Digital Audio Resampling Detection Based on Sparse Representation Classifier and Periodicity of Second Derivative

Jing XU[1], Jeffrey XIA[2]
[1] College of Computer Science & Technology, Guizhou University
Guiyang, Guizhou, 550025, China
[2] Ole Wolff Electroacoustic & Magnetic Solutions Company
1525 McCarthy Road, Suite 1093, Milpitas CA, 95035, U. S. A.
xujcan@163.com

JDIM
Journal of Digital Information Management

ABSTRACT: *Digital audio detection is a forensics authenticity request. However, the classic methods for digital audio detection are no longer effective, especially given the use of compound tampering, such as adding background noise, and the size of processed data sets easily exceeding gigabytes. This article aims to discuss a detection method that uses a sparse representation classifier based on adaptive least squares (recursive least squares sparse representation classification [RLS-SRC]) and periodicity in the second derivative of an audio signal as a classification feature for digital media forensics. Using adaptive least squares, the proposed RLS-SRC can perform online updates and thus reduce the burden of training. In cases with background noise, our proposed method yields better classification compared with the method based on K-singular value decomposition (K-SVD).*

Categories and Subject Descriptors
H.5.5 [Sound and Music Computing]: Signal analysis, synthesis, and processing **I.5 [Pattern Recognition]**: Signal processing

General Terms: Collect, Analysis, Information, Experimentation

Keywords: Resampling Detection; Sparse Representation Classifier, Periodicity of Second Derivative; Dictionary Learning; RLS-SRC

Received: 12 January 2015, Revised 18 February 2015, Accepted 24 February 2015

1. Introduction

With the development of digital information, digital audio and multimedia have gradually become a part of our daily lives. Digital audio in particular is prone to possible distortions and artifacts. When digital audio is submitted to the court as judicial evidence, verifying its authenticity becomes a top priority. Such conditions drive the research and development of relevant technology for testing multimedia authenticity [1, 2, 3]. Currently available methods for detecting digital voice tampering are divided into two types. The first is an active detection technology that mainly uses watermarking technology to ensure the primitiveness of audio files; however, the technology does not delete certain sound bites, making it an unpopular choice [3]. The second is passive detection technology, also known as digital audio blind detection technology; this method detects audio authenticity characteristics by extracting audio objects.

Digital audio blind detection technology is based on digital audio signals and the characteristics of such signals; it does not require an embedded watermark, and it addresses the needs inherent in the detection of digital audio tampering. In reality, a person can copy the speaking content of a digital audio file and paste it to another file. Hence, the detection of tampering patterns in multimedia has recently attracted significant attention. In the copy–paste process, the audio sampling rate is sometimes changed. Such operation often requires the use of an interpolation technique for resampling shown in Figure 1. Checking an object for resampling is an important means of digital audio tamper detection. Zuo [4] used the

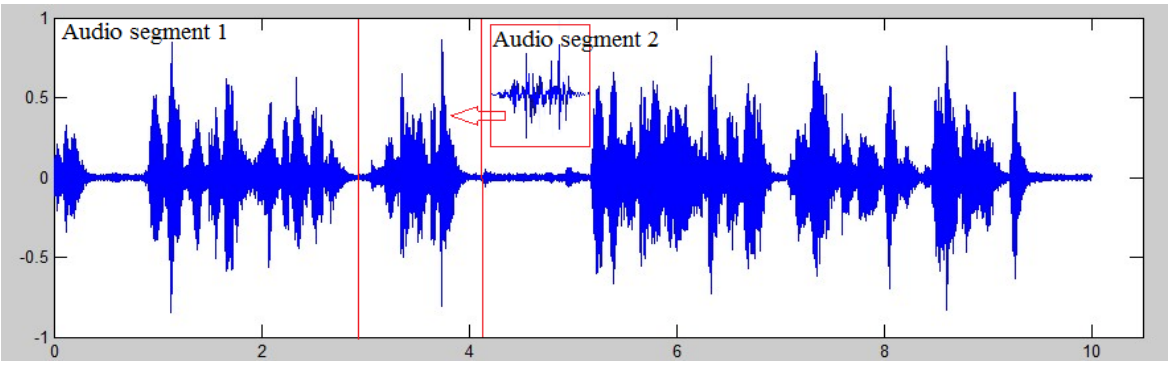


Figure 1. The Copy-Paste Processing of Audio

expectation– maximization algorithm to detect the periodic interpolation features of a resampling image; although good results were achieved, the algorithm requires a considerable amount of computation time for iterative convergence.

After the extraction of digital audio resampling features, resampling detection is performed to identify the presence of a different sampling rate conversion. Such process can be regarded as a multiclass classification. Traditional methods such as SVM [5] do not address these pattern recognition problems in multimedia field. To close this gap, we propose the use of sparse representation classifiers (SRCs).

In recent years, the use of SRCs has gained increasing attention in the area of signal processing. Sparse representation refers to the use of a linear combination of an over-complete dictionary to approximate a given original signal such that it satisfies sparse constraints. The dictionary is obtained by training samples and can effectively reflect the characteristics of an object signal; thus, such signal can be sparsely represented by an over-complete dictionary with sparse weight and can be used as an effective dimension reduction method and a classifier [6, 7, 8]. The classical dictionary learning method K-SVD [9] has been applied in face recognition [6] and image classification fields [7] as a classifier and is known to achieve good classification effects, from handwritten digit classification, digital art identification to nonlinear inverse image problems. However, K-SVD involves singular value decomposition and matrix inversion, which cannot be applied to the digital signals of large sample data.

