

Fingerprint Recognition Using Translation Invariant Scattering Network

Shervin Minaee and Yao Wang

ECE Department, New York University, USA.

ABSTRACT

Fingerprint recognition has drawn a lot of attention during the last few decades. Different features and algorithms have been used for fingerprint recognition in the past. In this paper, a powerful image representation called scattering transform/network is used for recognition. Scattering network is a convolutional network where its architecture and filters are predefined wavelet transforms. The first layer of scattering representation is similar to SIFT descriptors and the higher layers capture higher frequency content of the signal. After extracting the scattering features, their dimensionality is reduced by applying principal component analysis (PCA). In the end, multi-class SVM is used to perform template matching for the recognition task. The proposed algorithm in this paper is one of the first works which explores the application of deep architecture for fingerprint recognition. The proposed scheme is tested on a well-known fingerprint database and has shown promising results with the best accuracy rate of 98%.

1. INTRODUCTION

To make an application more secure and less accessible to undesired people, we need to be able to distinguish a person from the others. There are various ways to identify a person such as keys, passwords and cards. However, biometrics are the most secure options so far. They are virtually impossible to imitate by any other than the desired person himself. They can be divided into two categories: behavioral and physiological features. Behavioral features are those actions that a person can uniquely create or express, such as signatures, walking rhythms, and the physiological features are those characteristics that a person possesses, such as fingerprints and iris patterns. Many works revolve around recognition and categorization of such data including, but not limited to, fingerprints, faces, palmprints and iris patterns [1]-[5].

Fingerprint is perhaps one of the most popular biometrics. It has been used in various applications such as forensics, transaction authentication, etc [1]. Many of the algorithms proposed for fingerprint recognition are minutiae-based matching. The major minutiae features of fingerprint ridges are ridge ending, bifurcation, and short ridge. In many of these algorithms, minutiae are extracted from the test and input fingerprint images, and the number of corresponding minutiae pairings between these two images is used to verify

the test fingerprint image. In the case of low-quality fingerprint images where segmentation of the minutiae could be challenging, new foreground segmentation approaches can be used to extract the minutiae from fingerprints [6], [7]. These algorithms use robust regression and sparse decomposition framework and provide a good segmentation even for low-quality images. Besides minutiae based algorithms, there are also a lot of image representations and feature-based algorithms for fingerprint recognition. In [8], Park proposed a fingerprint recognition system based on SIFT features. They extract SIFT feature points in scale space and perform matching based on the texture information around the feature points using the SIFT operator. Among more recent works, in [9], Cappelli proposed a new representation based on 3D data structure built from minutiae distances and angles called Minutia Cylinder-Code (MCC). In [10], Zhao proposed to use pore matching approach toward fingerprint recognition. Also in [11], Zhao introduced an adaptive pore modeling for fingerprint recognition.

Many of the biometric recognition systems involve a lot of pre-processing steps which are specifically designed for that kind of data and the final performance largely depends on the quality of those steps. Most of the previous feature based algorithms, use a single-layer representation of the image which may not be able to extract a set of features which are discriminative enough, and some of them may not work on some of the datasets although they may work for others. Therefore it is interesting to design features which are suitable for broader category of images and datasets. One way to extract more powerful features which can work for a broader range of tasks and datasets is to use deep architectures or multi-layer representation instead of a single-layer representation. There have been a lot of efforts in recent years to design supervised or unsupervised feature representations which work pretty well over various datasets and problems. These image representations should be invariant with respect to intra-class variation in the data. Many researchers developed deep architecture which can achieve the above goals to some extent. Scattering transform/network is one such representation. Scattering network is a convolutional network in which the filters and architectures are predefined wavelet filters [12]. It can be designed such that it is invariant to a family of transformation and small deformations [13]. Scattering networks is successfully used for several image and audio classification tasks

[14] and achieved promising results in several benchmarks. In [15], the scattering transform is used for iris recognition and achieved a very high accuracy rate. Due to tremendous success of deep scattering network in achieving state-of-the-art results for several tasks, it is interesting to know how this representation performs for fingerprint recognition. Here we have studied the application of this transformation for fingerprint recognition. One advantage of deep scattering network is that the entire architecture is known in advance and it does not require any learning of the weights and one could get a very rich set of features by going up to two levels in this network. Therefore the proposed algorithm is very fast and can be implemented in electronic devices in conjunction with energy-efficient algorithms [17], [18]. After the scattering features are extracted, their dimensionality is reduced using PCA [19]. In the end, multi-class SVM is used to perform classification using PCA features. This representation can also be used for extracting features from MRI and other medical images [16]. This algorithm is tested on the well-known PolyU fingerprint database [30] and achieved very high accuracy rate. Four sample fingerprint images of this database are shown in Figure 1.

