences because CMBs are more clearly outlined in SWI [56].

Early detection of CMBs is very important because they are indicators of multiple life-threatening brain disorders. Therefore, a sound and reliable technique for the detection of CMBs is critical, which may assist neurologists to suggest medicines

☆ Reviews processed and recommended for publication to the Editor-in-Chief by Guest Editor Dr. Y. Zhang.

* Corresponding author.
E-mail addresses: irfan@sejong.ac.kr, irfanmehmood@ieee.org (I. Mehmood).

https://doi.org/10.1016/j.compeleceng.2018.02.021
0045-7906/

and anticipate the best solution. With the advancement in medical imaging techniques, minor organs in the brain can be visualized with more details that facilitate the diagnosis of CMBs [7]. Generally, manual inspection is performed to annotate CMBs that is based on visual analysis. It can be time consuming and monotonous task especially when there are a large number of subjects. It takes an observer about 30 min to detect all CMBs in a single Magnetic Resonance scan [8]. Manual detection of CMBs is also prone to errors due to their small size. They can be easily confused with other structures in brain Magnetic Resonance Imaging (MRI) such as blood vessels, calcium deposits etc. Moreover, as the number of CMBs increases, the time consumed to detect them manually also increase.

On the other hand, an automated procedure of CMB detection can achieve high sensitivity and reduce the workload for annotator. However, automating the detection of CMBs is a challenging task due to the similar morphological properties of CMBs and other structures in the brain, large variation in the size of CMBs and their widespread distribution in MRI scans. Many automated techniques proposed for CMBs detection are centered around native features based on size, intensity, and geometry of CMBs. Fazlollahi et al. [9] implemented radon transform to extract the local shape-based features of CMBs. However, only radon transform is not sufficient to capture complicated nature of CMBs. Kuijf et al. [10] used the Radial Symmetry Transform (RST) to detect CMBs. The accuracy of this system was not satisfactory. Therefore, it is not suitable for medical practices In order to differentiate CMBs from CMB mimics more accurately, Bian et al. [11] performed 2D fast RST to compute the geometric based features of CMBs but the results were not promising for this technique. Ghafaryasl et al. [12] extracted shape and intensity based features for CMB classification. However, their technique requires manual review for removal of false positives that is time-consuming. To make the automated process more efficient, some researchers, first, removed non-CMBs regions and then extracted a small number of candidates for classification based on some features [13,14]. However, the structure of these native features is dependent on the domain knowledge of CMBs. In addition, these features are not sufficient to capture the complicated nature of CMBs.

In this context, an automated technique is proposed for the detection of CMBs in SWI MRI scans. The objective is to develop a system that will be able to achieve high sensitivity and minimize the CMBs mimics. To overcome the limitations of existing techniques, we have explored a wide variety of features involving a total of 29 features. Out of these 29 features, 6 features are based on binary patches and 23 features are based on grayscale patches. These features are based on size, geometry and transform that differentiate CMBs from other similar structures. Some of these features haven't been used in the previous automated systems for CMB detection. To check the effectiveness of the proposed feature vector, three classifiers: Support Vector Machine (SVM), Quadratic Discriminant Analysis (QDA) and Ensemble classifiers are used. The proposed technique consists of three steps. (1) Preprocessing for brain extraction (2) Region of Interests (ROIs) extraction using statistical thresholding and hole filling. The extracted ROIs include both true CMBs and false positives. (3) Classification using SVM, QDA and ensemble classifiers to remove false positives. The proposed technique is validated on a dataset of 20 subjects. These 20 subjects have 167 CMBs in total. The training of classifiers is carried out with 14 subjects (104 CMBs) and testing is carried out with 6 subjects (63 CMBs). The proposed technique has the following contributions

1. A wide variety of features based on binary and grayscale patches are explored for CMB detection.
2. The proposed system achieves high sensitivity and also reduces the false positive rate considerably low.
3. Classification of CMBs is achieved without any post-processing step that requires manual review for removal of CMB mimics.

The remaining paper is organized as follows. In Section 2 we discuss previous techniques for CMBs detection, their effectiveness, and drawbacks. In Section 3 methodology is described in detail followed by Section 4 in which experimentation and results are reported.

2. Literature review

Many researchers have proposed automated techniques for CMBs detection using different classification algorithms. Most of the proposed techniques are a combination of image segmentation with features extraction and classification. Seghier et al. [15] proposed a unified probabilistic normalization segmentation technique using a mixture of Gaussians to detect CMBs in T2-Weighted MRI scans. Their proposed technique successfully detected CMBs in 8 out of 13 patients who had a single CMB and detected all CMBs in 9 patients who had two or more CMBs. Therefore, this method works well only for those patients that have multiple CMBs but did not work well for the patients having single CMB. Kuijf et al. [10] proposed a technique that computes the RST of MRI scans to detect CMBs. They achieved 65%, 78% and 84% sensitivities with an average of 20, 49 and 96 false positives per patient. Although their proposed system was reasonably efficient in terms of sensitivity their technique missed a significant number of CMBs. Barnes et al. [13] proposed a technique in which thresholding is applied first then shape and intensity based features were extracted and SVM classifier was used to remove false positives. They achieved 81.7% of sensitivity with 107 false positives per subject. Their system was not completely automated as they required to perform a manual review for the removal of leftover false positives. The manual removal of CMBs need human expertise and also consumes a lot of time.

Ghafaryasl et al. [12] used a large number of features based on local image descriptors and used four classifiers; QDC, Linear Discriminant Analysis (LDA), SVM, and Parzen classifier for CMB detection. Their system achieved high sensitivity but also produced a large number of false positives. Their system also requires manual review for the removal of false positives. Bian et al. [11] computed 2D fast RST to find geometric feature measurements followed by removal of false positives. They

