



Article Discriminative Convolutional Sparse Coding of ECG Signals for Automated Recognition of Cardiac Arrhythmias

Bing Zhang and Jizhong Liu *

Nanchang Key Laboratory of Medical and Technology Research, School of Advanced Manufacturing, Nanchang University, Nanchang 330031, China

* Correspondence: liujizhong@ncu.edu.cn

Abstract: Electrocardiogram (ECG) is a common and powerful tool for studying heart function and diagnosing several abnormal arrhythmias. In this paper, we present a novel classification model that combines the discriminative convolutional sparse coding (DCSC) framework with the linear support vector machine (LSVM) classification strategy. In the training phase, most existing convolutional sparse coding frameworks are unsupervised in the sense that label information is ignored in the convolutional filter training stage. In this work, we explicitly incorporate a label consistency constraint called "discriminative sparse-code error" into the objective function to learn discriminative dictionary filters for sparse coding. The learned dictionary filters encourage signals from the same class to have similar sparse codes, and signals from different classes to have dissimilar sparse codes. To reduce the computational complexity, we propose to perform a max-pooling operation on the sparse coefficients. Using LSVM as a classifier, we examine the performance of the proposed classification system on the MIT-BIH arrhythmia database in accordance with the AAMI EC57 standard. The experimental results show that the proposed DCSC + LSVM algorithm can obtain 99.32% classification accuracy for cardiac arrhythmia recognition.

Keywords: electrocardiogram signal; discriminative convolutional sparse coding; dictionary filter learning; linear SVM

MSC: 68U01

1. Introduction

Electrocardiogram (ECG) is used to record cardiac activity and detect different abnormalities in cardiac function and is a commonly used non-invasive tool for non-invasive diagnosis of cardiac arrhythmias. However, visual analysis is extremely limited and imprecise due to the large amount of information contained in the ECG, which may lead to misdiagnosis or inaccurate detection of arrhythmias. Therefore, computer-aided analysis helps doctors to detect cardiac arrhythmias quickly and efficiently.

The automatic arrhythmia detection system mainly includes feature extraction, feature selection, and classifier construction. ECG signal feature extraction techniques can be divided into time-based methods [1–3], frequency methods [4–6], and time-frequency techniques [7,8]. The time domain features mainly include heartbeat interval, duration parameters, and amplitude parameters. Due to the subtle changes in ECG amplitude and duration, time-based methods do not provide good discrimination [9,10]. Therefore, frequency methods using such as Fourier transform and power spectral density (PSD) and time-frequency methods using wavelet transform are proposed. However, the frequency method does not provide time information from the ECG signal. Time-frequency technology based on wavelet transform is widely used in time-frequency feature extraction of ECG signals. Before the time-frequency feature vectors extracted by wavelet transform are applied to the classifier, it is important to choose the best dimensionality reduction method.



Citation: Zhang, B.; Liu, J. Discriminative Convolutional Sparse Coding of ECG Signals for Automated Recognition of Cardiac Arrhythmias. *Mathematics* **2022**, *10*, 2874. https://doi.org/10.3390/ math10162874

Academic Editors: Carmen Lacave and Ana Isabel Molina

Received: 13 July 2022 Accepted: 9 August 2022 Published: 11 August 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https:// creativecommons.org/licenses/by/ 4.0/). Dimensionality reduction methods for linear and non-linear transforms based on wavelet transforms have been proposed [7,11–13]. Martis et al. [7] transformed the ECG heartbeat using DWT and then applied independent component analysis (ICA) methods to extract features. The experimental results show that the features extracted using the ICA method combined with a probabilistic neural network (PNN) have better classification results. In the classification step, the most commonly used techniques in ECG signal classification are support vector machines (SVM) [1,14] and artificial neural networks [15].

In this paper, unlike most existing methods, we propose a novel classification model that combines the discriminative convolutional sparse coding framework with the linear support vector machines (LSVM) classification strategy.

The convolutional sparse coding (CSC) model assumes that the signal can be represented as a superposition of a few local filters, convolved with sparse feature maps. The CSC model handles the signal globally, and yet pursuit and dictionary learning are feasible due to the specific structure of the dictionary involved. CSC has been utilized for a variety of computer vision and pattern recognition tasks, such as inpainting [16], image separation [17], image fusion [18], object recognition [19], pedestrian detection [20], and tissue classification [21].

Although the CSC model has achieved state-of-the-art performance in many fields, most of the existing CSC frameworks are unsupervised and ignore label information in the convolutional filter training stage. Chen et al. [22] proposed a novel convolutional sparse coding classification (CSCC) approach, which introduces label information during the training process. The results show that dictionaries trained by convolution with label information can obtain more representative image information and achieve better classification performance. However, in the CSCC model, a set of dictionary filters are trained for each class of data, and then the dictionary filters from all classes build the final discriminative filter bank. This is not an efficient approach. At the same time, the intra-class and inter-class information are not considered, so the coding coefficients have small within-class scatter but large between-class scatter.

How to improve the classification performance of convolutional sparse coding has become a research direction in this field. One of the solutions is to obtain a discriminative dictionary filter bank through training so that the samples of different classes encoded by the dictionary filter bank are discriminative. Methods for learning a discriminative dictionary for sparse coding from training data have been recently proposed. Zhang and Li [23] proposed a discriminative KSVD dictionary learning algorithm. The algorithm incorporated the "classification error" term into the objective function, obtained the dictionary through the KSVD algorithm, and finally realized the classification task through a linear classifier. Yang et al. [24] learned a structured dictionary with class labels by adding Fisher's discriminant criterion during encoding. The method considers the intra-class and inter-class information in the coding process so that the coding coefficients have small within-class scatter but large between-class scatter. Following the work in [23], Jiang et al. [25] proposed to incorporate a "discriminative sparse code error" term into the objective function to enhance the discriminative power. It forces the signals from the same class to have very similar sparse representations, which results in good classification performance even using a simple linear classifier.

Obtaining a discriminative dictionary for classification by sparse coding has been successfully applied and achieved excellent performance. However, the discriminative dictionary obtained by sparse coding also has shortcomings, that is, it cannot capture the shifted local features in the sample, and at the same time, for high-dimensional signals, there is a curse of dimensionality. The dictionary filter bank obtained by convolutional sparse coding has shift invariance, local features at the sample translation position are extracted by convolution, and there is no curse of dimensionality.

However, how to learn discriminative convolutional dictionary filters with convolutional sparse representation and classification functions by supervised training is still an issue worth investigating. In this work, we propose a discriminative convolutional sparse coding (DCSC) model to learn discriminative dictionary filters for sparse coding of ECG signals. Different from the CSCC algorithm, we explicitly incorporate a label consistency constraint called "discriminative sparse-code error" into the objective function, transform the DCSC model to the Fourier domain to reduce the computational cost [26], and optimize the DCSC model using the alternating direction method of the multiplier framework [27] (ADMM) algorithm. By adding label information to the training phase of the convolutional dictionary filters, the learned dictionary filters encourage signals from the same class to have similar sparse codes, and signals from different classes to have dissimilar sparse codes. To reduce the computational complexity, we use the max-pooling operation on the sparse coefficients. These pooled coefficients are then used as features and fed to an LSVM classifier for the ECG classification task.

Our contributions are as follows:

- We propose a discriminative convolutional sparse coding (DCSC) model in which the "discriminative sparse-code error" is inserted into the objective function.
- In the process of solving the objective function, the DCSC model is first transformed into the Fourier domain, the convolution operation is converted into a multiplication operation, and then the function solution is obtained using the alternating direction method of the multiplier framework.
- The discriminative sparse coefficients are obtained via convolutional sparse coding, then dimensionally reduced by the max-pooling method, and finally fed into the LSVM classifier to complete the ECG classification task.

The rest of the paper is organized as follows: Section 2 presents the background of CSC, CDL, and LC-KSVD. Section 3 describes the proposed DCSC model. Section 4 describes the experimental results. The conclusion is given in Section 5.