To overcome the limitation of the traditional SRC, we introduce an online SRC method to the solution for large-scale classification for digital signal processing. The proposed method can use test sample update classifiers, namely, recursive least squares sparse representation classification (RLS-SRC), and realize the purpose of online learning. The earliest online learning method was suggested by Oja [10] in the study of the first principal component online update. Honeine [11] further expanded the online kernel PCA algorithm to multiple principal components and applied it to images of handwritten digits. In the current work, we propose the use of the recursive least square, which was first proposed by Skretting [12], as an online dictionary learning method. Through the iterative formula update feature, the proposed method does not need to solve the inverse matrix and singular value decomposition, thereby offering an online classification solution.

The rest of this manuscript is organized as follows. First, the audio resampling detection principle and resampling feature extraction are introduced in Section 2. The generation of the RLS-SRC is discussed in Section 3. The application of the online SRC method in audio tampering detection is presented in Section 4, where the sensitivity of the experiment is also evaluated.

2. Resampling Feature Extraction

2.1 Digital Audio Resampling Detection Principle

The digital audio resampling process includes three steps: zero padding, interpolation, and sampling. This process is made distinct by its difference interpolation filters. Using a linear or quadratic interpolation function can make a given signal appear periodic [13]. Figure 2 illustrates the spectrum without resampling (a) and with two (b) and four (c) instances of resampling digital audio signal's second derivative. The original sampling characteristics of the spectrum are smooth, and the spectrum peak of the resampling audio signal appears more than once.

2.2 Feature Extraction

Set $y(n)$ is the original digital audio signal; h is the interpolation function; the interpolated samples $i(p_0)$, where n is a real number, therefore $i(p_0)$ can be written as a matrix product

$$i(p_0) = y^T h . \qquad (1)$$

where matrix y has entries $y_k = y(n_0 - k)$, and n_0 is the lowband integer of p_0. The interpolation matrix h has entries $h_k = h(k - \delta)$, where $\delta = p_0 - n_0$. These samples which precede and follow $i(p_0)$ are given as $i(p_0 - \Delta)$ and $i(p_0 + \Delta)$. The second derivative $s(p_0)$ is calculated as

$$s(p_0) = 2^* i(p_0) - i(p_0 - \Delta) - i(p_0 - \Delta) \qquad (2)$$

Y_i denotes the voice clips, in which $Y_i = [y_{i1}, y_{i2}, ..., y_{in}]'$; and n is the frame length. (2) written in matrix form, it is $s(p_0) = y^T c(\delta)$, where the entries of matric c are