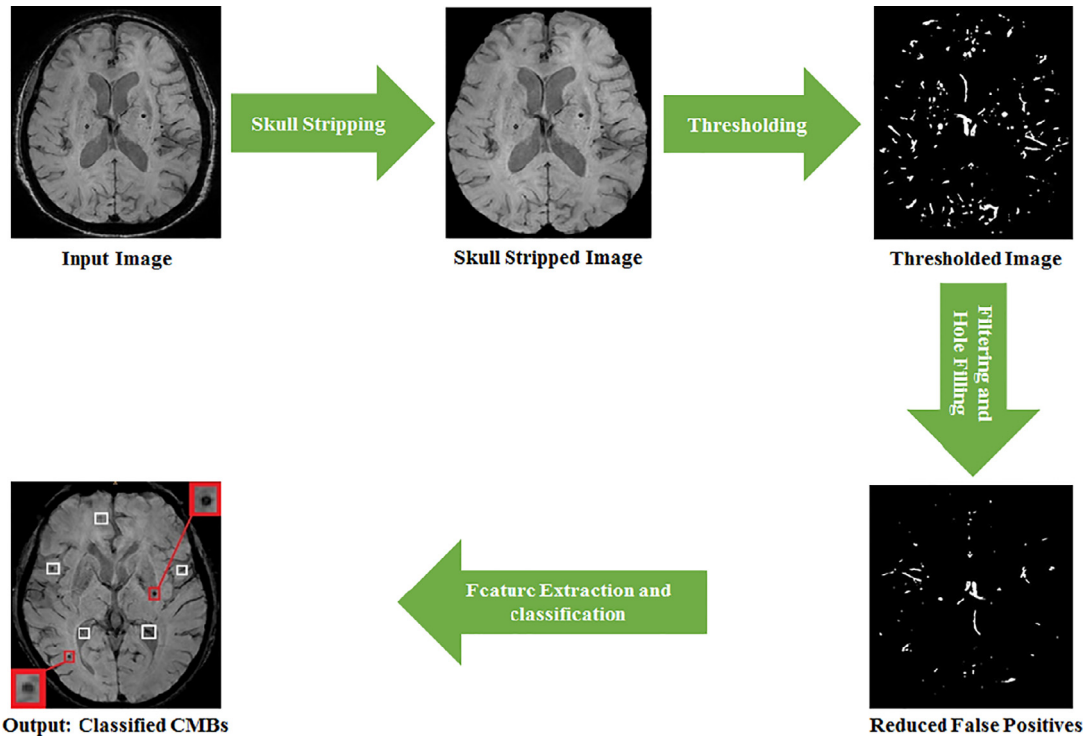

Fig. 1. Schematic overview of proposed framework for CMBS detection using binary and grayscale features.

achieved a sensitivity of 86.5% and an average 45 false positives per subject that is very high Fazlollahi et al. [9] extracted Radon-based features to use the shape information of CMBs. Random Forest (RF) were used to classify the CMBs. They achieved high sensitivity with a large number of false positives which indicate that only radon transform is not sufficient to capture complex nature of CMBs. Dou et al. [16] proposed a technique using threshold and 3D feature extraction from an unsupervised convolutional Independent Subspace Analysis (ISA) network followed by the classification to remove false positives. This technique achieved performance with the recall of 89.44% and an average of 7.7 false positives per subject. Van Den Heuvel et al. [17] extracted intensity and local shape-based features and removed the false positive using RF classifier. They achieved a sensitivity of 89% with 25.9 false positives per subject. The limitation of this technique is that only one expert annotated CMBs in all patients so some CMBs might have been missed.

Some researchers have also used neural networks in their techniques for CMBs detection. Chen et al. [18] used threshold and feature extraction using convolutional neural networks and removal of false positives using classification with SVM classifier. This technique achieved 90% accuracy and average of 6.4 false positives per subject. The limitation of this technique is the expensive computational cost of convolutional neural networks which limits its use in medical practice. Dou et al. [19] detected CMBs using 3D Fully Convolutional Network (FCN) model which produce a 3D score volume and then the 3d convolutional neural network is used to remove false positives. This technique produced a sensitivity of 93.16% with 2.74 false positives per patient. This technique was efficient as compared to other neural networks based techniques but it is still slow as compared to classifiers based technique.

However, all these systems face some limitations like high false positive rate, low sensitivity and manual removal of leftover false positives after classification. Therefore, an efficient and fully automated technique is required for the detection of CMBs. In this paper, we have explored a large number of diverse natured features to cover all the aforementioned limitations. Three machine learning classifiers are used to classify CMBs. The proposed system provides a fully automated detection of CMBs without involving manual leftover false positives.

3. Methodology

Most of the automatic CMB detection methods mainly use a limited number of features based on shape, size and intensity information. These features are not sufficient to capture complex nature of CMBs. As compared to these aforementioned techniques, the proposed method explored a larger set of features to achieve better performance. The proposed automated technique for CMB detection system consists of three phases, skull stripping, initial candidate selection, feature extraction and classification using SVM, QDA and ensemble classifiers. The overview of the proposed methodology is shown in Fig. 1. The first step of the methodology is skull stripping followed by extraction of initial candidates based on threshold and hole

Table 1
Overview of features.

Features	Description
Area	Total on pixels in a binary patch of a CMB
Maximum box length	Maximum distance between two pixels in a binary slice of a CMB
Forward Walsh Hadamard transform	Geometry of a CMB
Orientation	Represent structure of a CMB
Fourier transform	Show geometry of CMB
Difference of Gaussians	Differentiate CMBs from false positives based on edges
Median of intensity values	Mean of intensity values of a binary patch of a CMBs
Variance	Differentiates CMB from false positive based on how each pixel varies from its neighboring pixel
Determinant of hessian	Represent shape of CMB
Eigenvalues (e1, e2)	Represent shape of a CMB
Trace of hessian	Differentiate CMBs from false positives based on shape
Karhunen Loeve Expansion	Represents size of CMBs
Gabor Transform	Geometry based feature which differentiates CMBs from false positives
Hough Transform	Represent shape of CMBs
Image integral	Geometric based feature
Mean of intensity values	Mean of intensity values CMB
Minimum intensity value	Minimum intensity value in a greyscale patch of a CMB
Radon transform	Shape-based feature of CMB
Wavelet Transform	Geometry based feature of CMB
Determinant of hessian x hessian	Shape-based feature of CMB
Trace of hessian x hessian	Local shape-based feature of CMB

filling technique. After selecting the candidate regions, features are extracted to train the classifiers. Three classifiers- SVM, QDA and ensemble classifiers are then used to discriminate the true CMBs from false positives.

3.1. Preprocessing

The main reason behind high false positive rate in CMB detection is the complex nature of skull in MRI scans. The classification step often confuses parts of skull with CMBs. Therefore, proper handling of skull removal in preprocessing step is highly important for an efficient CMB detection system. An efficient removal of the skull in preprocessing step results in considerable reduction in false positives. Unlike previous techniques, the proposed technique does not require much preprocessing. The preprocessing step is only limited to skull-stripping from T2*-weighted images using the BrainSuite Tool [20].

3.2. Initial candidate selection

The preprocessed image after removing the skull is then used for initial candidate selection. A threshold is used for SWI image to convert it into a binary image that contains a lot of connected components including CMBs and false positives. In order to reduce the ROIs, two techniques are applied separately on this binary image. In the first technique, all connected components are filtered based on their size to remove extremely large objects. In the second technique, the hole filling is applied to remove unrealistically large objects. Then, the results of these two aforementioned techniques are combined to make sure that none of the CMBs are unintentionally missed. This results in initial candidates for CMBs after removing the large objects from MRI images.

3.3. Feature extraction

The candidates resulted from the previous step were then subjected to feature extraction. The feature vector will consist of geometric, intensity, transformation and local shape-based features. A total of 29 features were extracted from candidates involving features from both binary and greyscale patches. These features are explained based on their categories are given in Table 1. The brief description of feature categories is given here.