2. Literature Survey

2.1. Convolutional Sparse Coding

The convolutional sparse coding (CSC) model assumes that a signal $\mathbf{s} \in \Re^N$ can be represented by the sum of M convolutions. These are built by feature maps $\{\mathbf{x}_m\}$, each of length N, convolved with M small support filters $\{\mathbf{d}_m\}$ of length $n \ll N$, * denotes convolution, and the CSC model can be formulated as:

$$\underset{\{\mathbf{x}_m\}}{\operatorname{argmin}} \frac{1}{2} \left\| \sum_m \mathbf{d}_m * \mathbf{x}_m - \mathbf{s} \right\|_2^2 + \lambda \sum_m \|\mathbf{x}_m\|_1 \tag{1}$$

where $\lambda > 0$ is a regularization parameter. Given the filters, the above problem becomes the CSC pursuit task of finding the representations {**x**_{*m*}}.

2.2. Convolutional Dictionary Learning

A common approach for convolutional dictionary learning (CDL) [26] entails using a batch of *K* training data to optimize the filters and sparse coefficient maps. This problem can be formulated as follows:

$$\underset{\{\mathbf{d}_m\},\{\mathbf{x}_{m,k}\}}{\operatorname{argmin}} \frac{1}{2} \sum_{k} \left\| \sum_{m} \mathbf{d}_m * \mathbf{x}_{m,k} - \mathbf{s}_k \right\|_2^2 + \lambda \sum_{k} \sum_{m} \left\| \mathbf{x}_{m,k} \right\|_1 \text{ such that } \left\| \mathbf{d}_m \right\|_2 = 1 \ \forall m$$
 (2)

where the constraint on the norms of dictionary filters $\{\mathbf{d}_m\}$ is required to avoid the scaling ambiguity between filters and coefficients. We denote the number of filters and the amount of training data by *M* and *K*, respectively.

The CDL problem is usually addressed by alternating optimization with respect to $\{\mathbf{x}_{m,k}\}$ and $\{\mathbf{d}_m\}$. Several works have shown that solving (2) with respect to $\{\mathbf{d}_m\}$ can also be done effectively and efficiently using ADMM in the frequency domain [26,28].

2.3. Label Consistent KSVD

Jiang et al. [25] proposed a label consistent KSVD (LC-KSVD1) algorithm to learn a discriminative dictionary. The objective function is defined as follows:

$$<\mathbf{D}, \mathbf{A}, \mathbf{X}> = \arg\min_{\mathbf{D}, \mathbf{A}, \mathbf{X}} ||\mathbf{S} - \mathbf{D}\mathbf{X}||_{2}^{2} + \alpha ||\mathbf{Q} - \mathbf{A}\mathbf{X}||_{2}^{2} \text{ s.t. } ||\mathbf{x}_{i}||_{0} < T$$
 (3)

where $||\mathbf{S} - \mathbf{DX}||_2^2$ denotes the reconstruction error and **D** is the learned dictionary. $||\mathbf{Q} - \mathbf{AX}||_2^2$ represents the discriminative sparse-code error, **Q** are the "discriminative" sparse codes of input signals **S** used for classification, and **A** is a linear transformation matrix. α is the scalar that controls the relative contribution of the corresponding term. **X** are the sparse codes of input signals **S** and \mathbf{x}_i is the column vector of **X**. *T* is a sparsity constraint factor (each signal has fewer than *T* non-zero items in its decomposition). By adding a discriminative sparse-code error constraint, the LC-KSVD1 model has good classification performance even with a simple linear classifier.

3. The Proposed ECG Signal Classification System

The proposed DCSC-based automatic arrhythmia identification system framework is shown in Figure 1. The proposed method consists of four stages, namely (1) in the discriminative convolutional sparse dictionary filters learning stage, we trained the DCSC model to obtain a discriminative dictionary filter. (2) In the sparse coding stage, we used the CSC model to obtain the discriminative sparse coefficients of the training and the test signals. (3) Pooling was performed on those sparse coefficients to reduce the large amount of data to an appropriate level. More importantly, it is used to obtain a compact representation of features that are invariant to local transformations. (4) In the ECG heartbeat testing stage, we used the pooled coefficients as features and fed them to an LSVM classifier for ECG classification.



Figure 1. Block diagram of the proposed ECG classification system.

3.1. Discriminative Convolutional Sparse Dictionary Learning Model

To improve the discriminative properties of the convolutional sparse coefficients, we incorporate the discriminative terms into the objective function during training. It can be formulated as the following optimization problem:

$$\underset{\{\mathbf{d}_m\},\{\mathbf{a}_m\},\{\mathbf{x}_{k,m}\}}{\operatorname{argmin}} \frac{1}{2} \sum_{k} \left\| \sum_{m} \mathbf{d}_m * \mathbf{x}_{m,k} - \mathbf{s}_k \right\|_2^2 + \frac{1}{2} \alpha \sum_{k} \left\| \sum_{m} \mathbf{a}_m * \mathbf{x}_{m,k} - \mathbf{q}_k \right\|_2^2 + \lambda \sum_{k} \sum_{m} \left\| \mathbf{x}_{m,k} \right\|_1$$
(4)
such that $\|\mathbf{d}_m\|_2 = 1, \|\mathbf{a}_m\|_2 = 1 \forall m$

where \mathbf{s}_k is the k^{th} training data set, $\mathbf{s}_k \in \Re^N$. $\lambda > 0$ is a regularization parameter and α is the scalar that controls the relative contribution of the corresponding terms. $\{\mathbf{d}_m\}$ and $\{\mathbf{a}_m\}$ are a set of M dictionary filters, and $\{\mathbf{x}_{m,k}\}$ is a set of coefficient maps corresponding to the m^{th} dictionary filter and the k^{th} training data, each having the same size as \mathbf{s}_k . $\mathbf{q}_k = [0, \ldots, 1, 1, \ldots, 0]^t \in \Re^N$ is a "discriminative" sparse code corresponding to an input signal \mathbf{s}_k , and \mathbf{q}_k is the column vector of \mathbf{Q} , whose size is the same as \mathbf{s}_k . For example,

assuming that $\mathbf{S} = [\mathbf{s}_0, \mathbf{s}_1, \mathbf{s}_2, \mathbf{s}_3]$, where $\mathbf{s}_0, \mathbf{s}_1$ are from class 1, $\mathbf{s}_2, \mathbf{s}_3$ are from class 2, and the length of the training data set N = 6, then \mathbf{Q} can be defined as:

$$\mathbf{Q} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix}$$
(5)

The codes $\{\mathbf{x}_{m,k}\}$ and the dictionary filters $\{\mathbf{d}_m\}$ and $\{\mathbf{a}_m\}$ can then be efficiently updated in an alternating manner, as follows.

A. Convolutional sparse coding (CSC) step

Given the dictionary filters $\{\mathbf{d}_m\}$ and $\{\mathbf{a}_m\}$, the codes can be updated by using ADMM in the frequency domain.

$$\operatorname{argmin}_{\{\mathbf{x}_{m,k}\}} \frac{1}{2} \sum_{k} \left\| \sum_{m} \mathbf{d}_{m} * \mathbf{x}_{m,k} - \mathbf{s}_{k} \right\|_{2}^{2} + \frac{1}{2} \alpha \sum_{k} \left\| \sum_{m} \mathbf{a}_{m} * \mathbf{x}_{m,k} - \mathbf{q}_{k} \right\|_{2}^{2} + \lambda \sum_{k} \sum_{m} \left\| \mathbf{x}_{m,k} \right\|_{1}$$
(6)

If we define the linear operators \mathbf{D}_m and \mathbf{A}_m such that $\mathbf{D}_m \mathbf{x}_{m,k} = \mathbf{d}_m * \mathbf{x}_{m,k}$, $\mathbf{A}_m \mathbf{x}_{m,k} = \mathbf{a}_m * \mathbf{x}_{m,k}$, and denote

$$\mathbf{D} = (\mathbf{D}_0 \quad \mathbf{D}_1 \quad \dots), \ \mathbf{A} = (\mathbf{A}_0 \quad \mathbf{A}_1 \quad \dots), \ \mathbf{S} = (\mathbf{s}_0 \quad \mathbf{s}_1 \quad \dots), \ \mathbf{Q} = (\mathbf{q}_0 \quad \mathbf{q}_1 \quad \dots), \ \mathbf{X} = \begin{pmatrix} \mathbf{x}_{0,0} \quad \mathbf{x}_{0,0} & \cdots \\ \mathbf{x}_{1,0} \quad \mathbf{x}_{0,0} & \cdots \\ \vdots \quad \vdots & \ddots \end{pmatrix}$$
(7)

then we can rewrite Equation (6) as

$$\underset{\mathbf{X}}{\operatorname{argmin}} \frac{1}{2} \|\mathbf{D}\mathbf{X} - \mathbf{S}\|_{2}^{2} + \frac{1}{2}\alpha \|\mathbf{A}\mathbf{X} - \mathbf{Q}\|_{2}^{2} + \lambda \|\mathbf{X}\|_{1}$$
(8)

The ADMM algorithm and shrinkage/soft thresholding algorithm can be employed to solve Equation (8), as described in Appendix A.