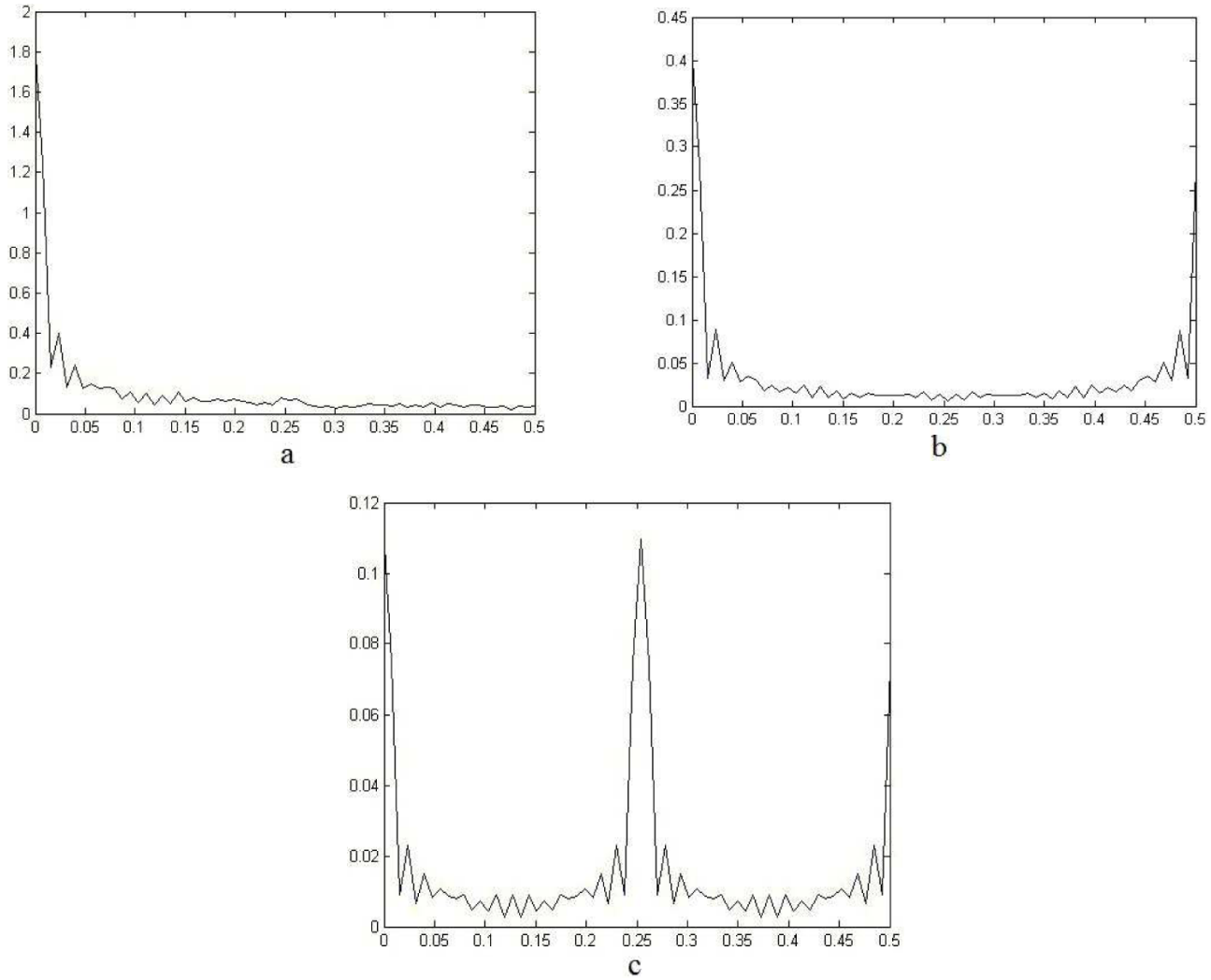


Figure 2. The spectrum of resampling digital audio signal's second derivative

$$c_k(\delta) = 2^* h(k-\delta) - h(k-\delta-\Delta) - h(k-\delta-\Delta). \quad (3)$$

Equation (2) is used to calculate the second derivative of the audio signal, which is given by $S_i=[s_{i1}, s_{i2}, ..., s_{in}]'$. the variance $v(p_0)$ of signal $s(p_0)$ is defined as

$$v(p_0) = c(\delta)^T K c(\delta) \quad (4)$$

where K is the covariance matrix of original signal $y(n)$. By transforming Equation (4) to obtain the frequency spectrum J_{ij}, the formula is defined as

$$J_{ij} = |F(v(p_0))|, \quad (5)$$

where F is the Fourier transform.

The spectrum amplitude is found to be small in the silent audio time. In this case, the periodic characteristic is not obvious. In addition, the voice energy of the spectrum concentrated in low frequency jams the periodic testing of the variance of the second-order differential.

The low frequency of the spectrum is a phonetic element because of the characteristics of the audio signal. To reduce interference while retaining the resampling characteristics, the spectrum J_{ij}, is passed through a high-pass filter, that is,

$$X_{ij} = J_{ij} * H,$$

which X_{ij} denotes the resample characteristics of the audio signal; H here is a function of high-pass filtering.

3. Sparse Classifier Design

3.1 Sparse Signal Representation

Sparse signal representation can be achieved by shearing the coefficients of an over-complete dictionary in the area of digital signal processing. The key problem in employing this dictionary in sparse representation is obtaining an over-complete dictionary D, from which atoms can be obtained and linearly combined to approximate or represent an original digital signal.

An input signal $X \in R^{N*L}$ may be represented as a sequence of column vector

$$\{X \in R^{N*L}\}_{i=1}^{L}.$$

Meanwhile, a dictionary D is composed of a set of atoms

$$\{d_i \in R^N\}_{i=1}^K, \text{ with } K > N.$$

Thus, $D \in R^{N*K}$. Let matrix $W \in R^{K*L}$ be a collection of column weight vectors with length K so that a sparse signal representation is expressed as

$$\tilde{x}_i \in Dw_j = \sum_{i=1}^{K} d_i w_j(i), \quad (6)$$

where w_j is a column vector that consists of the coefficients for representation of signal x_j. The reconstructed signal matrix may be expressed as

$$\tilde{X} = DW.$$

The representation error for x_j is given by $r_j = x_j - \tilde{x}_j = x_j - Dw_j$, and the total reconstructed error becomes

$$R = X - \tilde{X} = X - DW. \quad (7)$$

Dictionary learning is usually formed into the following optimization problem:

$$\arg\min_{D,W} \|R\|_F^2 = \arg\min_{D,W} \sum_{j=1}^{L} \|r_i\|_F^2 = \arg\min_{D,W} \sum_{j=1}^{L} \|x_j - Dw_j\|_F^2, \quad (8)$$

where the function $||.||_F$ denotes the Frobenius norm for matrices. Generally, sparse representation is obtained by setting the coefficient number to a small numeral s or by limiting the error to ζ, which is a positive number.

A practical optimization strategy is to split the dictionary learning problem into two stages, namely, sparse coding and dictionary update, both of which are generally solved iteratively as follows:

(1)For a given signal X, initiate the dictionary D.

(2)For the *j-th* column of X, keep D fixed, and find weight vector w_j so that this column is sparsely represented.

(3)Keep w_j fixed, and update D to minimize the total error. Detailed descriptions of the last two steps are provided in the following subsection.

3.2 Order recursive matching pursuit (ORMP)

In the second step described above, when D is fixed, the coefficients are found by sparse constraint or error limitation. The formula is given by

$$w_{j\,opt} = argmin \|x_j - Dw_j\|_2$$
$$s.t \|w_j\|_0 \le s \; or \; \|x_j - Dw_j\|_2 \le \zeta \quad (9)$$

where the pseudo-norm $\|.\|_0$ defines the nonzero element number of a vector and ζ is a threshold for controlling the representation error. The above equation is clearly an NP-hard problem. The matching pursuit algorithm and its variants can serve as a suboptimal, but not necessarily optimal technique. The basic matching pursuit (BMP), which refers to the selection of vectors one at a time, is a sequential greedy method. At first, the algorithm chooses a dictionary vector d_{i0} that best matches the signal vector so that

$$x_j = (x_j^T d_{i_0})\, d_{i_0} + r_1, \quad (10)$$

$$||x_j||_2^2 = |x_j^T d_{i_0}|^2 + ||r_1||_2^2 \quad (11)$$

The algorithm projects the signal vector x_j on every column vector d_j of dictionary D and computes the residual r_1. To minimize r_1, the selected atom d_{i_0} must ensure that

$$|x_j^T d_{i_0}| \ge sup|x_j^T d_i|, i = 1...K \quad (12)$$

An approximation can be built by iteratively selecting a new vector from the last dictionary vector that best matches the residual r_k. The approximation after iteration s can be written as

$$x_j = (x_j^T d_{i_0})\, d_{i_0} + r_j^T d_{i_1} + ... (r_{j_{s-1}}^T d_{i_{s-1}}) + r_s \quad (13)$$

