Supervised Representation Learning for Audio Scene Classification

A Rakotomamonjy

Supervised Representation Learning for Audio Scene Classification

A. Rakotomamonjy

***Abstract*—This paper investigates the use of supervised feature learning approaches for extracting relevant and discriminative features from acoustic scene recordings. Owing to the recent release of open datasets for acoustic scene classification (ASC) problems, representation learning techniques can now be envisioned for solving the problem of feature extraction. This paper makes a step towards this goal by first studying models based on convolutional neural networks (ConvNet). Because the scale of the datasets available is still small compared to those available in computer vision, we also introduce a technical contribution denoted as supervised non-negative matrix factorization (SNMF). Our goal through this SNMF is to induce the matrix decomposition to carry out discriminative information in addition to the usual generative ones. We achieve this objective by augmenting the NMF optimization problem with a novel loss function related to class labels of acoustic scenes. Our experiments show that despite the small-scale setting, supervised feature learning is favorably competitive compared to the current state-of-the-art features. We also point out that for smaller scale dataset, supervised NMF is indeed slightly less prone to overfitting than convolutional neural networks. While the performances of these learned features are interesting per se, a deeper analysis of their behavior in the acoustic scene problem context raises open and difficult questions that we believe, need to be addressed for further performance breakthroughs.**

***Index Terms*—time-frequency representation; audio scene classification; feature learning; non-negative matrix factorization; convolutional neural networks.**

I. Introduction

Audio scene classification (ASC) is a complex problem which aims at recognizing acoustic environments solely based on an audio recording of the scene. These acoustic scenes can be defined according to some geographical contexts (beach, park, road, etc...), some social situations in indoor or outdoor locations (restaurant, office, home, market, library, ..) or according to some transportation ground (car, bus, tramway, ...). Being able to accurately recognize such scenes is relevant for applications in which context awareness is of primary importance. Examples of relevant applications can be the monitoring of elderly patient routine in smart homes or the analysis of human activity for surveillance or tracking of traffic in an urban context.

AR is with Normandie Université, Université de Rouen, LITIS EA 4108, France. alain.rakoto@insa-rouen.fr

While recognizing these contexts can also be addressed through computer vision techniques, audio analysis has several advantages with respect to vision. Because of it is omnidirectional property, its robustness to occlusion and lighting condition, audio perception plays a major role in machine intelligence provided that it is coupled with a system capable of understanding an audio input related to sounds. Making a step towards to this goal is the objective of computational auditory scene analysis.

In the last decade, advances in the state-of-the-art in this domain were few but a steady increase in novel methodologies with improved performances occurred in the last few years. They have been essentially fueled by the release of open and established datasets for benchmarking. These datasets include the one used for the challenge DCASE 2013 [1] and the LITIS Rouen Audio scene dataset [2]. For the DCASE 2016 Challenge, a novel dataset for audio scene classification [3] has also been released for further fostering development of novel methodologies. This strong correlation with the release of open datasets and advances in the state of the art stresses again the need for the community to team up for making publicly available large and diverse datasets similar to those proposed in the computer vision community [4].

Since these datasets have been released, most works have focused on investigating discriminative representations for audio acoustic scenes. Most features that have been investigated for describing acoustic scenes are derived from related problems involving signal classifications. For instance, mel-frequency cepstral coefficients (MFCC) have been a widely used tool [5], [6], [7]. Features extracted from time-frequency decompositions based on matching pursuit have also been evaluated [8], [9]. Among hand-crafted features that have shown some successes in providing discriminative information about audio scenes, we can also mention recurrence quantitative analysis (RQA) [10] which aims at capturing some recurring patterns in MFCC representations. Since, in most cases, the first step when classifying acoustic

scenes is to compute a 2D time-frequency representation, some works have investigated features that are typically used in computer vision such as histogram of gradient (HOG) [2], [11], local binary pattern [12] or texture-based features [13], [14].

Most the above described features have been engineered based on on prior knowledge. One problem that may arise from hand-crafted features is their lack of adaptation to new datasets. In addition, they are usually designed to specifically model some aspects of the signals. For instance, the HOG features purposely model variation of energy in time-frequency representations. An alternative to hand-crafting features is the use of a feature learning strategy. In acoustic scene classification task, few works have considered a representation learning strategy for extracting features and most of them investigated unsupervised approaches. In this context, matrix factorization techniques have play an important role owing to theirs simplicity and effectiveness [6], [15], [16], [17]. For instance, Nam et al. [18] have employed an alternative approach based on Restricted Boltzmann machines which goal was to estimate the probability distribution of mel-frequency spectrogram.