1) *Local shape-based features*

Area of a binary image is the total number of pixels in the whole image, but it might not be exactly the same because different patterns of pixels are weighted differently. It is calculated using Eq. (1).

$$A_C = \sum_{p=1}^{P_C} k(p-1)\Delta L(p) \tag{1}$$

where P_C is the perimeter of enclosed object and A_C is area of connected component.

$$k(p) = \sum_{i=1}^{p} \Delta k(i) \tag{2}$$

Maximum box length in a binary image is the maximum distance between two pixels in a binary slice and can be calculated using Eq. (3).

$$Max\ box\ length = \max\left(\sqrt{(y_2 - y_1)^2 + (x_2 - x_1)^2}\right) \tag{3}$$

where (x_1, y_1) and (x_2, y_2) are coordinates of two pixels. Orientation specifies the angle between the *major axis* and the *x*-axis of the blob in a binary image. The difference of Gaussian is calculated by taking the difference of two Gaussian filters then convolving the result with input image using Eq. (4). The concept of Gaussian is generally used for edge detection but here it is used as a feature for classification.

$$C\ (m,\ n) = \sum_x \sum_y A(x,\ y)B(m - x + 1,\ n - y + 1) \tag{4}$$

where A and B are convolving signals, x and y run over all values that lead to legal subscripts of A(*x*,*y*) and B(m − x + 1,n − y + 1).

Median of intensity values is M and it is calculated using Eq. (5).

$$M = median(A) \tag{5}$$

where A represents an input image. Variance specifies difference of the elements in a vector as shown in Eq. (6).

$$V = \frac{1}{K-1} \sum_{x=1}^{K} |A_x - \mu|^2 \tag{6}$$

where A is a vector made up of K observations and μ is mean of A_x which is an input image as shown in Eq. (7).

$$\mu = \frac{1}{K} \sum_{x=1}^{K} A_x \tag{7}$$

Hessian matrix is a square matrix shown in Eq. (8).

$$H = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1\ x_2} \\ \frac{\partial^2 f}{\partial x_2 x_1} & \frac{\partial^2 f}{\partial x_2^2} \end{bmatrix} = \begin{bmatrix} I1 & I2 \\ I3 & I4 \end{bmatrix} \tag{8}$$

Six features were computed from a hessian matrix including determinant of Hessian matrix, trace of Hessian matrix, trace of hessian *x*. The Hessian and determinant of hessian *x* hessian. Eigenvalues of hessian were computed using Eqs. (9) and (10).

$$Eigen\ Value\ 1 = \frac{\left(I1 + I4\ +\ \sqrt{(I1 - I4)^2 + 4I2^2}\right)}{2} \tag{9}$$

$$Eigen\ Value\ 2 = \frac{\left(I1 + I4 -\ \sqrt{(I1 - I4)^2 + 4I2^2}\right)}{2} \tag{10}$$

The values of $I1$, $I4$, and $I2$ are calculated from Eq. (8). The next two features are based on intensity values; minimum intensity and mean of intensity values. Image integral [21] calculate the summations over image sub-regions. Every pixel in an integral image is the summation of the pixels to the left and above.

2) *Transformation-based features*

The forward Walsh-Hadamard transform [22] is an orthogonal, non-sinusoidal transformation technique that breaks a signal into a set of Walsh functions. These Walsh functions are square or rectangular waves with values ranging from −1 to +1. The forward Walsh Hadamard transform is calculated using Eq. (11).

$$y_m = \frac{1}{N} \sum_{k=0}^{N-1} x_k WAL(m, k),\ \ m = 1,\ 2,\ \ \ldots,\ N - 1 \tag{11}$$

where N is the length of the input signal and x is the input signal. The Discrete Fourier Transform (DFT) converts an equally-spaced sampled finite sequence of a function into an equivalent-length sequence of equally-spaced samples of the Discrete-Time Fourier Transform (DTFT). The 2 dimensional Fourier transform is used in the processing of 2-D signals and data such as images. It can be calculated using Eq. (12)

$$Y_{a+1,\ b+1} = \sum_{s=0}^{x-1}\sum_{t=0}^{y-1} w_x^{sa} w_y^{tb} X_{s+1,\ t+1} \tag{12}$$

where w_x and w_y are complex roots of unity and can be calculated using Eqs. (13) and (14).

$$w_x = e^{-2\pi i/x} \tag{13}$$

$$w_y = e^{-2\pi i/y} \tag{14}$$

where i is the imaginary part, a and s are the indices that run from 0 to $x - 1$. While b and t are the indices that run from 0 to $y - 1$. This formula shifts X and Y by 1 in order to reflect matrix indices.

Wavelet transform [23] computes a single-level two-dimensional signal decomposition. The Radon Transform is the projection of the image intensity along a radial line oriented at a specific angle. Hough transform is generally used to detect lines in an image by utilizing a parametric representation of a line and due to this nature, we have used it as a feature in order to differentiate line from circular edges. It can be calculated using Eq. (15)

$$rh = a*\cos(theta) + b*\sin(theta) \tag{15}$$

where rh represent the distance from the origin to the line along a vector perpendicular to the line and theta represents the angle between this vector and the x-axis. While a and b represents parametric representation of the line.

Gabor Transform is used to generate the phase and magnitude response of a Gabor filter for the given grayscale image. The output of a Gabor transform is magnitude and phase of the Gabor filter. *Karhunen–Loeve Transform (KLT)* is closely related to the *Principal Component Analysis* (*PCA*) and it is frequently used in the analysis of data in multiple fields. It can be calculated using Eq. (16).

$$KLT = V' * y' \tag{16}$$

where V' is the transpose eigenvector calculated using Eq. (17).

$$V = eig(con(y)) \tag{17}$$

where y is input image and y' is transpose of input image.

3.4. Classification

To efficiently detect the CMBs, three machine learning classifiers- SVM, QDA and ensemble classifiers are used. All classifiers are implemented in MATLAB.

1) *Quadratic Discriminant Analysis classifier*

QDA classifier is based on discriminant analysis classification technique. Based on Gaussian distributions, Quadratic Discriminant Analysis generates different data from different classes. First of all, QDA classifier computes sample mean of each class followed by computation of sample co-variance using Eq. (18).

$$\Sigma = \frac{\sum_{i=1}^{N}\sum_{k=1}^{K} M_{ik}(x_i - \mu_k)(x_i - \mu_k)^T}{N - K} \tag{18}$$

where M is an N by K class membership matrix and $\mu_k$is estimate of class mean calculated by Eq. (19).

$$\mu_k = \frac{\sum_{i=1}^{N} M_{ik} x_i}{\sum_{i=1}^{N} M_{ik}} \tag{19}$$

In the next step, QDA classifier predicts the classes using posterior probability, prior probability, and cost. It can be calculated using Eq. (20).

$$y = arg_{y=1....n} min \sum_{n=1}^{K} P(n|x)C(y|n) \tag{20}$$

where y is the predicted class, K represents a total number of classes and $P(n|x)$represents posterior probability of a class. While n represents the number of observations, x and $C(y|n)$ represents the cost of classifying observation y correctly.