From Equations (6) and (13), we can observe that the coefficients used in the approximation of the signal vector x_j is the inner product between the residual at that iteration stage and the chosen dictionary atoms

$$w_j(i_k) = \begin{cases} r_{j_k}^T d_{i_k} & k = 0, 1, ..., s, \\ 0 & other \end{cases} \quad (14)$$

where $r_{j0} = x_j$.

The matching pursuit algorithm ORMP [14] is different from BMP; the atom selected from each iteration orthogonalizes the remaining vectors in the dictionary. After the last iteration, the coefficients are recalculated using the least squares method. In our work, we use ORMP because it can yield better results compared with OMP [15] and BMP and because its complexity is similar to an effective QR implementation.

3.3 Recursive Least Squares for Dictionary Learning

Karl Skretting [11] proposed the recursive least squares-based dictionary learning algorithm (RLS-DLA), which can be used to learn over-complete dictionaries for sparse signal representation. The algorithm can be considered as a generalization of the continuous k-means algorithm; it follows the same lines as the derivation of RLS for adaptive filters, hence the name RLS-DLA. In the second step described previously, when W is fixed, the solution proposed by RLS-DLA for Equation (7) is

$$D = (XW^T)(WW^T)^{-1} = B_L C_L, \quad (15)$$

where B_L and C_L are defined as

$$B_L = (XW^T) = \sum_{j=1}^{L} x_j w_j^T; \quad (16)$$

$$C_L^{-1} = (WW^T) = \sum_{j=1}^{L} w_j w_j^T. \quad (17)$$

Given RLS-DLA with a forgetting factor λ, let the minimization problem be a weighted sum of the least square errors

$$f(D) = \sum_{j=1}^{L} \lambda^{L-j} \| r_j \|_2^2. \quad (18)$$

When a new signal x_{L+1} becomes available, the dictionary D is updated by

$$D_{L+1} = \arg\min_D f(D) = \arg\min_D \sum_{j=1}^{L+1} \lambda^{L+1-j} \|r_j\|_2^2$$

$$= \arg\min_D (\lambda \|\hat{X}_j - D_L \hat{W}_L\|_F^2 + \|x_{L+1} - D_L W_{L+1}\|_2^2) \quad (19)$$

where matrices X and W are recursively defined as

$$\hat{X}_j = [\sqrt{\lambda}\hat{X}_{j-1}, x_j],\ \hat{X}_1 = x_1\ , \quad (20)$$

$$\hat{W}_j = [\sqrt{\lambda}\hat{W}_{j-1}, w_j],\ \hat{W}_1 = w_1. \quad (21)$$

Therefore, the new dictionary $D_{L+1} = B_{L+1}C_{L+1}$ can be defined, with

$$B_{L+1} = \hat{X}_{L+1}\hat{W}_{L+1}^T = [\sqrt{\lambda}\hat{X}_L, x_{L+1}][\sqrt{\lambda}\hat{W}_L, x_{L+1}]^T$$
$$= \lambda\hat{X}_L\hat{W}_L^T + x_{L+1}w_{L+1}^T = \lambda B_L + x_{L+1}w_{L+1}^T \quad (22)$$

$$C_{L+1}^{-1} = \hat{W}_{L+1}\hat{W}_{L+1}^T = [\sqrt{\lambda}\hat{W}_L, w][\sqrt{\lambda}\hat{W}_L, x]^T$$
$$= \lambda\hat{W}_L\hat{W}_L^T + w_{L+1}w_{L+1}^T = \lambda C_L^{-1} + w_{L+1}w_{L+1}^T \quad (23)$$

In RLS-SRC, a codeword of the dictionary d_i is used to represent a resampling category that corresponds to a column of the dictionary D. In the stage of learning classifiers, a matching pursuit algorithm, such as ORMP, is used to calculate the weight of the sample in the codeword. The largest weight value of the codeword is the sample's classification. Then, RLS-DLA is utilized to update the selected codeword. In the stage of resampling detection, the ORMP algorithm is used to select a codeword which involves a nonzero element, which is closest to that in the testing sample as the resampling category.

3.4 Algorithm Implementation

To enhance the periodicity of silent frame's second derivative and reduce the interference of speech, we propose five audio resampling detection steps: (1) use of the Voice Active Detect (VAD) algorithm to divide a given audio signal into voice clips and silent frames; (2) calculation and transformation of the second-order difference into spectrum; (3) filtration of the audio frequency components through the high-pass filter and preservation of the frequency components of heavy resampling period; (4) use of the dictionary learning method to train resampling sparse classifiers; and (5) according to the test sample belonged to which codeword of the dictionary, calculated by ORMP, classification of the test sample.