B. Convolutional dictionary update (CDU) step

In developing the dictionary filter update, it is convenient to switch the index of the coefficient map from $\{\mathbf{x}_{m,k}\}$ to $\{\mathbf{x}_{k,m}\}$. With the codes fixed, the dictionary filters $\{\mathbf{d}_m\}$ and $\{\mathbf{a}_m\}$ can be updated by solving the following optimization problems,

$$\underset{\{\mathbf{d}_m\},\{\mathbf{a}_m\},\{\mathbf{x}_{k,m}\}}{\operatorname{argmin}} \frac{1}{2} \sum_{k} \left\| \sum_{m} \mathbf{x}_{k,m} * \mathbf{d}_m - \mathbf{s}_k \right\|_2^2 + \frac{1}{2} \alpha \sum_{k} \left\| \sum_{m} \mathbf{x}_{k,m} * \mathbf{a}_m - \mathbf{q}_k \right\|_2^2$$
such that $\|\mathbf{d}_m\|_2 = 1, \|\mathbf{a}_m\|_2 = 1 \forall m$
(9)

Updating $\{\mathbf{d}_m\}$

With the $\{\mathbf{x}_{k,m}\}$ and $\{\mathbf{a}_m\}$ fixed, the dictionary filters $\{\mathbf{d}_m\}$ can be updated by solving the following optimization problem,

$$\underset{\{\mathbf{d}_m\}}{\operatorname{argmin}} \frac{1}{2} \sum_k \left\| \sum_m \mathbf{x}_{k,m} * \mathbf{d}_m - \mathbf{s}_k \right\|_2^2 \text{ such that } \|\mathbf{d}_m\|_2 = 1 \ \forall m$$
(10)

Addressing the CDL optimization problem (10) over $\{\mathbf{d}_m\}$ is equivalent to solving the following optimization problem

$$\underset{\{\mathbf{d}_m\}}{\operatorname{argmin}} \frac{1}{2} \sum_{k} \left\| \sum_{m} \mathbf{x}_{k,m} * \mathbf{d}_m - \mathbf{s}_k \right\|_2^2 + \sum_{m} \iota c_{PN}(\mathbf{d}_m)$$
(11)

where $\iota_{PN}(\mathbf{d}_m)$ is an indicator function [26]. Problem (11) can be solved efficiently using the consensus ADMM method, please refer to Appendix B.

Updating $\{\mathbf{a}_m\}$

With the $\{\mathbf{x}_{k,m}\}$ and $\{\mathbf{d}_m\}$ fixed, the dictionary filters $\{\mathbf{a}_m\}$ can be updated by solving the following optimization problem,

$$\operatorname{argmin}_{\{\mathbf{a}_m\}} + \frac{1}{2} \sum_k \left\| \sum_m \mathbf{x}_{k,m} * \mathbf{a}_m - \mathbf{q}_k \right\|_2^2 \text{ such that } \|\mathbf{a}_m\|_2 = 1 \ \forall m$$
(12)

Addressing the CDL optimization problem (12) over $\{a_m\}$ is similar to problem (10), please refer to Appendix B.

Algorithm 1: The DCSC Algorithm. **Input:** sample $\{\mathbf{s}_k\}$, parameters λ , α **Output:** $\{\mathbf{d}_m\}$ Precompute: $\mathbf{s}_k \rightarrow \hat{\mathbf{s}}_k$, $\mathbf{q}_k \rightarrow \hat{\mathbf{q}}_k$, **Initialize**: $\{\mathbf{Y}\} = \{\mathbf{U}\} = \{\mathbf{g}_m\} = \{\mathbf{h}_m\} = 0$, **while** *j* = 0 to convergence do (CSC step) Compute FFTs of $\{\mathbf{Y}\} \rightarrow \{\hat{\mathbf{Y}}\}, \{\mathbf{U}\} \rightarrow \{\hat{\mathbf{U}}\}, \{\mathbf{d}_m\} \rightarrow \{\hat{\mathbf{D}}\}, \{\mathbf{a}_m\} \rightarrow \{\hat{\mathbf{A}}\}$ Compute $\hat{\mathbf{X}}$ with the algorithm in Appendix A. Compute inverse FFTs of $\hat{X} \rightarrow X$ $\mathbf{Y}^{(j+1)} = S_{\lambda/\rho} (\mathbf{X}^{(j+1)} + \mathbf{U}^{(j)})$ $\mathbf{U}^{(j+1)} = \mathbf{U}^{(j)} + \mathbf{X}^{(j+1)} - \mathbf{Y}^{(j+1)}$ (CDU step) Compute FFTs of $\{\mathbf{x}_{k,m}\} \rightarrow \{\hat{\mathbf{X}}_k\}, \{\mathbf{g}_m\} \rightarrow \{\hat{\mathbf{g}}_m\}, \{\mathbf{h}_m\} \rightarrow \{\hat{\mathbf{h}}_m\}$ Compute $\hat{\mathbf{d}}$ with the algorithm in Appendix B. Compute inverse FFTs of $\hat{\mathbf{d}} \rightarrow \{\mathbf{d}_m\}$ Compute $\{\mathbf{g}_m\}$ with the algorithm in Appendix **B**. $\mathbf{h}_m^{(j+1)} = \mathbf{h}_m^{(j)} + \mathbf{d}_m^{(j+1)} - \mathbf{g}_m^{(j+1)}$ Compute $\{\mathbf{a}_m\}$ end

We trained the DCSC algorithm with the training data and obtained the dictionary filters. The dictionary filters $\{\mathbf{d}_m\}$ were then used to obtain the sparse coefficients of both the training and test signals for each class in the sparse coding stage.

3.2. Sparse Coding of Training and Test Signals

We obtained $\{\mathbf{d}_m\}$ by employing the DCSC algorithm (Algorithm 1). Sparse coefficient vectors $\{\mathbf{x}_{m,k}\}$ for each signal were obtained by using the CSC algorithm as follows:

$$\operatorname{argmin}_{\{\mathbf{x}_{m,k}\}} \frac{1}{2} \sum_{k} \left\| \sum_{m} \mathbf{d}_{m} * \mathbf{x}_{m,k} - \mathbf{s}_{k} \right\|_{2}^{2} + \lambda \sum_{k} \sum_{m} \|\mathbf{x}_{m,k}\|_{1}$$
(13)

The sparse coefficient vectors of each signal thus obtained were then combined columnwise to form a vector of larger dimensions. This process is depicted in Figure 2. The sparse coefficients of these training data were used to train the LSVM model for ECG classification.



Figure 2. Extraction of sparse coefficients using convolutional sparse coding.

3.3. Pooling of Coefficient Matrix

Inspired by the feature extraction methods [29,30], we apply max-pooling methods to each column of the sparse coefficient matrix X.

The max-pooling function

$$\mathbf{f}_k = \max(\mathbf{x}_k^{n \times 1} u(n, 1)) \tag{14}$$

applies a window function u(n, 1) to each column of the sparse coefficient matrix X and computes the maximum value in the neighborhood.