2) *Support Vector Machine*

SVM is an extremely effective classifier, especially for binary classification problems. It works on the principle that by using an efficient kernel technique, high dimensional data can be separated into two classes using hyperplanes. In order to classify the given data, SVM first identifies support vectors v_i, weights w_i and bias b. Then it uses Eq. (21) to classify the data.

$$C = \sum_i w_i k(v_i,\ x\ ) + b \tag{21}$$

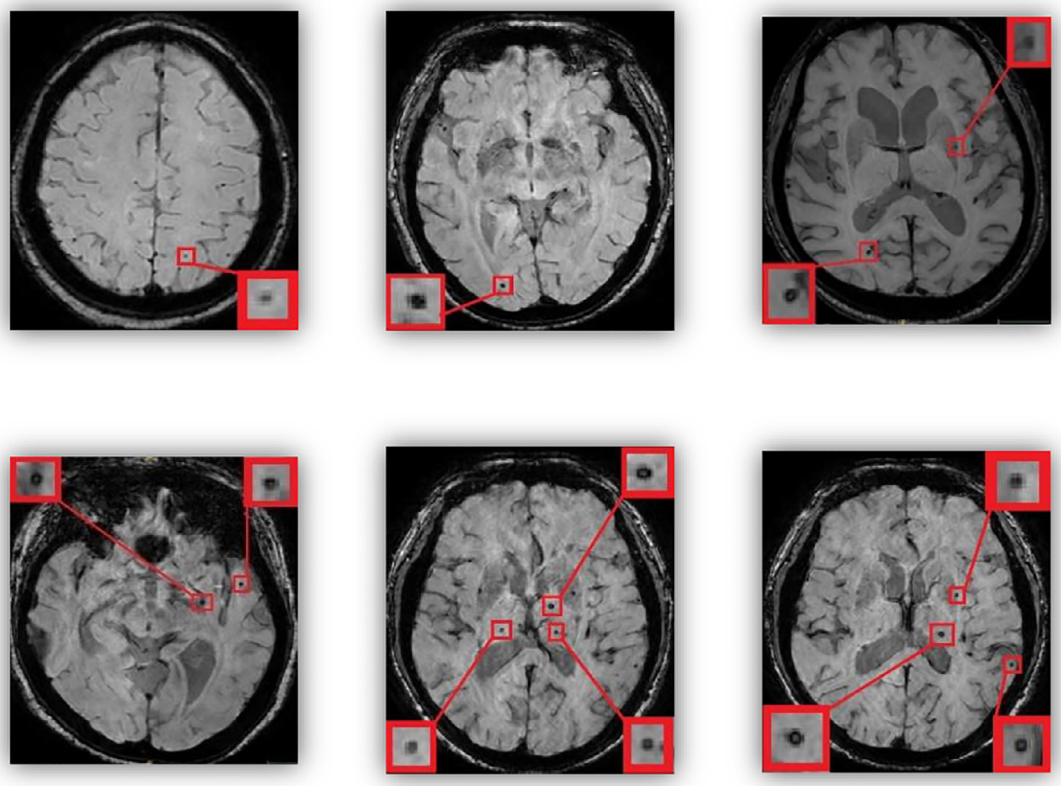

Fig. 2. Sample of images from the dataset. column with one, two and three CMBS per scan.

where k is the kernel function, v_i is the support vectors, w_i is the weight, $\underline{x}$ is input to classify and b is biased. We trained the classifier using Radial Basis Kernel function with a sigma value of 1.3, setting box constraint to 2 and auto selection of kernel scale.

3) *Ensemble classifier*

Ensemble classifiers use a technique that runs multiple classifiers instead of a single classifier. It makes the decision based on the majority votes of the classifiers. We have used RUSBOOST as an ensemble classifier. The boosting procedure used in RUSBOOST follows the procedure in AdaBoostM2 for constructing the ensemble. In our work, feature vector was given as input, predictors, and responses were chosen. The classifier was trained using three-fold cross-validation with a number of learners set to 70, learning rate to 0.1. RUSBOOST is mostly used in the classification of imbalanced data which means that some class in the training data has fewer samples as compared to another. We used RUSBOOST classifier to ensure that our system also produces efficient results in case of class imbalance problem.

4. Experiments and results

4.1. Dataset

In order to verify the effectiveness of the proposed technique, we used SWI dataset of 20 subjects for the detection of CMBs. This dataset is a part of the larger dataset used in [19]. Some example slices of the dataset with CMBs are shown in Fig. 2. The three images in Fig. 2 shows one, two and three CMBs respectively. The dataset is generated from a 3.0T Philips Medical System by setting repetition time of 17 ms, volume size of $512 \times 512 \times 150$, echo time of 24 ms, in-plane resolution of 0.45×0.45 mm, and slice thickness of 2 mm and slice spacing of 1 mm. The whole dataset is labeled by two experienced raters by following Microbleed Anatomical Rating scale [24]. In this work, we have worked on an axial view of 3D images. The dataset is divided into testing and training as shown in Table 2. The training set uses 104 images from 14 subjects while 63 images from 6 subjects have been used for testing of the algorithm.

Table 2
Overview of dataset.

Dataset	Training	Testing
Subjects	14	6
CMBs	104	63

Table 3
Details of features used to train SVM, QDA and ensemble classifiers extracted from binary and grayscale patches.

	Feature vector for SVM	Feature vector for QDA	Feature vector for ensemble
Features extracted from binary patches	Area	Area	Area
	Maximum box length	Maximum box length	Maximum box length
	Mean of Walsh transform	Orientation	Orientation
	Mean of Fourier transform	STD of Fourier transform	STD of Fourier transform
Features extracted from greyscale patches	Difference of Gaussians	Median of intensity values	Median of intensity values
	Variance	Variance	Variance
	Determinant of hessian	Determinant of hessian (g)	Determinant of hessian
	Eigen values (e1, e2)	Eigen values (e1)	Eigen values (e1)
	Trace of Hessian matrix	Trace of hessian	Trace of hessian
	Trace of hessian x hessian	Mean & std of Walsh transform	Mean & std of Walsh transform
	Determinant of hessian x hessian	Wavelet Transform	Wavelet Transform
	Mean & std of Radon transform	Std and mean of Fourier Transform	Std and mean of Fourier Transform
	Minimum intensity value	Minimum intensity value	Minimum intensity value
	Mean of intensity values	Mean of Image integral	Mean Image integral
	Mean and STD of Image integral	Mean of Hough Transform	Mean of Hough Transform
		Mean of Gabor Transform	Mean of Gabor Transform
		Mean of Karhunen Loeve Expansion	Mean of Karhunen Loeve Expansion

4.2. Experimentation

First of all, experiments were performed on training set which includes 14 subjects with CMBs. Thresholding is used for the slices in training set that had CMBs to get 104 CMB candidates from all 14 subjects. Then thresholds are used to extract ROIs of randomly chosen slices from the training set that did not have any CMB. After extracting ROIs, size-based filtering and the hole filling are applied to remove large objects from these ROIs based on the size of CMBs that resulted in 334 ROIs with no CMBs. Feature extraction is performed by combining 104 CMB candidates and 334 non-CMB candidates. These candidates are either binary patches or grayscale patches. For technical convenience, the features are divided into two categories; features extracted from binary patches and features extracted from grayscale patches. Different combinations of handpicked features were used to train SVM, QDA and ensemble classifiers. The details of features for all three classifiers used in this research are shown in Table 3.