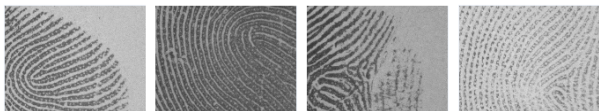


Fig. 1. Four different fingerprint images

The rest of this paper is organized as follows. Section 2 describes the features which are used in this work. The details of scattering transformation is provided in Section 2.1 and the PCA algorithm is explained in Section 2.2. Section 3 contains the explanation of the classification scheme. The results of our experiments and comparisons with other works are presented in Section 4 and the paper is concluded in Section 5.

2. FEATURES

Images of the same object could have variability due to translation, scale, rotation, illumination changes, etc. These changes in the images of a single object class are called intra-class variations which make object recognition very difficult in some scenarios. Therefore it is very important to design some image representations which are invariant under some of these intra-class variations. Various image descriptors have been proposed during past 20 years. SIFT and HOG are two popular hand-crafted image descriptors which achieved very good results on several object recognition tasks. Sparse representation has also been used for extracting features in image classification [20]-[22]. But some of these traditional descriptors are not very successful for some of the more challenging

datasets with many object classes and large intra-class variations. In the more recent works, deep neural network and also dictionary learning approaches have achieved state-of-the-art results on various datasets, most notably Alex-net [23] which is trained on ImageNet competition. In the deep learning framework, the images are fed as the input of the multi-layer neural network and the network discovers what is the best way to combine the pixels for maximizing the recognition accuracy. In the dictionary learning approach, different algorithms such as K-SVD or K-LDA are used to learn a set of features which are suitable for a given training set [24], [25]. In a recent work, a wavelet-based representation is proposed by Mallat [14], which is similar to deep convolutional network where instead of learning the filters and representation, it uses predefined wavelets [26]. These wavelets can be adopted such that they achieve some desired geometric invariance such as translation, rotation and scale invariance [12]. The details of scattering transform are described in the following section.

2.1. Scattering Features

The scattering operator is a deep convolutional network which uses wavelet transform as its filter and can be designed such that it is invariant under a group of transformations such as translation, rotation, etc [12]. The scattering transform computes local image descriptors with a cascade of three operations: wavelet decompositions, complex modulus and a local averaging. The scattering transform provides a multi-layer representation for a signal. As discussed in [14], some image descriptors such as SIFT can be obtained by averaging the amplitude of wavelet coefficients, calculated using directional wavelets. This averaging provides translation-invariance to some extent, but it also reduces the high-frequency information. Scattering transform is designed such that it recovers the high-frequency information lost by this averaging. It can be shown that the coefficients in the first layer of the scattering transform are similar to SIFT descriptors and the coefficients in the higher layers contain higher-frequency information of the image. We can get different versions of scattering transform by modifying it such that it is invariant under a new family of transformations. In this work, translation-invariant scattering transform is used and a brief description thereof is provided here.

Suppose we have a signal $f(x)$. The first scattering coefficient is the average of the signal and can be obtained by convolving the signal with an averaging filter ϕ_J as $f * \phi_J$. The scattering coefficients of the first layer can be obtained by applying wavelet transforms at different scales and orientations, removing the complex phase and taking their average by ϕ_J as shown below:

$$|f * \psi_{j_1, \lambda_1}| * \phi_J$$

where j_1 and λ_1 denote different scales and orientations. Tak-

ing the magnitude of the wavelet coefficients can be thought as the non-linear pooling functions used in convolutional neural networks. Note that by removing the complex phase of the wavelet we can make these coefficients invariant to local translation.

Now to recover the high-frequency contents of the signal, which are eliminated from the wavelet coefficients of the first layer by averaging, we can convolve $|f * \psi_{j_1, \lambda_1}|$ by another set of wavelets at scale $j_2 < J$ and take the average of their absolute values:

$$||f * \psi_{j_1, \lambda_1}| * \psi_{j_2, \lambda_2}| * \phi_J$$

It can be shown that $|f * \psi_{j_1, \lambda_1}| * \psi_{j_2, \lambda_2}$ is negligible for scales where $2^{j_1} \leq 2^{j_2}$. Therefore we only need to calculate the coefficients for $j_1 > j_2$.