Set $\{f_{ij}\}$ is the training sample of codeword d_i, h is the number of samples, and w_j is the weight calculated by ORMP. The RLS-SRC algorithm is described as follows:

(1) Training Classifiers Stage

a) Randomly select resampling feature vectors of training samples from initial dictionary D; initialize identity matrix C, and forget factor λ;

b) For an input sample f_{ij}, get $w_j = f_{ij}^T d_i$ by ORMP;

c) Calculate error $r = f_{ij} - Dw_j$; update matrix $C_j^* = \lambda^{-1} C_{j-1}$;

d) Update and normalize dictionary D_j;

e) If the input sample is unfinished, repeat Step b.

(2) Detecting resampling stage:

a) Set nonzero number of weight vector $s = 1$, and divide test sample into frames;

b) Choose codeword, in which the speech frame's weight calculated by ORMP is nonzero; use the codeword as a resampling class of the frame;

c) View statistics of each resampling class for the testing sample; the resampling class with the largest number is the detected result.

The principle of resampling detection is shown in Figure 3.

4. Experimental Results

We select 100 samples of six audio sampling rates (8, 11.025, 16, 22.05, 32, and 44.1 KHz), which are 5 min long. Resampled with cubic Lagrangian interpolation functions, the training samples are transformed into 15 combinations of sampling rates. However, some transitions are similar, such as the transformations of 8 KHz to 16 KHz and 16 KHz to 32 KHz. Thus, the number of resampling classes is narrowed down to eight, as shown in Table 1.

The amplitude variation of silence is small in terms of waveform; thus, it only detects segments of speech. Audio samples are divided by the VAD algorithm into silent and speech segments. Then, the speech segments are framed, with the length of the segments being n = 4096. The size of the dictionary D is 64 × 8, and the cutoff frequency of the high-pass filter fs is 300 Hz, with forgetting factor ë= 0.9. The feature vectors of the resampling classes are shown in Figure 4.

In reality, the digital audio may be tampered by other operations, such as the addition of background noise, after resampling and joining. These operations may disturb resampling detection. To test the validity of the algorithm proposed in this work, we add 10 dB to 60 dB of random noise to the test sample after resampling.

We then compare the RLS-SRC-based resampling detection with the SRC based on K-SVD, which is often referred to as the dictionary learning algorithm for SRCs [6]. The experimental results (Figure 5) indicate that the RLS-SRC-based resampling detection achieves higher accuracy than the SRC based on K-SVD. When the random noise is less than 30 dB, RLS-SRC maintains its relatively stable detection ability. The results of the experiment prove that the RLS-SRC algorithm has high

	8 KHz	11.025 KHz	16 KHz	22.05 KHz	32 KHz	44.1 KHz
8 KHz	Class 1	Class 4	Class 2	Class 5	Class 3	Class 6
11.025KHz		Class 1	Class 7	Class 2	Class 8	Class 3
16 KHz		Class 1	Class 4	Class 2	Class 5	
22.05 KHz				Class 1	Class 7	Class 2
32 KHz				Class 1	Class 4	
44.1 KHz					Class 1	