1) *Features extracted from binary patches*

The first category includes features extracted from binary candidates. There are six features extracted from binary slices. First of all, we computed the area of the ROIs using Eq. (1) followed by computation of maximum box length using Eq. (3). The next feature is an orientation that specifies the angle between x-axis and major axis of the ellipse. Its value could be in the range −90 to 90. Then the Walsh transform was computed using Eq. (11) and the double mean of the resultant matrix was computed as a feature. At the end, Fourier transform was computed using Eq. (12), mean and standard deviation of resultant matrices were also used as features.

2) *Features extracted from grayscale patches*

The second category contains 23 features extracted from grayscale patches. First of all, we computed the difference of Gaussians by finding the difference of two Gaussian filters and convolving it with a grayscale patch using Eq. (4). Then three intensity-based features are computed that are minimum intensity value, mean of intensity values and median of intensity values. The variance is computed using Eq. (6) that specifies the difference of the elements in a vector. The next six features are local shape-based features on the Hessian matrix that is a square matrix shown in Eq. (8).

Features computed from the Hessian matrix are a determinant of Hessian matrix, the determinant of Hessian x hessian, trace of Hessian matrix and trace of Hessian x Hessian. Eigenvalues of Hessian were computed using Eqs. (9) and (10). The next feature extracted from grayscale patches is Walsh transform that was computed using Eq. (11). The standard deviation and mean of the resultant matrix were computed and also considered as features. Then the wavelet transform was extracted and only the approximation value was taken as a feature on which wavelet was applied multiple times till a single value is obtained. Then the radon transform of candidates was computed that specifies projections of an image matrix along specified directions. For this purpose, Theta was taken as 0 and 45°. Then the double mean and the standard deviation of the results were computed as features.

Table 4
Results of classification.

Results	Sensitivity	FPs/Sub	FPs/CMB
Ensemble classifier (for all 29 features)	93.7	75.5	7.2
QDA (for all 29 features)	92.1	84	8
SVM (for all 29 features)	64	25.5	2.4
Ensemble classifier (using features from Table 3)	92.1	42.5	4
QDA (using features from Table 3)	93.7	56	5.3
SVM (using features from Table 3)	90.5	65.8	6.3

The Fourier transform was computed for grey scale slice using Eq. (12). Then the image integral was calculated, the standard deviation and the double mean of resultant matrices were also considered as features. To compute the Hough transform, firstly, all the edges are detected using a canny detector and then the Hough transform was computed. Then the Gabor transform was computed using the values 4 and 90 for wavelength and orientation respectively. The resultant magnitude values are considered as features. The Karhunen Loeve Expansion was computed using Eq. (16) and the double mean of the result was considered as features.

For the testing set, we extracted a large number of candidates from 6 subjects and then the extracted candidates were filtered based on size and hole filling technique which resulted in a total of 12,593 ROIs including 63 CMBs and 12,530 normal candidates. Then the features in Table 1 were extracted from these candidates and the classifiers were trained on these features for classification.

4.3. Evaluation parameters

We measured the performance of our proposed technique in terms of following 3 parameters: sensitivity (S), number of FPs per subject (FP/Subs) and number of FPs per CMB (FP/CMB) using following Eqs. (22)–(24).

$$S = \frac{TPs}{TPs + FNs} \tag{22}$$

$$\frac{FPs}{Subs} = \frac{FPs}{N} \tag{23}$$

$$\frac{FPs}{CMB} = \frac{FPs}{TPs + FNs} \tag{24}$$

where TPs, FNs, and N represent the number of true positives, false negatives and a total number of testing subjects respectively.

4.4. Results

After extracting the ROIs from testing data and their corresponding features, the resultant feature vector was subjected to classification using SVM, QDA ensemble classifiers. In the first experiment, QDA classifier achieved a sensitivity of 93.7% with 56 false positives per patient and 5.3 false positives per CMB. Using ensemble classifier, a sensitivity of 92.1% was achieved with 42.5 false positives per patient and 4 false positives per CMB. The SVM achieved a sensitivity of 90.5%, 65.8 false positives per subject and 6.3 false positives per CMB. As QDA outperformed SVM and ensemble classifier in terms of sensitivity by a margin of 1.6% and 3.2% respectively. Therefore, we have presented the detailed results in Fig. 3 using QDA analysis. The red rectangles show true classifications and white rectangles represents false positives.

In the second experiment, all three classifiers; SVM, QDA and ensemble classifier are trained using all 29 features. In the third experiment, all these classifiers are also trained with handpicked selected features. QDA outperforms other two classifiers when handpicked features as shown in Table 4.

We have also presented the results in the form of confusion matrix for QDA because it outperforms SVM and ensemble classifier in Fig. 4. The results are also presented using ROC curves in Fig. 5 for QDA and ensemble classifier is better than SVM. The ROC shows that QDA performs best among these classifiers.

In the last experiment, we have evaluated the effect of feature vector length on sensitivity and computational time. In Fig. 6 shows the evaluation of sensitivity for SVM, QDA and ensemble classifiers by varying the feature vector length respectively. As each classifier uses a different number of features, therefore, the variation of sensitivity for each classifier is different. The results show that higher the number of features, the more sensitivity classifier achieve.

In Fig. 7, the effect of feature vector length is evaluated on computational time for SVM, QDA and ensemble classifiers. As each classifier uses a different number of features, therefore, the variation of computational time for each classifier is different. The computational time for all three classifiers is almost comparable.

A comparison of proposed technique with previous methods is presented in Table 5. To make the comparison dataset independent we have used a parameter of FPs/CMB [11]. Results showed that our algorithm outperformed the other methods in terms of sensitivity and produced comparable results in some cases. It should be noted that the results of these

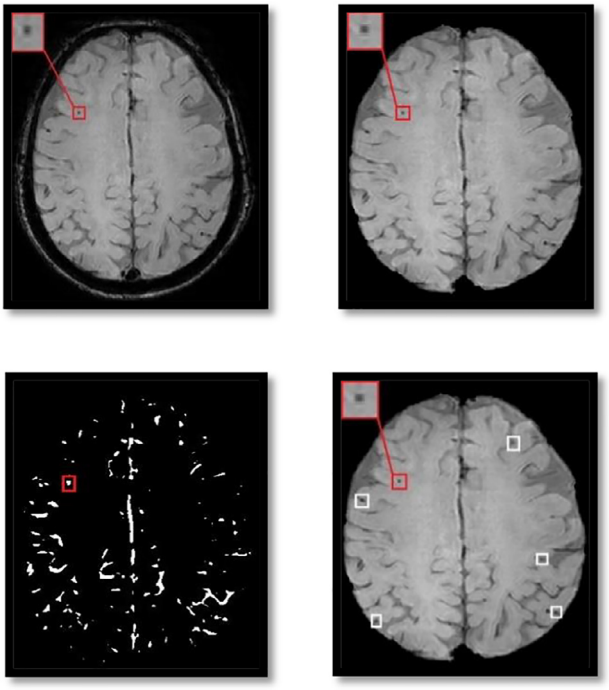

Fig. 3. From top to bottom, left to right, represents the input image, skull removed image, image with initial candidates and image with CMBS classification.