The convolution with ϕ_J at the second layer removes high frequencies and results in locally translation-invariant second-order coefficients. This high-frequency information can be restored again by finer scale wavelet coefficients in the next layers. We can continue this procedure to obtain the coefficients of the k -th layer of scattering network as:

$$S_{k,J}(f(x)) = ||f * \psi_{j_1, \lambda_1}| * \dots * \psi_{j_k, \lambda_k}| * \phi_J$$

$$j_k < \dots < j_2 < j_1 < J, (\lambda_1, \dots, \lambda_k) \in \Gamma^k$$

It can be shown that the scattering vector of the k -th layer has a size of $p^k \binom{J}{k}$ where p denotes the number of different orientations and J denotes the number of scales. A scattering vector is formed as the concatenation of the coefficients of all layers up to m which has a size of $\sum_{k=0}^m p^k \binom{J}{k}$. For many signal processing applications, a scattering network with two or three layers will be enough. At the end, we can extract the mean and variance from each scattering transform image to form the scattering feature vector. One can also extract further information from each image to form the scattering feature vector.

The transformed images of the first and second layers of the scattering network for a sample fingerprint image are shown in Figures 2 and 3. These images are derived by applying bank of filters of 5 different scales and 6 orientations.

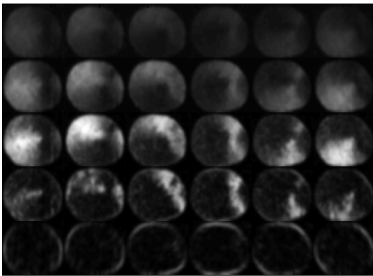


Fig. 2. The images from the first layer of scattering transform

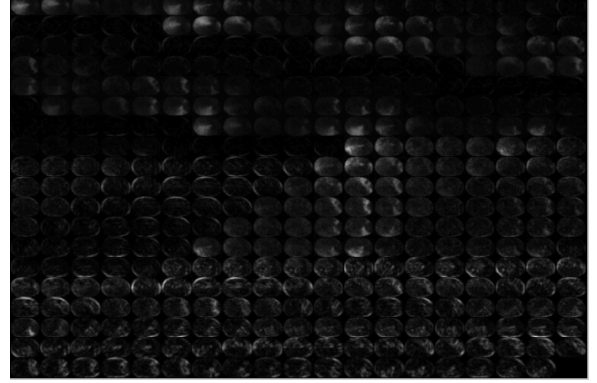


Fig. 3. The images from the second layer of scattering transform

2.2. Principal Component Analysis

In many applications, one needs to reduce the dimensionality of the data to make the algorithm faster and more efficient. Principal component analysis (PCA) is a powerful algorithm used for dimensionality reduction [19]. Given a set of correlated variables, PCA transforms them into another domain such that the transformed variables are linearly uncorrelated. This set of linearly uncorrelated variables are called principal components. PCA is usually defined in a way that the first principal component has the largest possible variance, the second one has the second largest variance and so on. Therefore after applying PCA, we could only keep a subset of principal components with the largest variance to reduce the dimensionality. PCA has a lot of applications in computer vision and neuroscience. Eigenface is one of the representative applications of PCA in computer vision, where PCA is used for face recognition.

Let us assume we have a dataset of M fingerprint images and $\{f_1, f_2, \dots, f_M\}$ denote their features where $f_i \in \mathbf{R}^d$. To apply PCA, all features need to be centered first by subtracting their mean: $z_i = f_i - \bar{f}$ where $\bar{f} = \frac{1}{M} \sum_{i=1}^M f_i$. Then the covariance matrix of the centered images is calculated as:

$$C = \sum_{i=1}^M z_i z_i^T$$

Next the eigenvalues λ_j and eigenvectors ν_j of the covariance matrix C are computed. Suppose λ_j 's are ordered based on their values. Then each z_i can be written as $z_i = \sum_{j=1}^d \alpha_j \nu_j$. The dimensionality of the data can be reduced by projecting them on the first K ($\ll d$) principal vectors as:

$$\hat{z}_i = (\nu_1^T z_i, \nu_2^T z_i, \dots, \nu_K^T z_i) = (\alpha_1, \dots, \alpha_K)$$

By keeping k principal components, the percentage of retained variance can be found as $\frac{\sum_{j=1}^k \lambda_j}{\sum_{j=1}^d \lambda_j}$. One way to choose k would be to pick a value such that the above ratio is less than ϵ , where ϵ is usually chosen between 90% to 99%.