These pooled features can then be ℓ_2 normalized by

$$\mathbf{f}_k = \mathbf{f}_k / \|\mathbf{f}_k\|_2 \tag{15}$$

These pooled sparse coefficients from each sub-region are concatenated and normalized to the final feature representation for the classification.

3.4. Classification by LSVM

The schematic diagram of the proposed ECG classification system is shown in Figure 3. By learning discriminative dictionary filters, it forces signals from the same class to have very similar sparse representations, so we use the LSVM for our ECG classification task. In the classification stage, each class in the training dataset contains 300 training samples, and the test dataset contains 51,722 ECG segments.



Figure 3. ECG classification using LSVM.

4. Experiments and Discussion

In this section, we test on the PhysioNet MIT-BIH arrhythmia database and compare the performance of the DCSC + LSVM method with other state-of-the-art classifiers.

4.1. Dataset

The proposed methodology is validated on the PhysioNet MIT-BIH Arrhythmia database [31] comprising 48 ECG records of about 30 min, sampled at 360 Hz with 11-bit resolution from 47 different patients. As recommended by the American Association of Medical Instrumentation (AAMI) [32], the MIT-BIH arrhythmia database is projected into five AAMI heartbeat classes, as described in Table 1. The ECG signal downloaded from the MIT-BIH database is processed with the help of wavelet techniques to remove baseline

wander and high-frequency noise. The ECG signals sampled at 360 Hz were decomposed up to eight levels using the Bior2.6 wavelet to remove various kinds of noise [33]. The ECG would not contain much information after 45 Hz. The high-frequency noise is removed by excluding the first and second detail coefficients, which consist of bands of 90–180 Hz and 45–90 Hz. The baseline wander is removed by excluding the approximate coefficients at the fourth level. After the corresponding wavelet reconstruction, it is obvious that we can obtain a denoised ECG signal. Then, the R-peak is detected from the denoised ECG signal followed by a window across each R-peak to isolate the ECG segments for processing. After detection of the QRS complex, 99 samples preceded the QRS peak and 180 samples after the peak, and the QRS peak itself are considered as 280 samples segment as a single beat for subsequent analysis. The mapping of the MIT-BIH arrhythmia database into the AAMI recommendations along with the summary of the training and testing datasets for five classes is summarized in Table 2.

Table 1. Heartbeats of the MIT-BIH arrhythmia database classified based on the ANSI/AAMIEC57:1998 standard.

	Ν	S	V	Q	
• • •	Normal Left bundle branch block Right bundle branch block Nodal (junctional) escape beat Atrial escape beat	 Atrial premature beat Aberrated atrial premature beat Nodal (Junctional) premature beat Supraventricular premature beat 	 Premature ventricular contraction Ventricular escape beat 	• Fusion of ventricular and normal beat	 Paced beat Fusion of paced and normal beat Unclassifiable beat

AAMI Classes	Training Data	Testing Data	Total Data
N	300	40,212	40,512
S	300	1388	1688
V	300	4610	4910
F	300	501	801
Q	300	5011	5311
Total	1500	51,722	53,222

Table 2. A summary of the 5 classes of beat subtypes.

4.2. Signal Preprocessing

Roshan et al. [7,11] pointed out that through wavelet transform decomposition, the power spectral density of each type of different heartbeat has discriminative information in these two sub-bands (level 4 approximation and detail), and independent component analysis (ICA) based on wavelet transform coefficients has strong robustness and high classification accuracy. Each beat consisting of 280 samples was decomposed into four levels using the FIR approximation of Mayer's wavelet ("dmey"). The ICA method was applied independently to the two DWT sub-bands, the fourth level approximation, and the details [7]. From each of the sub-bands fifteen ICA components were selected, so a total of thirty features from the two sub-bands were selected for subsequent pattern recognition.

4.3. Parameter Selection

In the first set of experiments reported in Figure 4, we compare the effects of λ and α values on classification accuracy. The λ and α values vary from 10^{-5} to 0. We can observe that good performance is achieved at $\lambda = 10^{-1}$ and $\alpha = 10^{-2}$.



Figure 4. Effects of λ and α parameter selection on classification accuracy.

The effect of the number of iterations on the classification accuracy is shown in Figure 5. With the increase in the number of iterations, the accuracy also increases. When the number of iterations increases to 130, the accuracy rate changes very little and stabilizes to a certain value. In order to balance accuracy and complexity, the number of iterations was fixed at 130.



Figure 5. The effect of the number of iterations on the classification accuracy.

Figure 6 shows the classification accuracy when the variable dictionary filter dimension is changed from 6 to 20. In this experiment, we choose the dictionary size M = 128, and the number of training samples per class is 300. However, the overall trend is that the dictionary filter dimension has little effect on the classification performance.



Figure 6. Classification accuracy based on variable dictionary filter dimension.

Figure 7 shows the classification results for the variable dictionary size *M* changing from 128 to 576. We can see that DCSC + LSVM shows an improvement of about 2.3% over LC-KSVD in all cases. In particular, even with the dictionary size M = 128, DCSC + LSVM can still have higher classification accuracy than LC-KSVD with M = 576. Moreover, from dictionary size M = 576 to M = 128, the recognition rate of DCSC + LSVM drops by 1.41%, while that of LC-KSVD drops by 6.25%. In the following classification experiments, we choose the dictionary filter dimension M = 128.



Figure 7. Classification accuracy based on variable dictionary size M.

Figure 8 shows the classification results with different numbers of training samples per class. We trained 100, 150, 200, 250, 300, 350, and 400 samples per class. We can see that the different number of training samples per class has little effect on the classification performance of the DCSC + LSVM algorithm, and in all cases, the classification results of the DCSC + LSVM algorithm outperform the LC-KSVD algorithm. The classification performance of the LC-KSVD algorithm is strongly influenced by the number of training samples per class. When the number of training samples per class is less than 250, the classification results of the LC-KSVD algorithm differ from those of the DCSC + LSVM algorithm by a maximum of 9%. In the following classification experiments, we set 300 training samples per class for training.



Figure 8. Classification accuracy based on different numbers of training samples per class.

Figure 9 shows the effect of pooling kernel size on classification accuracy. In this experiment, the pooling stride is 2, and the pooling kernel is increased from 3 to 8. It can be seen from Figure 9 that with the increase of the pooling kernel, the classification accuracy decreases. At the same time, comparing the effects of max pooling and average pooling on the classification accuracy, it can be seen that the classification effect of max pooling is better than that of average pooling. So, we choose the max-pooling method, and the pooling kernel is set to 3.



Figure 9. Effects of the Pooling Kernel Size on the Classification Accuracy.

4.4. Statistical Parameters

In order to analyze the performance of the classifiers, four evaluation metrics are used: sensitivity (*SEN*), positive predictive value (*PPV*), F_1 , and accuracy (*Acc*). The *SEN*, *PPV*, F_1 , and *Acc* can be written as

$$SEN = TP/(TP + FN) \times 100 \tag{16}$$

$$PPV = TP/(TP + FP) \times 100 \tag{17}$$

$$F_1 = 2TP/(2TP + FP + FN) \times 100 \tag{18}$$

$$Acc = (TP + TN)/(TP + FP + FN + TN) \times 100$$
⁽¹⁹⁾

where true positive (*TP*) corresponds to the number of times that the classifier correctly predicts a heartbeat without arrhythmia, i.e., normal. False positive (*FP*) gives the number of arrhythmic heartbeats classified as normal. True negative (*TN*) quantifies the number of heartbeats with arrhythmia that are predicted correctly. False negative (*FN*) indicates the total of normal beats misclassified as arrhythmic. F_1 being defined as the harmonic mean of precision and sensitivity.