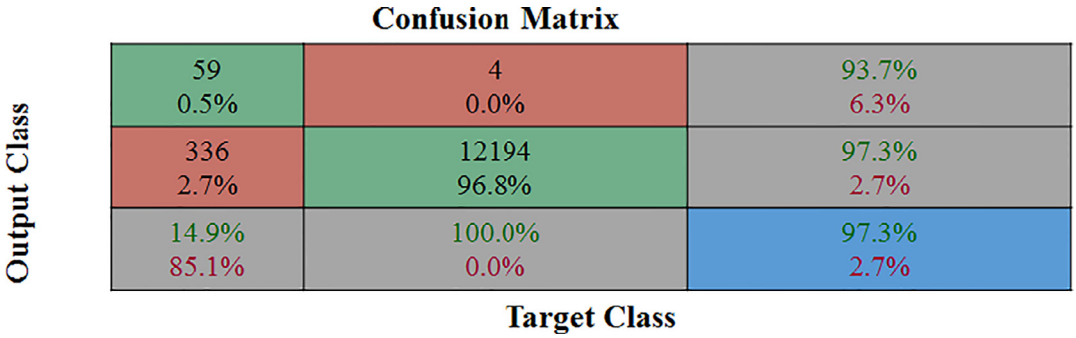

Fig. 4. Confusion matrix for (a) QDA.

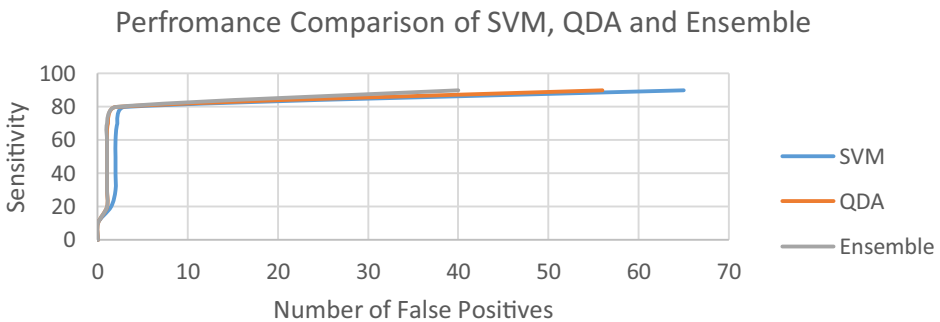

Fig. 5. The ROC curve for SVM, QDA and ensemble classifier to represent sensitivity and false positives.

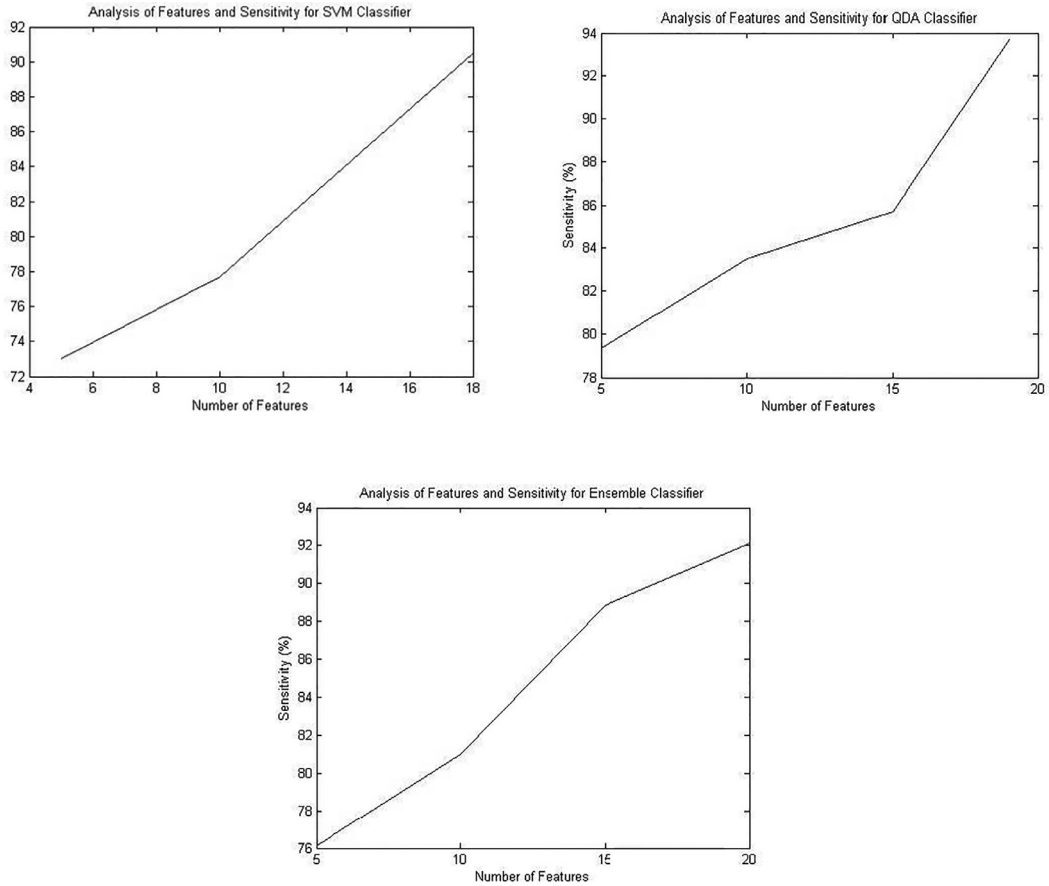

Fig. 6. Top to bottom, left to right shows the graphs that represents the effect of feature vector length on sensitivity for SVM, QDA and ensemble classifier respectively.

Table 5
Comparison with previous techniques.

Methods	Sensitivity	FPs/Sub	FPs/CMB
Saghier et al. [15]	50	–	–
Barnes et al. [13]	81.7	107.5	5.4
Ghafaryasl et al. [12]	90.9	4.1	1.8
Bian et al. [11]	86.5	44.9	1.5
Fazlollahi et al. [9]	92	16.8	6.7
Dou et al. [16]	89.4	7.7	0.9
Chen et al. [18]	89.1	6.4	–
Dou et al. [19]	93.16	2.74	–
Heuvel et al. [17]	89	25.9	0.29
Proposed method (with ensemble classifier)	92.1	42.5	4
Proposed method (with QDA classifier)	93.7	56	5.3
Proposed method (with SVM classifier)	90.5	65.8	6.3

different techniques are dataset dependent. The comparison shows that our proposed system achieves high sensitivity but the false positive rate per subject is considerably high for QDA. Many existing systems decreases these false positives by a manual review in post-processing. Our proposed algorithm involves no manual review in post-processing step which makes it difficult to reduce false positives. Considering these limitations, our proposed give even reasonably good performance in terms of false positives.