Table 1. Classes of Resampling Rate

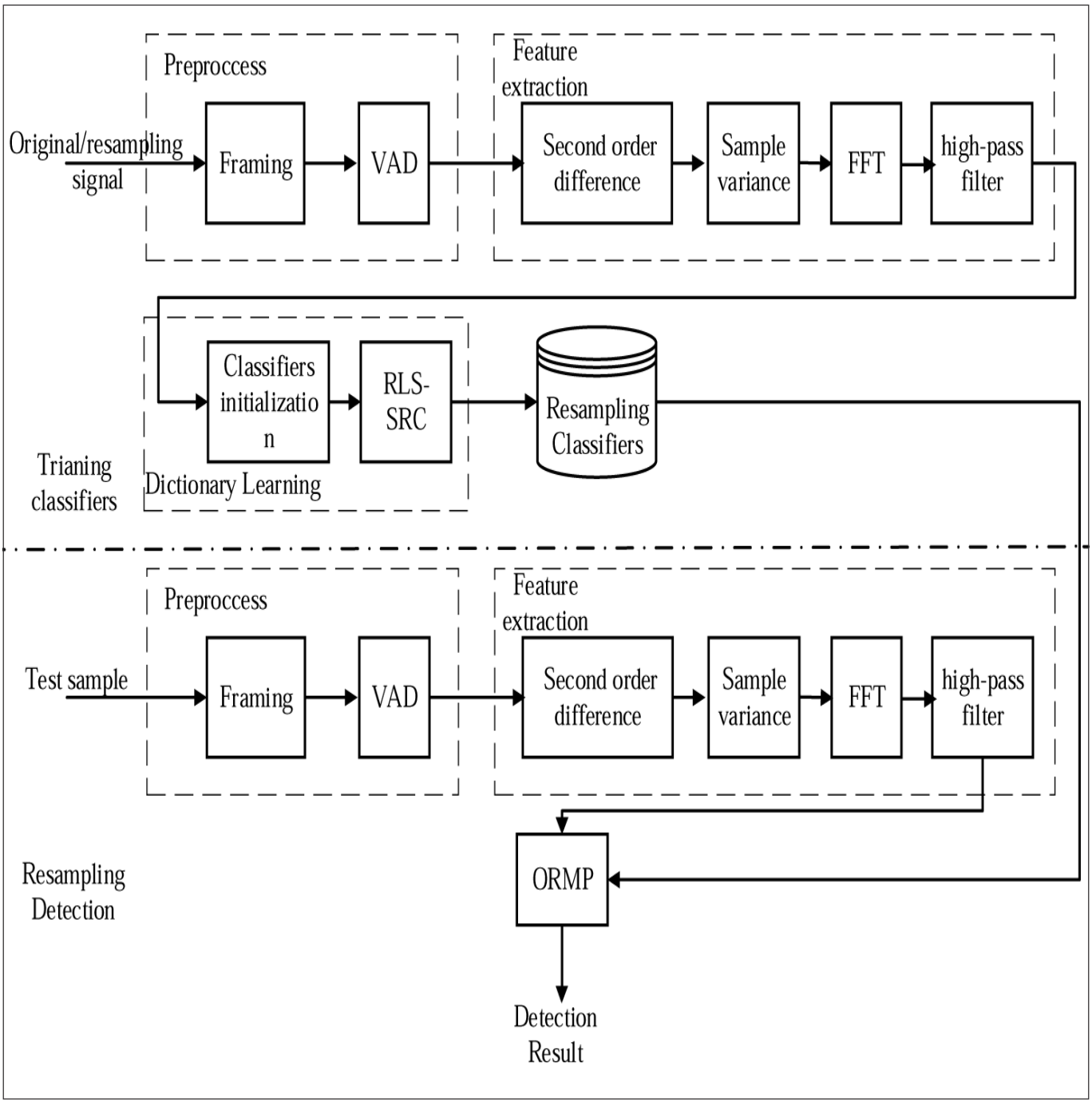
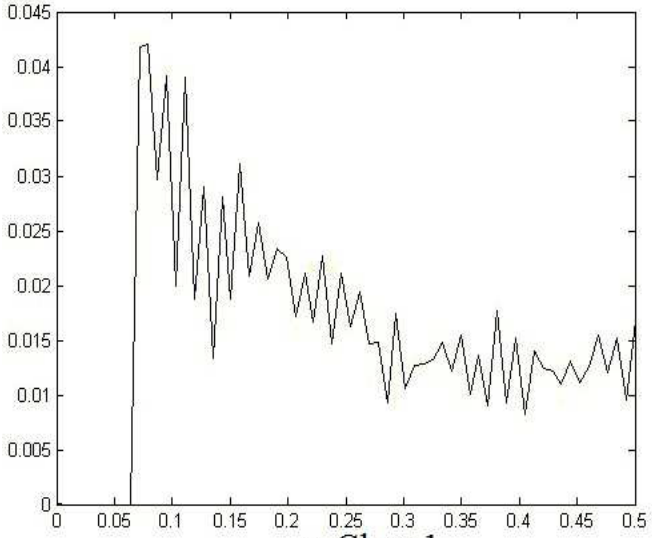
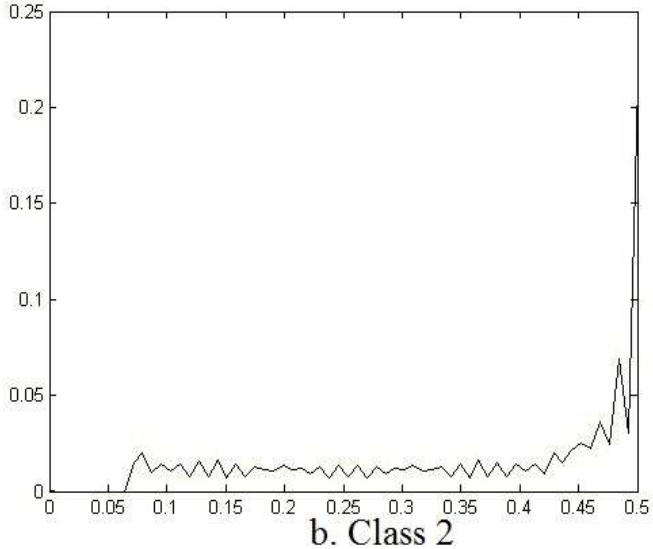


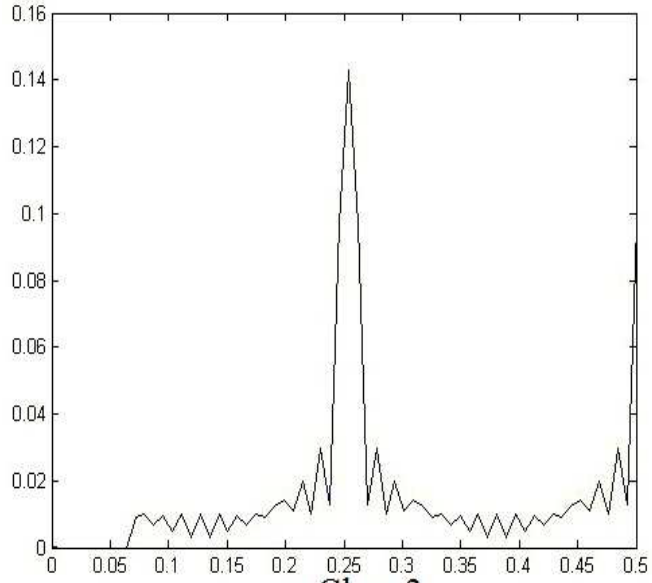
Figure 3. The principle of Resampling Detection