4.5. Results

We evaluated our approach on the PhysioNet MIT-BIH arrhythmia database and compared our approach with LC-KSVD [25], Fisher discrimination dictionary learning (FDDL) [24], class-specific dictionary learning (CSDL) [34], Euler label consistent KSVD (ELC-KSVD) [35], convolutional sparse coding classification (CSCC) [22], label embedded dictionary learning (LEDL) [36]. The hardware platform is Intel Xeon Gold 5122 CPU 3.60 GHz, 128.0 GB RAM, and the experimental analysis is performed by the MATLAB 2015b software package installed on the Windows 10 21H2 Microsoft USA platform. During the experiment, we also adjust the parameters of other classifiers to obtain the best classification effect. For example, in the LC-KSVD algorithm, $\alpha = 2^{-2}$ and $\beta = 2^{-6}$. In the FDDL algorithm, $\lambda_1 = 0.005$, $\lambda_2 = 0.05$. In the CSDL algorithm, $\alpha = 10^{-1}$, $\omega = 10^{-5}$, and $\varepsilon = 10^{-3}$. The number of atoms per class for the FDDL, CSDL, and CSCC algorithms is the same as the number of training samples per class, which is 300. The dictionary atomic number of LC-KSVD, ELC-KSVD, and LEDL algorithms is set to 1500. The experimental results are as follows.

Figure 10 illustrates the confusion matrix using the LSVM method. The test dataset contains 51,722 ECG signal segments, the accuracy of the LSVM method was 87.09%, and 45,044 ECG signals were correctly classified. Figure 11 demonstrates the confusion matrix using the LC-KSVD method. The accuracy of the LC-KSVD method was 96.97%, and 50,156 ECG signals were correctly classified. Figure 12 illustrates the confusion matrix using the FDDL method. The accuracy of the FDDL method was 98.80%, and 51,100 ECG signals were correctly classified. Figure 13 illustrates the confusion matrix using the CSDL method. The accuracy of the CSDL method was 94.15%, and 48,698 ECG signals were correctly classified. Figure 14 illustrates the confusion matrix using the ELC-KSVD method. The accuracy of the ELC-KSVD method was 99.02%, and 51,213 ECG signals were correctly classified. Figure 15 illustrates the confusion matrix using the CSCC method. The accuracy of the CSCC method was 94.70%, and 48,983 ECG signals were correctly classified. Figure 16 illustrates the confusion matrix using the LEDL method. The accuracy of the LEDL method was 95.17%, and 49,222 ECG signals were correctly classified. Figure 17 illustrates the confusion matrix using the DCSC + LSVM method. The accuracy of the DCSC + LSVM method was 99.32%, and 51,371 ECG signals were correctly classified.

N	36,463	451	32	1588	1678
	,				
S	- 98	1168	0	51 -	
V	958	38	2251	701	662 -
F	- 8	0	0	491	2
Q	142	33	47	118	4671
	N	S	V	F	Q

Figure 10. Confusion matrix using the LSVM model.

Ν	- 39,134	232	284	338	224
S	20	1318	21	19	10
V	79	47	4381	58	45
F	13	6	7	467	8
Q	41	27	42	45	4856
	N	S	V	F	Q

Figure 11. Confusion matrix using the LC-KSVD model.

Ν	39,781	96	213	34	88
S	- 5	1379	3	1	0
V	8	31	4571	0	0
F	0	2	2	495	2
Q	18	18	91	10	4874
	N	S	V	F	Q

Figure 12. Confusion matrix using the FDDL model.

Ν	38,155	398	812	216	631
S	- 58	1254	40	12	24
V	218	108	4056	57	171
F	- 18	19	14	438	12
Q	72	41	86	17	4795
	N	S	V	F	Q

Figure 13. Confusion matrix using the CSDL model.

Ν	39,895	78	181	24	34
S	- 1	1377	7	1	2
V	13	39	4536	16	6
F	- 5	2	5	489	0
Q	12	39	35	9	4916
	N	S	V	F	Q

Figure 14. Confusion matrix using the ELC-KSVD model.

Ν	38,502	436	894	2	378
S	55	1217	57	54 -	
V	277	48	4150	6	129
F	- 20	12	14	449	6
Q	126	36	180	4	4665
	N	S	V	F	Q

Figure 15. Confusion matrix using the CSCC model.

	N	S	V	F	Q
Q	139	30	222	4	4616
F	17	10	22	448	4
V	212	73	4242	0	83 -
S	66	1213	61	45	
Ν	38,703	255	1013	2	239
	· · · · · · · · · · · · · · · · · · ·				

Figure 16. Confusion matrix using the LEDL model.

Ν	40,037	42	88	1	44
S	- 7	1356	16	1	8
0		1000	10		0
V	- 11	12	4574	0	13
F	8	5	11	469	8
Q	32	13	31	0	4935
	Ň	S	V	F	Q

Figure 17. Confusion matrix using the DCSC + LSVM model.

Table 3 shows the classification performance (*Acc, SEN, PPV*, and *F*₁) obtained by different models on the test dataset. Compared with the LSVM algorithm, after DCSC coding, the classification accuracy is improved by 12.23%. Meanwhile, compared with LC-KSVD, FDDL, CSDL, ELC-KSVD, CSCC and LEDL, the accuracy of the proposed DCSC + LSVM is improved by 2.35%, 0.52%, 5.17%, 0.3%, 4.62%, and 4.15%, respectively. The disadvantage is that 32 S-class and 32 F-class ECG beats are misclassified. This explains why the *SEN* of class S and class F are lower, and this also affects the *PPV* value of class N and Q. Future efforts will look for discriminable features which get higher accuracy for classifying these beats.

In this system, to complete the classification of a segment of ECG signal, it needs to go through four stages, namely, the DWT feature extraction stage, the DCSC encoding stage, the pooling stage, and the classification stage. Since the test results of the same ECG signal were slightly different each time, we calculated the average of 10 test times as the final time. Table 4 lists the calculation time of each stage. The results show that the DCSC coding stage requires the longest computation time, the classification stage requires the shortest computation time due to the use of the linear SVM classifier, and a segment of ECG signal needs 0.336 s to complete the classification process.

Method	Acc	Ν			S			V			F			Q		
	%	SEN	PPV	F_1												
LSVM	87.09	90.68	96.80	93.64	84.15	69.11	75.89	48.83	96.61	64.87	98.00	16.54	28.30	93.21	66.12	77.37
LC-KSVD	96.97	97.32	99.61	98.45	94.96	80.86	87.34	95.03	92.52	93.76	93.21	50.38	65.41	96.91	94.42	95.65
FDDL	98.80	98.93	99.92	99.42	99.35	90.37	94.65	99.15	93.67	96.33	98.80	91.67	95.10	97.27	98.19	97.72
CSDL	94.15	94.88	99.05	96.92	90.35	68.90	78.18	87.98	80.99	84.34	87.43	59.19	70.59	95.69	85.12	90.10
ELC-KSVD	99.02	99.21	99.92	99.57	99.21	89.71	94.22	98.39	95.21	96.78	97.60	90.72	94.04	98.10	99.15	98.63
CSCC	94.70	95.75	98.77	97.24	87.68	69.58	77.59	90.02	78.38	83.80	89.62	96.35	92.86	93.10	89.16	91.09
LEDL	95.17	96.25	98.89	97.55	87.39	76.72	81.71	92.02	76.29	83.42	89.42	98.03	93.53	92.12	92.56	92.34
DCSC + LSVM	99.32	99.56	99.86	99 71	97 69	94 96	96.31	99 22	96 91	98.05	93.61	99.58	96.50	98 48	98.54	98.51

Table 3. Performance metrics of different algorithms.

Table 4. Computation time for each stage for a test sample.