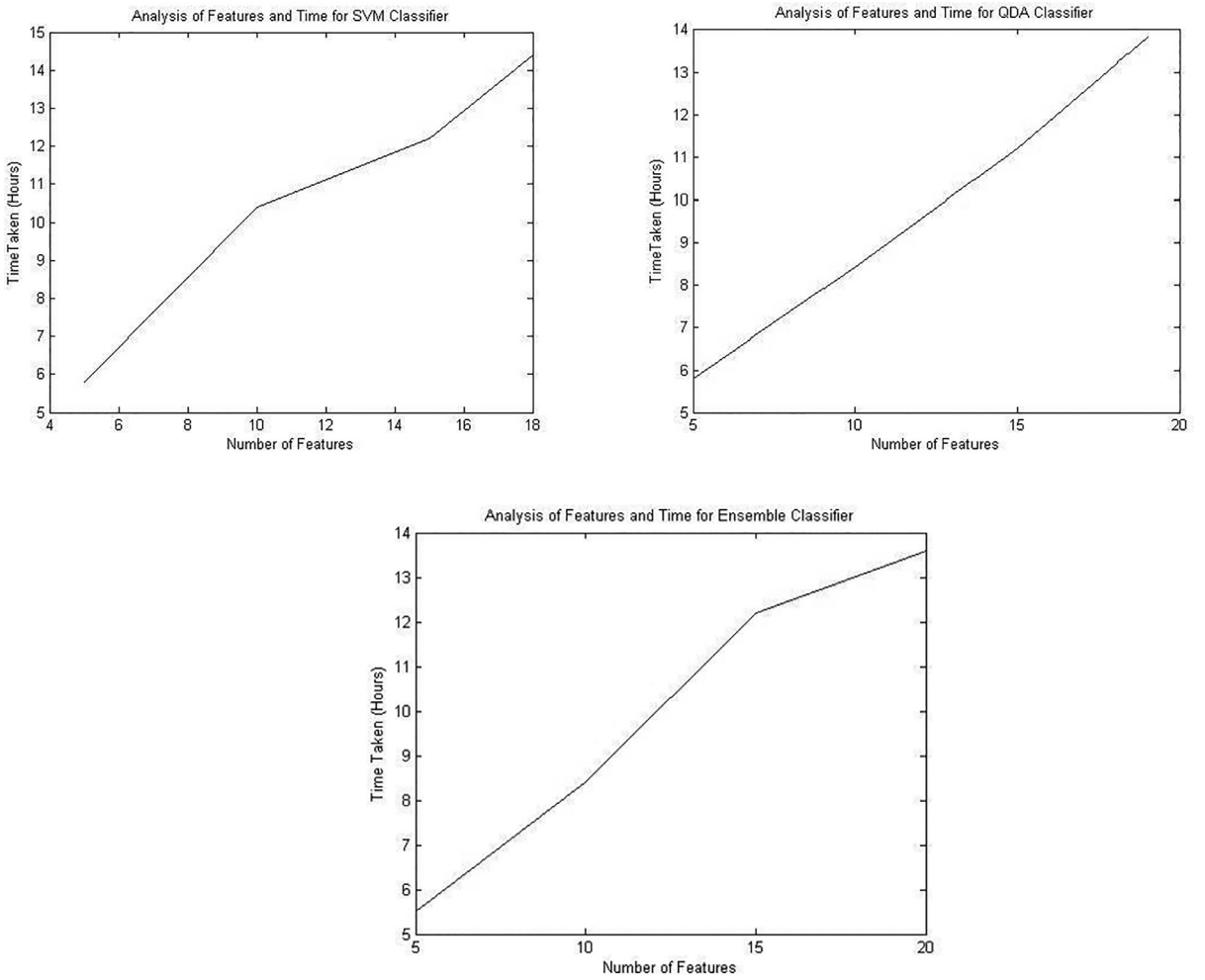

Fig. 7. Top to bottom, left to right shows the graphs that represents the effect of feature vector length on computational time for SVM, QDA and ensemble classifier respectively.

5. Conclusion

A computer-aided system for classifying the region of interest can be extremely beneficial for the analysis of CMBs. However, these systems face some limitations in terms of sensitivity, and false positives. Manual removal of CMBs is a tedious and time consuming task. Therefore, CMBs detection and classification from SWI scans remains a challenging task. The main objective of this study is to design an efficient and fully automated computer-assisted system for the detection of CMBs in SWI scans with high sensitivity. In this context, three different machine learning classifiers namely 1) SVM, 2) QDA, and 3) ensemble classifier, are used to detect and classify CMBs. The proposed framework utilizes number of diverse handpicked features from binary and grayscale patches to improve the sensitivity and reduce the false positive rate. The best sensitivity of 93.7% is achieved using QDA classifier in comparison with SVM (90.5%) and ensemble classifier (92.1%) while keeping FP considerably low. The ensemble classifier offers slightly better performance in terms of FPs/Subject and FPs/CMB in comparison to other two classifiers. The ensemble classifier has reduced FPs/Subjects to 42.5 in comparison to QDA (56 FPs/Subject) and SVM (65.8% FPs/Subject). A slight decrease in FPs/CMB is also observed using ensemble classifier but QDA offers overall better performance in terms of sensitivity and FPs. In addition, the proposed automated approach requires no human involvement in post-processing step for removal of CMBs mimics. As for the future directions of this work, CMBs detection can be further investigated by using more advanced machine learning classification algorithms like neural networks and deep learning. Moreover, the proposed system can be enhanced by developing a more sophisticated and automated feature selection technique. This will enable us to automatically select more descriptive features that can further improve the performance of the system.

Acknowledgments

This research was supported by the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2017-2016-0-00312) supervised by the IITP (Institute for Information & communications Technology Promotion).