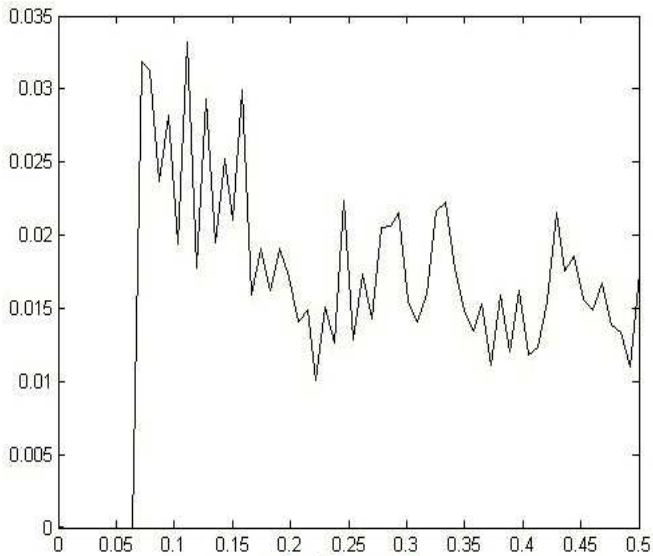
a. Class 1



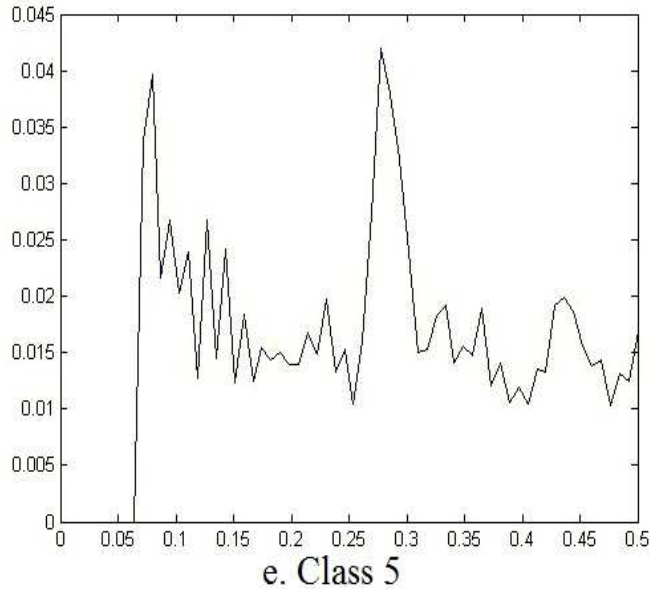
b. Class 2



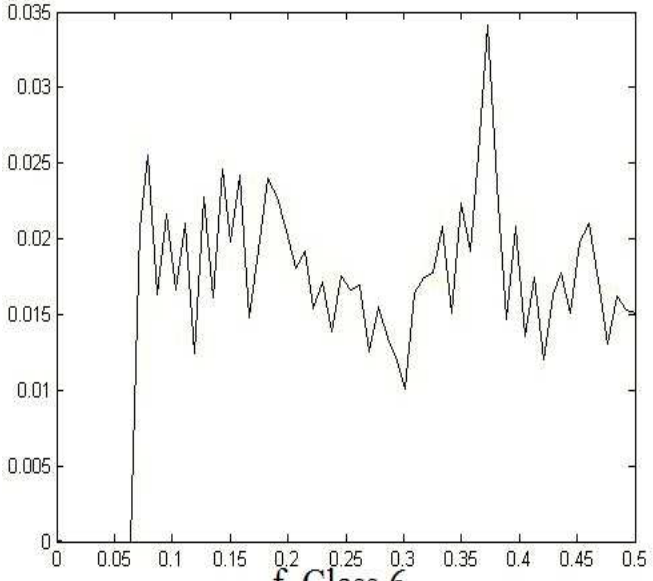
c. Class 3



d. Class 4



e. Class 5



f. Class 6

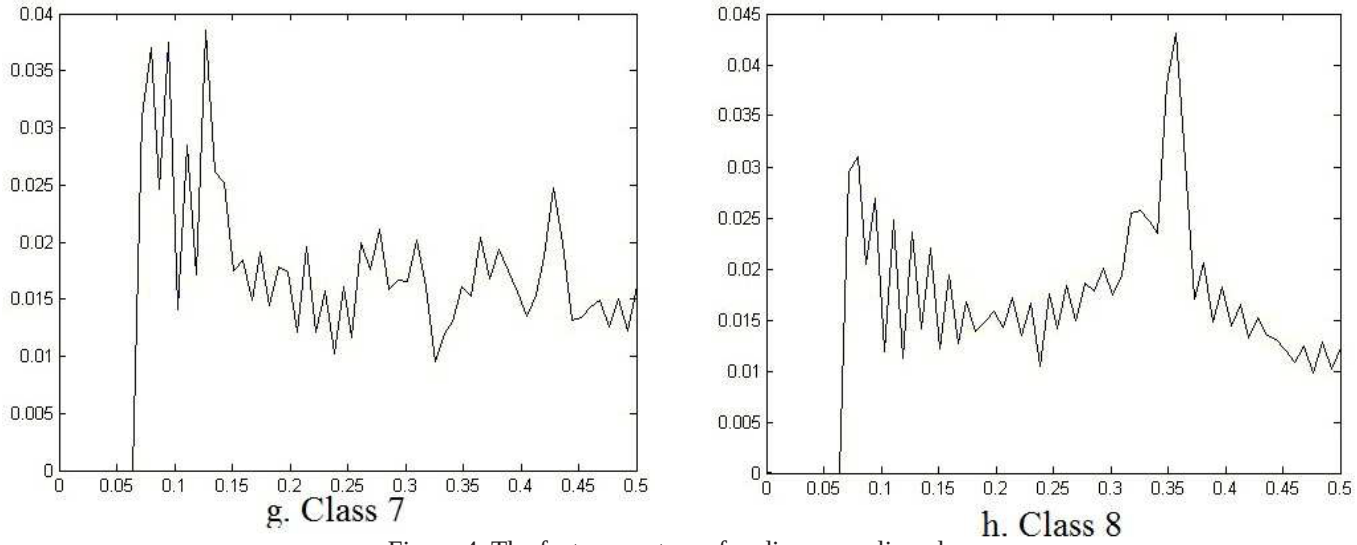


Figure 4. The feature vectors of audio resampling class

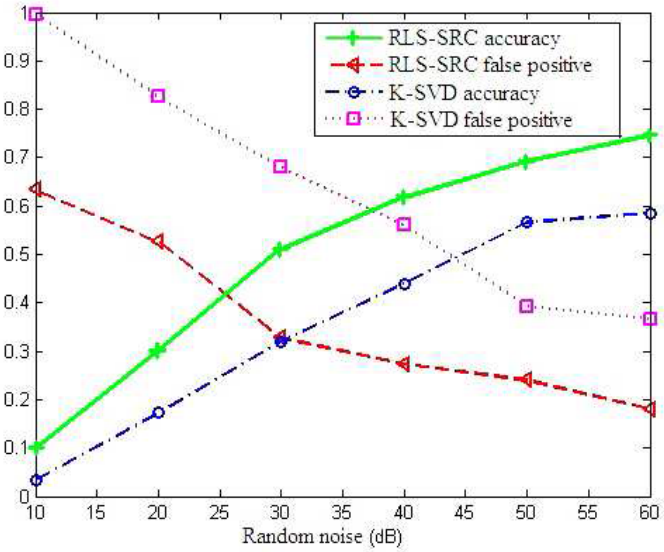


Figure 5. The Experimental Results of RLS-SRC vs. K-SVD

accuracy and low false positive rate.

Conclusion

This work explored the framework of online SRCs and proposed a method that can support online updates, which is a feature that is suitable for pattern recognition problems in digital information and multimedia. The proposed method solutes for large-scale classification, which has the advantage that existing SRCs do not have. It uses periodicity in the second derivative of an audio signal as a resampling feature, which can be used for the digital audio authenticity. Compared with the K-SVD-based method, the RLS-SRC has better detection ability and robustness under the interference of background noise.

Acknowledgements

The related works are supported by Social development & livelihood technology Program of Guiyang S&T Bureau (Qian Ke He GY Word (2012103) 74), Teaching reform research Program of Guizhou University (Xiao Jiao Fa Word (2013)58).

References

[1] Farid, H. (2009). A survey of Image forgery detection. *Signal Processing Magazine,* 26 (2),16-25.

[2] Malik, H., Mahmood, H. (2014). Acoustic environment identification using unsupervised learning. *Security Informatics*, 3 (1), 1-17.

[3] Gupta, S., Cho, S., Kuo, C C J. (2012). Current developments and future trends in audio authentication. *MultiMedia,* 19 (1), 50-59.

[4] Zuo Juxian., Pan Shenjun., Liu Benyong., Liao Xiang. (2011). Tampering detection for composite images based on re-sampling and JPEG compression. *In: Proc. of 2011 First Asian Conference on Pattern Recognition (ACPR),* Beijing, China: *IEEE,* p 169-173. Nov. 2011.