Methods	DWT + ICA	DCSC	Pooling Classification	
Time(s)	0.058	0.198	0.078	0.002

4.6. Discussion

In this work, we explicitly incorporated a label consistency constraint called "discriminative sparse-code error" into the objective function of learning discriminative dictionary filters for sparse coding. The learned dictionary filters encouraged signals from the same class to have similar sparse codes, and signals from different classes to have dissimilar sparse codes. These discriminative coding coefficients allowed us to obtain good classification results even with simple linear classifiers. Compared with the LSVM algorithm, after DCSC coding, the classification accuracy is improved by 12.23%, which means that by adding label information, after DCSC coding, the coding coefficients of different classes have more obvious differences. Compared with the CSCC algorithm, the algorithm trains filter banks for each class separately but does not use the intra-class and inter-class information of the training samples during the training process, so it does not achieve good classification results. The FDDL, LC-KSVD, CSDL, ELC-KSVD, and LEDL algorithms are all supervised training algorithms and incorporate label information. The reason why their classification effect is lower than the DCSC + LSVM algorithm is that the dictionary filter bank has shift-invariance. The local features at translated positions of the sample can be extracted through the convolution operation, so a better classification effect can be obtained. We also arranged comparative experiments between DCSC + LSVM algorithm and the LC-KSVD algorithm under different dictionary sizes and different training numbers. The experimental results show that the discriminative convolutional dictionary-based learning model can obtain better classification results than LC-KSVD at a smaller dictionary size. At the same time, it is less affected by the number of training datasets. This shows that the DCSC + LSVM algorithm can be applied to a database with a small number of samples.

A comprehensive summary of the automated classification of ECG beats using the MIT-BIH arrhythmia database is shown in Table 5. Mathews et al. [7] used ICA on DWT coefficients to extract features and reported an accuracy of 99.28% using the PNN classifier. Desai et al. [37] also used ICA on DWT coefficients to extract features and obtained 98.49% accuracy using SVM quadratic kernels. Elhaj et al. [38] used PCA on DWT and ICA on HOS cumulants as features and obtained 98.91% accuracy using the SVM-RBF classifier. Acharya et al. [39] classified with an accuracy of 94.03% using a 9-layer deep convolutional neural network. Kachuee et al. [40] used a deep residual CNN network to classify with an accuracy of 93.40%. Yildirim et al. [41] used CAE and LSTM networks to classify with an accuracy of 99.00%. Romdhane et al. [42] classified with an accuracy of 98.41% using a deep CNN model and a focal loss function. Li et al. [43] used a deep residual network to classify with an accuracy of 99.06%.

Literature	Features	Classifier	Classes	Acc
Mathews et al. [7]	DWT + ICA	PNN	5	99.28
Desai et al. [37]	DWT + ICA	SVM quadratic kernel	5	98.49
Elhaj et al. [38]	PCA + DWT + HOS + ICA	SVM-RBF	5	98.91
Acharya et al. [39]		9-layer deep convolutional neural network	5	94.03
M. Kachuee et al. [40]		deep residual CNN	5	93.40
Yildirim et al. [41]		CAE and LSTM	5	99.00
Romdhane et al. [42]		Deep CNN	5	98.41
Li et al. [43]		Deep residual network	5	99.06
Proposed	DWT + ICA	DCSC + LSVM	5	99.32

Table 5. Comparison of works on ECG heartbeat classification from MIT-BIH.

The results show that the proposed model achieves higher classification accuracy compared to existing works presented in the literature and can be utilized for automated computer-aided diagnosis of several cardiovascular diseases.

5. Conclusions

In this paper, we present a novel classification model that combines the DCSC model with the LSVM classification strategy. In this work, we explicitly incorporate a label consistency constraint called "discriminative sparse-code error" into the objective function and transform the DCSC model into the Fourier domain to reduce the computational cost and optimize the DCSC model using the ADMM algorithm. By adding label information to the training phase of the convolutional dictionary filters, the learned dictionary filters encourage signals from the same class to have similar sparse codes, and signals from different classes to have dissimilar sparse codes. To reduce the computational complexity, we used a max-pooling operation for the sparse coefficients. These pooled coefficients were then used as features and fed to the LSVM classifier for the ECG classification task. The evaluation and experiments on the MIT-BIH arrhythmia database for the five classes as recommended by AAMI validate the effectiveness of the proposed DCSC + LSVM method and show that the proposed DCSC + LSVM model outperforms LSVM, LC-KSVD, FDDL, CSDL, ELC-KSVD, CSCC, LEDL and achieves state-of-the-art performance.

Author Contributions: Conceptualization, B.Z. and J.L.; methodology, B.Z. and J.L.; software, B.Z. and J.L.; validation, B.Z.; investigation, B.Z. and J.L.; writing—original draft preparation, B.Z.; writing—review and editing, B.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Natural Science Foundation of China (Grant no. 61863027), the Key Research and Development Plan of Jiangxi Province (Grant no. 20202BBGL73057), the Natural Science Foundation of Jiangxi Province (Grant no. 20171BAB201013), and the Project of Nanchang Key Laboratory of Medical and Technology Research (Grant no. 2018-NCZDSY-002).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Conflicts of Interest: The authors declare that there are no conflicts of interest regarding the publication of this paper.

Appendix A. The Coding Algorithm

Using variable splitting, problem (8) in ADMM form can be reformulated as follows:

$$\underset{\mathbf{X}}{\operatorname{argmin}} \frac{1}{2} \|\mathbf{D}\mathbf{X} - \mathbf{S}\|_{2}^{2} + \frac{1}{2}\alpha \|\mathbf{A}\mathbf{X} - \mathbf{Q}\|_{2}^{2} + \lambda \|\mathbf{Y}\|_{1} \quad \text{s.t. } \mathbf{X} - \mathbf{Y} = 0$$
(A1)

for which we have the ADMM iterations with dual variables U, which are

$$\mathbf{X}^{(j+1)} = \underset{\{\mathbf{x}_m\}}{\operatorname{argmin}} \frac{1}{2} \|\mathbf{D}\mathbf{X} - \mathbf{S}\|_2^2 + \frac{1}{2}\alpha \|\mathbf{A}\mathbf{X} - \mathbf{Q}\|_2^2 + \frac{\rho}{2}\sum_m \|\mathbf{X} - \mathbf{Y}^{(j)} + \mathbf{U}^{(j)}\|_2^2$$
(A2)

$$\mathbf{Y}^{(j+1)} = \underset{\{\mathbf{y}_{m}\}}{\operatorname{argmin}} \lambda \sum_{m} \|\mathbf{Y}\|_{1} + \frac{\rho}{2} \sum_{m} \|\mathbf{X}^{(j+1)} - \mathbf{Y} + \mathbf{U}^{(j)}\|_{2}^{2}$$
(A3)

$$\mathbf{U}^{(j+1)} = \mathbf{U}^{(j)} + \mathbf{X}^{(j+1)} - \mathbf{Y}^{(j+1)}$$
(A4)

Sub-problem Equation (A3) is solved via shrinkage/soft thresholding as

$$\mathbf{Y}^{(j+1)} = S_{\lambda/\rho} (\mathbf{X}^{(j+1)} + \mathbf{U}^{(j)})$$
(A5)

where

$$S_{\gamma}(\mathbf{u}) = sign(\mathbf{u}) \odot \max(0, |\mathbf{u}| - \gamma)$$
(A6)

with $sign(\cdot)$ and $|\cdot|$ of a vector being considered to be applied element-wise and \odot denoting element-wise multiplication.

The most computationally expensive step is Equation (A2), which requires solving the linear system

$$(\mathbf{D}^{T}\mathbf{D} + \alpha \mathbf{A}^{T}\mathbf{A} + \rho \mathbf{I})\mathbf{X} = \mathbf{D}^{T}\mathbf{S} + \alpha \mathbf{A}^{T}\mathbf{Q} + \rho(\mathbf{Y} - \mathbf{U})$$
(A7)

Since $\mathbf{D}^T \mathbf{D}$ is a very large matrix, it is impractical to solve this linear system using the approaches that are effective when \mathbf{D} is not a convolutional dictionary. An obvious approach is to attempt to efficiently implement convolution via the DFT convolution theorem using the Fast Fourier Transform (FFT). The variables \mathbf{D} , \mathbf{A} , \mathbf{X} , \mathbf{S} , \mathbf{Q} , \mathbf{Y} , and \mathbf{U} in the DFT domain are denoted by $\mathbf{\hat{D}}$, $\mathbf{\hat{A}}$, $\mathbf{\hat{X}}$, $\mathbf{\hat{S}}$, $\mathbf{\hat{Q}}$, $\mathbf{\hat{Y}}$, and $\mathbf{\hat{U}}$, respectively. It is easy to demonstrate via the DFT convolution theorem that Equation (A7) is equivalent to

$$(\hat{\mathbf{D}}^{H}\hat{\mathbf{D}} + \alpha\hat{\mathbf{A}}^{H}\hat{\mathbf{A}} + \rho\mathbf{I})\hat{\mathbf{X}} = \hat{\mathbf{D}}^{H}\hat{\mathbf{S}} + \alpha\hat{\mathbf{A}}^{H}\hat{\mathbf{Q}} + \rho(\hat{\mathbf{Y}} - \hat{\mathbf{U}})$$
(A8)