References

[1] Greenberg SM, Vernooij MW, Cordonnier C, Viswanathan A, Al-Shahi Salman R, Warach S, et al. Cerebral microbleeds: a guide to detection and interpretation. Lancet Neurol 2009;8:165–74.
[2] Charidimou A, Krishnan A, Werring DJ, Jäger HR. Cerebral microbleeds: a guide to detection and clinical relevance in different disease settings. Neuroradiology 2013;55:655–74.
[3] Fan YH, Zhang L, Lam WW, Mok VC, Wong KS. Cerebral microbleeds as a risk factor for subsequent intracerebral hemorrhages among patients with acute ischemic stroke. Stroke 2003;34:2459–62.
[4] Imaizumi T, Horita Y, Hashimoto Y, Niwa J. Dotlike hemosiderin spots on T2*-weighted magnetic resonance imaging as a predictor of stroke recurrence: a prospective study. J Neurosurg 2004;101:915–20.
[5] Zhang Y-D, Zhang Y, Hou X-X, Chen H, Wang S-H. Seven-layer deep neural network based on sparse autoencoder for voxelwise detection of cerebral microbleed. Multimedia Tools Appl 2017:1–18.
[6] Zhang Y-D, Hou X-X, Chen Y, Chen H, Yang M, Yang J, Wang SH. Voxelwise detection of cerebral microbleed in CADASIL patients by leaky rectified linear unit and early stopping. Multimedia Tools Appl 2017:1–21.
[7] Goos JD, van der Flier WM, Knol DL, Pouwels PJ, Scheltens P, Barkhof F, Wattjes MP. Clinical relevance of improved microbleed detection by susceptibility-weighted magnetic resonance imaging. Stroke 2011;42:1894–900.
[8] Kuijf HJ, de Bresser J, Biessels GJ, Viergever MA, Vincken KL. Detecting cerebral microbleeds in 7.0 t mr images using the radial symmetry transform. In: 2011 IEEE international symposium on biomedical imaging: from nano to macro; 2011. p. 758–61.
[9] Fazlollahi A, Meriaudeau F, Villemagne VL, Rowe CC, Yates P, Salvado O, Bourgeat P. Efficient machine learning framework for computer-aided detection of cerebral microbleeds using the Radon transform. In: 2014 IEEE 11th international symposium on biomedical imaging (ISBI); 2014. p. 113–16.
[10] Kuijf HJ, de Bresser J, Geerlings MI, Conijn MM, Viergever MA, Biessels GJ, Vincken KL. Efficient detection of cerebral microbleeds on 7.0 T MR images using the radial symmetry transform. NeuroImage 2012;59:2266–73.
[11] Bian W, Hess CP, Chang SM, Nelson SJ, Lupo JM. Computer-aided detection of radiation-induced cerebral microbleeds on susceptibility-weighted MR images. *NeuroImage* 2013;2:282–90.
[12] Ghafaryasl B, van der Lijn F, Poels M, Vrooman H, Ikram MA, Niessen WJ, de Bruijne M. A computer aided detection system for cerebral microbleeds in brain MRI. In: 2012 9th IEEE international symposium on biomedical imaging (ISBI); 2012. p. 138–41.
[13] Barnes SR, Haacke EM, Ayaz M, Boikov AS, Kirsch W, Kido D. Semiautomated detection of cerebral microbleeds in magnetic resonance images. Magn Reson Imaging 2011;29:844–52.
[14] van den Heuvel T, Ghafoorian M, van der Eerden A, Goraj B, Andriessen T, ter Haar Romeny B, et al. Computer aided detection of brain micro-bleeds in traumatic brain injury SPIE Medical Imaging; 2015. 94142F-94142F-7.
[15] Seghier ML, Kolanko MA, Leff AP, Jäger HR, Gregoire SM, Werring DJ. Microbleed detection using automated segmentation (MIDAS): a new method applicable to standard clinical MR images. PloS One 2011;6:e17547.
[16] Dou Q, Chen H, Yu L, Shi L, Wang D, Mok VC, Heng PA. Automatic cerebral microbleeds detection from MR images via independent subspace analysis based hierarchical features. In: 2015 37th annual international conference of the IEEE engineering in medicine and biology society (EMBC); 2015. p. 7933–6.
[17] van den Heuvel T, van der Eerden A, Manniesing R, Ghafoorian M, Tan T, Andriessen T, Platel B. Automated detection of cerebral microbleeds in patients with traumatic brain injury. *NeuroImage* 2016;12:241–51.
[18] Chen H, Yu L, Dou Q, Shi L, Mok VC, Heng PA. Automatic detection of cerebral microbleeds via deep learning based 3d feature representation. In: 2015 IEEE 12th international symposium on biomedical imaging (ISBI); 2015. p. 764–7.
[19] Wang S, Li P, Chen P, Phillips P, Liu G, Du S, Dou Q, Chen H, Yu L, Zhao L, Qin J, Wang D, Heng PA. Automatic detection of cerebral microbleeds from MR images via 3D convolutional neural networks. IEEE Trans Med Imaging 2016;35:1182–95.
[20] Shattuck DW, Leahy RM. BrainSuite: an automated cortical surface identification tool. Med Image Anal 2002;6:129–42.
[21] Viola P, Jones M. Rapid object detection using a boosted cascade of simple features. In: Computer vision and pattern recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE computer society conference on; 2001 I-I.
[22] Beauchamp K. Applications of walsh and related functions. With an introduction to sequence theory: Academic Press, Inc., Microelectronics and Signal Processing Series, 24–28 Oval Road, London NW1 7DX and Orlando, FL 32887, 1984, xvi+ 308 pp., ISBN 0-12-084180-0. Signal Process 1985;9:135–6.
[23] Wang S, Li P, Chen P, Phillips P, Liu G, Du S, Zhang Y. Pathological brain detection via wavelet packet tsallis entropy and real-coded biogeography-based optimization. Fundam Inform 2017;151:275–91.
[24] Gregoire S, Chaudhary U, Brown M, Yousry T, Kallis C, Jäger H, Werring DJ. The Microbleed Anatomical Rating Scale (MARS) reliability of a tool to map brain microbleeds. Neurology 2009;73:1759–66.

Tayyab Ateeq is an M.S. scholar at University of Engineering and Technology (UET) Taxila, Pakistan. He has done his B.S. in Computer Engineering from UET Taxila. His research interests are image processing, signal processing, and machine learning.

Muhammad Nadeem Majeed has done his Ph.D. in Computer Engineering from UET Taxila in 2015. His research interests are computer networks, image processing, and recommender systems. He is currently serving as an Assistant Professor at Department of Software Engineering at UET Taxila.

Syed Muhammad Anwar has done his Ph.D. from the University of Sheffield. His research interests include signal processing, medical imaging, and machine learning. He is currently serving as an Assistant Professor at Department of Software Engineering at UET Taxila.

Muazzam Maqsood has done his Ph.D. in Software Engineering from UET Taxila. His research interest includes machine learning, recommender systems, and image processing. He is currently serving as an Assistant Professor in the department of computer science at COMSATS Institute of Information Technology, Pakistan.

Zahoor-ur-Rehman has experience both in academia and research. He has received his educational and academic training at the University of Peshawar, Foundation University Islamabad and UET Lahore, Pakistan. He joined COMSATS Institute of information technology as an assistant professor in early 2015. Along with teaching responsibilities, he is an active researcher and reviewers of various conferences and reputed journals.

Jong Weon Lee received M.S. degree in Electrical and Computer Engineering from University of Wisconsin at Madison in 1991, and a Ph.D. degree from the University of Southern California in 2002. He is presently Professor of Department of Software at Sejong University. His research interests include augmented reality, human-computer interaction and serious game.

Khan Muhammad is a Ph.D. Fellow at Intelligent Media Lab, Department of Digital Contents, Sejong University, Republic of Korea. His research interests include information security, steganography, computer vision, wireless capsule endoscopy, wireless sensor networks, video summarization, CCTV video analysis, big data, and deep learning.

Shui-hua Wang received a B.S. from Southeast University in 2008 and an M.S. from The City University of New York in 2012. She received her Ph.D. from Nanjing University in 2017. At present, she works as an assistant professor at Nanjing Normal University.

Sung Wook Baik is Professor at Sejong University. He is conducting research to provide natural and convenient communication between people and computers through the use of innovative and intelligent technologies in media. Major areas of research include data mining/visual mining, computer vision/intelligent robots, hybrid reality, restoration of cultural properties/tourism contents, and digital contents authoring.

Irfan Mehmood is an Assistant Professor at Sejong University. He conducts research in a number of basic and applied areas such as image and scene segmentation, motion and video analysis, perceptual grouping, shape analysis and object recognition. His research methods emphasize the use of segmented (part- based) symbolic descriptions of objects.