[5] Hsu Chih-Wei., Lin Chih-Jen. (2002). A comparison of methods for multiclass support vector machines. *IEEE Transactions on Neural Networks*, 13 (2), 415-425.

[6] Wright, J., Yang, A. Y., Ganesh, A., *et al*. (2009). Robust face recognition via sparse representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31 (2), 210-227.

[7] Mairal J., Bach, F., Ponce, J. (2012). Task-driven dictionary learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34 (4), 791-804.

[8] Xu Jing., Liu Benyong., Liao Xiang. (2011). Speech signal representation via dictionary learning in STFT transformed domain. In: *Proc. of 2011 International Conference on Multimedia and Signal Processing (CMSP'11),* Guilin, China:*IEEE*, p 20-24. May.

[9] Aharon, M., Elad, M., Bruckstein, A. (2006). K-SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation. *IEEE Transactions on Signal Processing*, 54 (11), 4311-4322.

[10] Oja, E. (1982). Simplified neuron model as a principal component analyzer. *Journal of mathematical biology*, 15(3), 267-273.

[11] Honeine, P. (2012). Online kernel principal component analysis: A reduced-order model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34 (9), 1814-1826.

[12] Skretting, K., Engan, K. (2010). Recursive least squares

dictionary learning algorithm. *IEEE Transactions on Signal*

Processing, 58 (4), 2121-2130.

[13] Gallagher, A. C. (2005). Detection of linear and cubic interpolation in JPEG compressed images. In: *Proc. of the 2nd Canadian Conference on Computer and Robot Vision (CRV'05)*, British Columbia, Canada: *IEEE Computer Society,* p 65-72. May.

[14] Rubinstein, R., Zibulevsky, M., Elad, M. (2008). Efficient implementation of the K-SVD algorithm using batch orthogonal matching pursuit. *CS Technion,* 40 (8), 1-15.

[15] Cotter, S. F., Rao, B. D., Kreutz-Delgado, K., *et al* (1999). Forward sequential algorithms for best basis selection. *IEEE Proceedings-Vision, Image and Signal Processing*, 146 (5),