Appendix B. Convolutional form of Method of Optimal Directions

The consensus ADMM formulation of problem (11) is given as

$$\underset{\{\mathbf{d}_m\},\{\mathbf{g}_m\}}{\operatorname{argmin}} \frac{1}{2} \sum_k \left\| \sum_m \mathbf{x}_{k,m} * \mathbf{d}_m - \mathbf{s}_k \right\|_2^2 + \sum_m \iota c_{PN}(\mathbf{g}_m) \quad \text{s.t. } \mathbf{d}_m - \mathbf{g}_m = 0 \; \forall m \tag{A9}$$

with ADMM iterations

$$\{\mathbf{d}_{m}\}^{(j+1)} = \underset{\{\mathbf{d}_{m}\}}{\operatorname{argmin}} \frac{1}{2} \sum_{k} \left\| \sum_{m} \mathbf{x}_{k,m} * \mathbf{d}_{m} - \mathbf{s}_{k} \right\|_{2}^{2} + \frac{\sigma}{2} \sum_{m} \left\| \mathbf{d}_{m} - \mathbf{g}_{m}^{(j)} + \mathbf{h}_{m}^{(j)} \right\|_{2}^{2}$$
(A10)

$$\{\mathbf{g}_{m}\}^{(j+1)} = \underset{\{\mathbf{g}_{m}\}}{\operatorname{argmin}} \frac{1}{2} \sum_{m} \iota c_{PN}(\mathbf{g}_{m}) + \frac{\sigma}{2} \sum_{m} \|\mathbf{d}_{m}^{(j+1)} - \mathbf{g}_{m} + \mathbf{h}_{m}^{(j)}\|_{2}^{2}$$
(A11)

$$\mathbf{h}_{m}^{(j+1)} = \mathbf{h}_{m}^{(j)} + \mathbf{d}_{m}^{(j+1)} - \mathbf{g}_{m}^{(j+1)}$$
(A12)

The problem with the $\{\mathbf{d}_m\}$ update is of the form

$$\underset{\{\mathbf{d}_m\}}{\operatorname{argmin}} \frac{1}{2} \sum_{k} \left\| \sum_{m} \mathbf{x}_{k,m} * \mathbf{d}_m - \mathbf{s}_k \right\|_2^3 + \frac{\sigma}{2} \sum_{m} \|\mathbf{d}_m - \mathbf{p}_m\|_2^2$$
(A13)

The variables $\mathbf{d}_m \mathbf{x}_{k,m}$, \mathbf{s}_k and \mathbf{p}_m are denoted in the DFT domain by $\hat{\mathbf{d}}_m$, $\hat{\mathbf{x}}_{k,m}$, $\hat{\mathbf{s}}_k$, and $\hat{\mathbf{p}}_{m}$, respectively, where $\hat{\mathbf{X}}_{k,m} = diag(\hat{\mathbf{x}}_{k,m})$, which becomes

$$\underset{\{\mathbf{d}_m\}}{\operatorname{argmin}} \frac{1}{2} \sum_{k} \left\| \sum_{m} \hat{\mathbf{X}}_{k,m} \hat{\mathbf{d}}_m - \hat{\mathbf{s}}_k \right\|_2^3 + \frac{\sigma}{2} \sum_{m} \left\| \hat{\mathbf{d}}_m - \hat{\mathbf{p}}_m \right\|_2^2$$
(A14)

13.

Defining

$$\hat{\mathbf{X}}_{k} = (\hat{\mathbf{X}}_{k,0} \quad \hat{\mathbf{X}}_{k,1} \quad \dots)\hat{\mathbf{d}} = \begin{pmatrix} \hat{\mathbf{d}}_{0} \\ \hat{\mathbf{d}}_{1} \\ \vdots \end{pmatrix} \hat{\mathbf{p}} = \begin{pmatrix} \hat{\mathbf{p}}_{0} \\ \hat{\mathbf{p}}_{1} \\ \vdots \end{pmatrix}$$
(A15)

this problem can be expressed as

$$\underset{\hat{\mathbf{d}}}{\operatorname{argmin}} \frac{1}{2} \sum_{k} \|\hat{\mathbf{X}}_{k} \hat{\mathbf{d}} - \hat{\mathbf{s}}_{k}\|_{2}^{2} + \frac{\sigma}{2} \|\hat{\mathbf{d}} - \hat{\mathbf{p}}\|_{2}^{2}$$
(A16)

with solution

$$\left(\sum_{k} \hat{\mathbf{X}}_{k}^{H} \hat{\mathbf{X}}_{k} + \sigma \mathbf{I}\right) \hat{\mathbf{d}} = \sum_{k} \hat{\mathbf{X}}_{k}^{H} \hat{\mathbf{s}}_{k} + \sigma \hat{\mathbf{p}}$$
(A17)

Equation (A11) is of the form

$$\underset{\mathbf{x}}{\operatorname{argmin}} \frac{1}{2} \|\mathbf{x} - \mathbf{y}\|_{2}^{2} + \iota c_{PN}(\mathbf{x}) = prox_{\iota c_{PN}}(\mathbf{y})$$
(A18)

It is clear from the geometry of the problem that

$$prox_{\iota c_{PN}}(\mathbf{y}) = \frac{PP^{T}\mathbf{y}}{\|PP^{T}\mathbf{y}\|_{2}}$$
(A19)

or, if the normalization $\|\mathbf{d}_m\|_2 \leq 1$ is desired instead

$$prox_{\iota c_{PN}}(\mathbf{y}) \begin{cases} PP^{T}\mathbf{y} & \text{if } \|PP^{T}\mathbf{y}\|_{2} \leq 1\\ \frac{PP^{T}\mathbf{y}}{\|PP^{T}\mathbf{y}\|_{2}} & \text{if } \|PP^{T}\mathbf{y}\|_{2} \geq 1 \end{cases}$$
(A20)

References

- Zhu, W.; Chen, X.; Wang, Y.; Wang, L. Arrhythmia recognition and classification using ECG morphology and segment feature 1. analysis. IEEE/ACM Trans. Comput. Biol. Bioinform. 2019, 16, 131–138. [CrossRef]
- Jekova, I.; Bortolan, G.; Christov, I. Assessment and comparison of different methods for heartbeat classification. Med. Eng. Phys. 2. 2008, 30, 248-257. [CrossRef]
- 3. Alfaras, M.; Soriano, M.C.; Ortin, S. A fast machine learning model for ECG-Based heartbeat classification and arrhythmia detection. Front. Phys. 2019, 7, 103. [CrossRef]
- Marinho, L.B.; Nascimento, N.D.M.M.; Souza, J.W.M.; Gurgel, M.V. A novel electrocardiogram feature extraction approach for 4. cardiac arrhythmia classification. Future Gener. Comput. Syst. 2019, 97, 564-577. [CrossRef]
- 5. Plawiak, P. Novel methodology of cardiac health recognition based on ECG signals and evolutionary-neural system. Expert Syst. Appl. 2018, 92, 334-349. [CrossRef]
- Aziz, S.; Ahmed, S.; Alouini, M. ECG-based machine learning algorithms for heartbeat classification. Sci. Rep. 2021, 11, 18738. 6. [CrossRef]
- Martis, R.J.; Acharya, U.R.; Min, L.C. ECG beat classification using PCA, LDA, ICA and discrete wavelet transform. Biomed. 7. Signal Proces. 2013, 85, 437–448. [CrossRef]
- 8. Minhas, F.A.; Arif, M. Robust electrocardiogram (ECG) beat classification using discrete wavelet transform. Physiol. Meas. 2008, 29, 555-570. [CrossRef]
- 9. Martis, R.J.; Chakraborty, C.; Ray, A.K. An integrated ECG feature extraction scheme using PCA and wavelet transform. In Proceedings of the 2009 Annual IEEE India Conference, Ahmedabad, India, 18-20 December 2009; p. 422.
- Lin, C. Frequency-domain features for ECG beat discrimination using grey relational analysis-based classifier. Comput. Math. 10. Appl. 2008, 55, 680-690. [CrossRef]