3. RECOGNITION ALGORITHM: SUPPORT VECTOR MACHINES

After capturing the features of all people in the dataset, a classifier should be used to find the closest match of each test sample. There are various classifiers which can be used for this task including support vector machines (SVM) [27], majority voting algorithm and neural network. In this work, multi-class SVM has been used which is quite popular for image classification. A brief overview of SVM for binary classification is presented here. For further details and extensions to multi-class settings we refer the reader to [28]. Let us assume we want to separate the set of training data $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$ into two classes where $x_i \in \mathbf{R}^d$ is the feature vector and $y_i \in \{-1, +1\}$ is the class label. If we assume two classes are linearly separable with a hyperplane $w \cdot x + b = 0$ with no other prior knowledge about the data, then the optimal hyperplane is the one with the maximum margin. One can show that the maximum margin hyperplane can be found by the following optimization problem:

$$\begin{aligned} & \underset{w, b}{\text{minimize}} && \frac{1}{2} \|w\|^2 \\ & \text{subject to} && y_i(w \cdot x_i + b) \geq 1, \quad i = 1, \dots, n. \end{aligned} \quad (1)$$

Since this problem is convex, we can solve it by looking at the dual problem and introducing Lagrange multipliers α_i which results in the following classifier:

$$f(x) = \text{sign}\left(\sum_{i=1}^n \alpha_i y_i w \cdot x + b\right) \quad (2)$$

where α_i 's and b are calculated by the SVM learning algorithm. Interestingly, after solving the dual optimization problem, most of the α_i 's are zero; those datapoints x_i which have nonzero α_i are called support-vectors. There is also a soft-margin version of SVM which allows for mislabeled examples. If there exists no hyperplane that can split the "-1" and "+1" examples, the soft-margin method will choose a hyperplane that splits the examples as cleanly as possible, while still maximizing the distance to the nearest cleanly split examples [27]. It introduces a penalty term in the primal optimization problem which is the misclassification penalty (C) times the degree of

To derive the nonlinear classifier, one can map the data from the input space into a higher-dimensional feature space \mathcal{H} as $x \rightarrow \phi(x)$, so that the classes are linearly separable in the feature space [29]. If we assume there exists a kernel function where $k(x, y) = \phi(x) \cdot \phi(y)$, then we can use the kernel trick to construct nonlinear SVM by replacing the inner product $x \cdot y$ with $k(x, y)$ which results in the following classifier:

$$f_n(x) = \text{sign}\left(\sum_{i=1}^n \alpha_i y_i K(x, x_i) + b\right) \quad (3)$$

To perform multi-class SVM for a set of data with M classes, we can train M binary classifiers which can discriminate each class against all other classes, and to choose the class which classifies the test sample with the largest margin (one-vs-all). In another approach, we can train a set of $\binom{M}{2}$ binary classifiers each of which separates one class from another one and choose the class that is selected by the majority of classifiers. There are also some other approaches to perform SVM for multi-class classification problem.

4. EXPERIMENTAL RESULTS AND ANALYSIS

A detailed description of experimental results is presented in this section. First, let us describe the parameter values of our algorithm. For each image, scattering transform is applied up to two levels with a set of filter banks with 5 scales and 6 orientations, resulting in 391 transformed images. For each image, the mean and variance are calculated and used as features, resulting in 782 scattering features. For scattering transformation, we used the software implemented by Mallat's group [31]. Then PCA is applied to all features and the first 200 PCA features are used for recognition. Multi-class SVM is used for the template matching. For SVM, we have used the LIBSVM library [32], and linear kernel is used with the penalty cost $C = 1$.

We have tested our algorithm on the PolyU fingerprint database which is provided by Hong Kong Polytechnic University. It contains 1480 images of 148 fingers. The images of 25 people are used as a validation set for parameter tuning of our algorithm. From the remaining fingers, half of the images are used for training and the other half for testing. To make feature extraction faster, we have resized all images to 80×60 .

Figure 4 shows the recognition rate of the proposed approach for different numbers of PCA features. Interestingly, even by using a few PCA features, we are able to get a very high accuracy rate. As it can be seen, using 200 PCA features results in an accuracy rate around 98%, which will not increase much by using more PCA features.

The equal error rate (EER) of the proposed algorithm is also calculated on this dataset. Equal error rate is a rate at which both acceptance and rejection errors are equal. To find EER we have used the minimum distance classifier. Figure 5 shows the false acceptance rate and false rejection rate versus the distance threshold. As we can see, an EER of 8% is obtained using the proposed algorithm.