- 11. Martis, R.J.; Acharya, U.R.; Mandana, K.M.; Ray, A.K. Application of principal component analysis to ECG signals for automated diagnosis of cardiac health. *Expert Syst. Appl.* **2012**, *39*, 11792–11800. [CrossRef]
- 12. Wan, M.; Lai, Z.; Yang, G.; Yang, Z. Local graph embedding based on maximum margin criterion via fuzzy set. *Fuzzy Set. Syst.* **2017**, *318*, 120–131. [CrossRef]
- Wan, M.; Chen, X.; Zhan, T.; Xu, C. Sparse Fuzzy Two-Dimensional Discriminant Local Preserving Projection (SF2DDLPP) for Robust Image Feature Extraction. *Inform. Sci.* 2021, 563, 1–15. [CrossRef]
- 14. Hammad, M.; Maher, A.; Wang, K.; Jiang, F. Detection of abnormal heart conditions based on characteristics of ECG signals. *Measurement* **2018**, 125, 634–644. [CrossRef]
- 15. Matta, S.C.; Sankari, Z.; Rihana, S. Heart rate variability analysis using neural network models for automatic detection of lifestyle activities. *Biomed. Signal Process* **2018**, *42*, 145–157. [CrossRef]
- 16. Heide, F.; Heidrich, W.; Wetzstein, G. Fast and flexible convolutional sparse coding. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 5135–5143.
- Papyan, V.; Romano, Y.; Sulam, J.; Elad, M. Convolutional dictionary learning via local processing. In Proceedings of the 16th IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 5306–5314.
- Liu, Y.; Chen, X.; Ward, R.K.; Wang, Z.J. Image fusion with convolutional sparse representation. *IEEE Signal Proc. Let.* 2016, 23, 1882–1886. [CrossRef]
- 19. Zeiler, M.D.; Taylor, G.W.; Fergus, R. Adaptive deconvolutional networks for mid and high level feature learning. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Washington, DC, USA, 6–13 November 2011; pp. 2018–2025.
- Sermanet, P.; Kavukcuoglu, K.; Chintala, S.; LeCun, Y. Pedestrian detection with unsupervised multistage feature learning. In Proceedings of the 26th IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Portland, OR, USA, 23–28 June 2013; pp. 3626–3633.
- Zhou, Y.; Chang, H.; Barner, K.; Spellman, P. Classification of histology sections via multispectral convolutional sparse coding. In Proceedings of the 27th IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Columbus, OH, USA, 23–28 June 2014; pp. 3081–3088.
- 22. Chen, B.; Li, J.; Ma, B.; Wei, G. Convolutional sparse coding classification model for image classification. In Proceedings of the IEEE International Conference on Image Processing ICIP, Phoenix, AZ, USA, 25–28 September 2016; pp. 1918–1922.
- Zhang, Q.; Li, B. Discriminative K-SVD for dictionary learning in face recognition. In Proceedings of the 23rd IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Portland, OR, USA, 23–28 June 2010; pp. 2691–2698.
- Yang, M.; Zhang, L.; Feng, X.; Zhang, D. Fisher discrimination dictionary learning for sparse representation. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Barcelona, Spain, 6–13 November 2011; pp. 543–550.
- 25. Jiang, Z.L.; Lin, Z.; Davis, L.S. Label consistent K-SVD: Learning a discriminative dictionary for recognition. *IEEE Trans. Pattern Anal.* **2013**, *35*, 2651–2664. [CrossRef] [PubMed]
- 26. Wohlberg, B. Efficient algorithms for convolutional sparse representations. IEEE T Image Process. 2016, 25, 301–315. [CrossRef]
- Boyd, S.; Parikh, N.; Chu, E.; Peleato, B. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Found. Trends Mach. Learn.* 2010, 3, 1–122. [CrossRef]
- Garcia-Cardona, C.; Wohlberg, B. Convolutional dictionary learning: A comparative review and new algorithms. *IEEE Trans. Comput. Imag.* 2018, 4, 366–381. [CrossRef]
- Zubair, S.; Yan, F.; Wang, W. Dictionary learning based sparse coefficients for audio classification with max and average pooling. Digit. Signal Process. 2013, 23, 960–970. [CrossRef]
- 30. Wang, J.; Yang, J.; Yu, K.; Lv, F. Locality-constrained linear coding for image classification. In Proceedings of the 23rd IEEE Conference on Computer Vision and Pattern Recognition (CVPR), San Francisco, CA, USA, 13–18 June 2010; pp. 3360–3367.
- 31. Moody, G.; Mark, R. The impact of the MIT-BIH arrhythmia database. IEEE Eng. Med. Biol. 2001, 20, 45–50. [CrossRef] [PubMed]
- ANSI/AAMI EC57:1998; Testing and Reporting Performance Results of Cardiac Rhythm and ST Segment Measurement Algorithms. Association for the Advancement of Medical Instrumentation: American National Standards Institute, Inc. (ANSI): Washington, DC, USA, 1998.
- Singh, B.N.; Tiwari, A.K. Optimal selection of wavelet basis function applied to ECG signal denoising. *Digit. Signal Process.* 2006, 16, 275–287. [CrossRef]
- 34. Liu, B.D.; Shen, B.; Gui, L. Face recognition using class Specific dictionary learning for sparse representation and collaborative representation. *Neurocomputing* **2016**, *204*, 198–210. [CrossRef]
- 35. Song, Y.; Liu, Y.; Gao, Q.; Gao, X. Euler label consistent k-svd for image classification and action recognition. *Neurocomputing* **2018**, *310*, 277–286. [CrossRef]
- Shao, S.; Xu, R.; Liu, W.; Liu, B. Label embedded dictionary learning for image classification. *Neurocomputing* 2020, 385, 122–131. [CrossRef]
- Desai, U.; Martis, R.J.; Nayak, C.G.; Sarika, K. Machine intelligent diagnosis of ECG for arrhythmia classification using DWT, ICA and SVM techniques. In Proceedings of the 12 IEEE International Conference on Elect Energy Env Communications Computer Control, New Delhi, India, 17–20 December 2015; p. 2015.
- Elhaj, F.A.; Salim, N.; Harris, A.R.; Swee, T.T. Arrhythmia recognition and classification using combined linear and nonlinear features of ECG signals. *Comput. Method Prog. Biomed.* 2016, 127, 52–63. [CrossRef]

- Acharya, U.R.; Oh, S.L.; Hagiwara, Y. A deep convolutional neural network model to classify heartbeats. *Comput. Biol. Med.* 2017, 89, 389–396. [CrossRef]
- 40. Mohammad, K.; Shayan, F.; Majid, S. ECG heartbeat classification: A deep transferable representation. In Proceedings of the International Conference Healthcare Informativa ICHI, New York, NY, USA, 4–7 June 2018; pp. 443–444.
- Yildirim, O.; Baloglu, U.B.; Tan, R. A new approach for arrhythmia classification using deep coded features and LSTM networks. Comput. Method Prog. Biomed. 2019, 176, 121–133. [CrossRef]
- 42. Romdhane, T.F.; Alhichri, H.; Ouni, R.; Atri, M. Electrocardiogram heartbeat classification based on a deep convolutional neural network and focal loss. *Comput. Biol. Med.* **2020**, *123*, 103866. [CrossRef]
- 43. Li, Z.; Zhou, D.; Wan, L.; Li, J. Heartbeat classification using deep residual convolutional neural network from 2-lead electrocardiogram. J. Electrocardiol. 2020, 58, 105–112. [CrossRef]