Table 1 shows a comparison between the EER of the proposed scheme and those of some of the recent works on this dataset. The proposed approach achieved a smaller EER compared to the other approaches, but there is still a great room for the improvement of EER on this dataset. That shows that the translation-invariant scattering network is more suitable for fingerprint identification than verification. The EER can be further improved by using rotation-translation invariant scat-

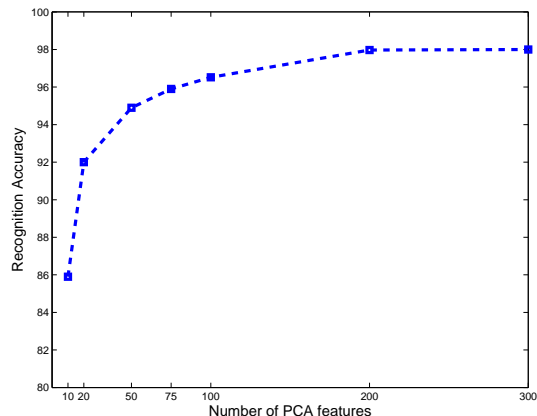


Fig. 4. Recognition accuracy as a function of number of PCA features

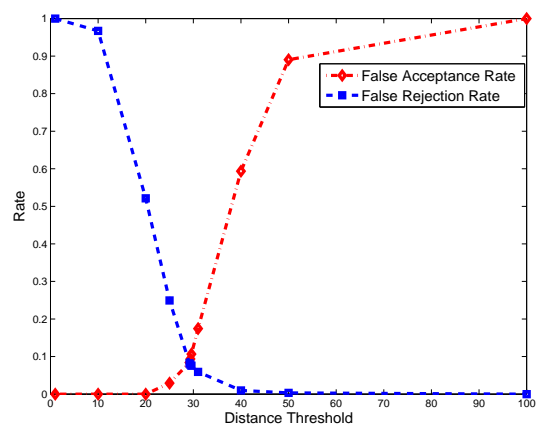


Fig. 5. FAR and FRR versus the distance threshold

tering features and also using more powerful classifiers than the minimum distance classifier.

The experiments are performed using MATLAB 2012 on a laptop with Core i5 CPU running at 2.6GHz. It takes around 97 milliseconds for each image to perform template matching using multi-class SVM.

5. CONCLUSION

This paper proposed to use a translation-invariant scattering network for fingerprint recognition. Scattering features are locally invariant and carry a lot of high-frequency information which are lost in other descriptors such as SIFT. This is one of the first works which explores the application of deep architecture for fingerprint recognition. Then PCA is applied on features to reduce their dimensionality. In the end, multi-class SVM is used to perform template matching. This shows the potential of scattering network for biometric recognition systems.

Table 1. A comparison between EER of the proposed scheme and previous approaches

Method	equal error rate
MICPP [10]	30.45%
Direct Pore Matching [10]	20.49%
Using Only Minutiae [10]	17.68%
Adaptive pore modeling [11]	11.51%
The proposed scheme	8.1%

Acknowledgments

The authors would like to thank Stephane Mallat’s research group at ENS for providing the software implementation of the scattering transform. We would also like to thank the CSIE group at NTU for providing LIBSVM software and the biometric research group at PolyU Hong Kong for providing the fingerprint dataset.

6. REFERENCES

- [1] D. Maltoni, D. Maio, A.K. Jain and S. Prabhakar, “Handbook of fingerprint recognition,” Springer Science and Business Media, 2009.
- [2] W. Zhao, R. Chellappa, P.J. Phillips and A. Rosenfeld, “Face recognition: A literature survey,” *ACM Computing Surveys (CSUR)* 35, no. 4: 399-458, 2003.
- [3] S. Minaee, A.A. Abdolrashidi, “Multispectral palmprint recognition using textural features,” *IEEE Signal Processing in Medicine and Biology Symposium*, 2014.
- [4] S. Minaee, A.A. Abdolrashidi, “Highly Accurate Multispectral Palmprint Recognition Using Statistical and Wavelet Features,” *IEEE Signal Processing Workshop*, 2015.
- [5] K.W. Bowyer, K.P. Hollingsworth and P.J. Flynn, “A survey of iris biometrics research: 2008-2010,” *Handbook of iris recognition*. Springer London, 15-54, 2013.
- [6] S. Minaee and Y. Wang. “Screen Content Image Segmentation Using Least Absolute Deviation Fitting,” *ICIP, IEEE*, 2015.
- [7] S. Minaee, A. Abdolrashidi and Y. Wang, “Screen Content Image Segmentation Using Sparse-Smooth Decomposition”, *49th Asilomar Conference on Signals, Systems, and Computers*, IEEE, 2015.
- [8] U. Park, S. Pankanti, A.K. Jain, “Fingerprint verification using SIFT features,” *SPIE Defense and Security Symposium*, 2008.
- [9] R. Cappelli, M. Ferrara, D. Maltoni, “Minutia cylinder-code: A new representation and matching technique for fingerprint recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2010.
- [10] Q. Zhao, L. Zhang, D. Zhang, N. Luo, “Direct pore matching for fingerprint recognition,” *In Advances in Biometrics*, pp. 597-606. Springer Berlin Heidelberg, 2009.

- [11] Q. Zhao, D. Zhang, L. Zhang, N. Luo, "Adaptive fingerprint pore modeling and extraction," *Pattern Recognition* 43, no. 8: 2833-2844, 2010.
- [12] L. Sifre and S. Mallat, "Rotation, scaling and deformation invariant scattering for texture discrimination," *IEEE Conference on Computer Vision and Pattern Recognition*, 2013.
- [13] S. Mallat, "Group invariant scattering," *Communications on Pure and Applied Mathematics*, 2012.
- [14] J. Bruna and S. Mallat, "Classification with scattering operators," *IEEE Conference on Computer Vision and Pattern Recognition*, pp.1561-1566, 2011.
- [15] S Minaee, AA Abdolrashidi and Y Wang, "Iris Recognition Using Scattering Transform and Textural Features," *IEEE Signal Processing Workshop*, 2015.
- [16] S Minaee, Y Wang and YW Lui, "Prediction of longterm outcome of neuropsychological tests of MTBI patients using imaging features," In *Signal Processing in Medicine and Biology Symposium (SPMB)*, IEEE, 2013.
- [17] M. Hosseini, A. Fedorova, J. Peters and S. Shirmohammadi, "Energy-aware adaptations in mobile 3D graphics," *ACM Multimedia*: 1017-1020, 2012.
- [18] M. Hosseini, J. Peters, S. Shirmohammadi, "Energy-budget-compliant adaptive 3D texture streaming in mobile games", *Proceedings of the 4th ACM Multimedia Systems Conference*, 2013.
- [19] H. Abdi and L.J. Williams, "Principal component analysis," *Wiley Interdisciplinary Reviews: Computational Statistics* 2.4: 433-459, 2010.
- [20] U Srinivas, H Mousavi, C Jeon, V Monga, A. Hattel and B. Jayarao, "SHIRC: A simultaneous sparsity model for histopathological image representation and classification", *ISBI, IEEE*, 2013.
- [21] M. Rahmani and G. Atia, "Randomized Subspace Learning Approach for High Dimensional Low Rank plus Sparse Matrix Decomposition", *49th Asilomar Conference on Signals, Systems, and Computers*, Nov 2015.
- [22] HS Mousavi, U Srinivas, V Monga, Y. Suo, M. Dao and TD. Tran, "Multi-task image classification via collaborative, hierarchical spike-and-slab priors", *International Conference on Image Processing*, IEEE, 2014.
- [23] A. Krizhevsky, I. Sutskever, GE. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, 2012.
- [24] Q. Zhang and B. Li, "Discriminative K-SVD for dictionary learning in face recognition," *IEEE Conference on Computer Vision and Pattern Recognition*, 2010.
- [25] J. Golmohammady, M. Joneidi, M. Sadeghi, M. Babaie-Zadeh and C. Jutten, "K-LDA: An algorithm for learning jointly over-complete and discriminative dictionaries," *Proceedings of the European in Signal Processing Conference*, IEEE, 2014.
- [26] J. Bruna and S. Mallat, "Invariant scattering convolution networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35.8: 1872-1886, 2013.
- [27] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning* 20.3: 273-297, 1995.
- [28] J. Weston, C. Watkins, "Multi-class support vector machines," *Technical Report CSD-TR-98-04*, Department of Computer Science, Royal Holloway, University of London, May, 1998.
- [29] B. Scholkopf, AJ. Smola, "Learning with kernels: Support vector machines, regularization, optimization, and beyond," MIT press, 2002.
- [30] <http://www.comp.polyu.edu.hk/biometrics/HRF/HRF.htm>
- [31] <http://www.di.ens.fr/data/software/scatnet/>
- [32] CC. Chang, CJ. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology (TIST)* 2.3: 27, 2011.