One main rationale for discarding supervised learning approaches such a convolutional neural networks for acoustic scene classification is that these methods usually need a large amount of training examples in order to extract relevant features. They thus tend to overfit in a small-scale context. However, the DCASE 2016 Challenge Task 1 dataset is 10-fold larger than the 2013 one. Furthermore, we present in this paper a variant of the LITIS Rouen dataset that corrects several its flaws and that has about six thousands examples.

By leveraging on these novel datasets, this paper investigates supervised feature learning strategies for acoustic scene classification. The two methods we present are based on learning features from time-frequency representation of audio scene. The first one relies on matrix factorization techniques and the second one on convolutional neural networks (ConvNet)[19]. In this framework, our main technical contribution is the proposal of a supervised non-negative matrix factorization strategy. This supervision is achieved by augmenting the optimization problem in non-negative matrix factorization with a term that induces the factorization to be discriminative in some sense to be made clear latter. Several ways for solving the resulting optimization problem are discussed. Regarding ConvNet, the goal of our study is to evaluate the effectiveness of these models for acoustic scene classification and to investigate neural network architectures that are able to perform well despite the scale of the datasets, which is small compared to computer vision standard. Our experimental results show that the supervised learning approaches we propose are favorably competitive compared to hand-designed features

The paper is organized as follows. We present in Section II our technical contribution, denoted as supervised non-negative matrix factorization. Section III details our machine learning pipeline for solving the acoustic scene classification problem. In particular, we describe how the supervised NMF and the ConvNet models are used in this context. Experimental results are discussed in Section IV. We also point out in that section important open questions that arise from our results. Section V concludes the paper and opens up to future work.

As we advocate result reproducibility, all the codes, datasets and ConvNet models used for this work will be made publicly available on the authors website.

II. Supervised Non-Negative Matrix Factorization

This section describes in an abstract way our technical contribution related to supervised non-negative matrix factorization. We first start by introducing the model that allows us to take advantage of labels in a NMF context and then we discuss about algorithms that can be used for solving the resulting optimization problem.

A. Model

Suppose we have a set of L signals gathered in a matrix $\mathbf{S} = [\mathbf{s}_1, \mathbf{s}_2, \cdots, \mathbf{s}_L]$, with $\mathbf{S} \in \mathbb{R}^{N \times L}$. In many applications, one can suppose that these signals are generated by linear combination of few vector elements of $\mathbb{R}^N$. Based on this assumption, each signal $\mathbf{s}_i$ can be represented as $\mathbf{s}_i = \sum_{j=1}^{K} \mathbf{d}_j a_{j,i}$ where $\{\mathbf{d}_j\}_{j=1}^{K}$ are the generative elements, denoted in the sequel as dictionary elements or atoms, and $\{a_{j,i}\}$ the associated codes in the linear combination. In a matricized notation, this hypothesis translates into $\mathbf{S} \approx \mathbf{DA}$ where $\mathbf{D}$ and $\mathbf{A}$ are respectively a matrix of the form $\mathbb{R}^{N \times K}$ and $\mathbb{R}^{K \times L}$.

When the matrix $\mathbf{S}$ is essentially composed of non-negative elements, a relevant assumption is to also consider that matrices $\mathbf{D}$ and $\mathbf{A}$ have non-negative elements. Finding these latter matrices based on the knowledge of $\mathbf{S}$ is known as the non-negative matrix factorization (NMF) problem. We refer the reader to relevant works [20], [21], [22] for more details about NMF .

Basically, the goal of NMF is to find the factor matrices $\mathbf{D}$ and $\mathbf{A}$ that solve the following optimization problem.

$$\min_{\mathbf{D} \succeq 0, \mathbf{A} \succeq 0} \mathcal{D}(\mathbf{S}, \mathbf{DA}) \tag{1}$$

where $\mathcal{D}$ is a sum of element-wise divergence that measures discrepancy between each component of $\mathbf{s}_i$ and its

approximation $\sum_j \mathbf{d}_j a_{j,i}$. Typical divergence measures are the Euclidean, the Kullback-Leibler and the Itakura-Saito ones [23].

NMF aims at finding matrix factors which products represent at best the original matrix $\mathbf{S}$. Since the signals $\mathbf{s}_i$ to be decomposed are known to belong to some classes, our objective is to go beyond the generative factorization given by NMF, by inducing the code matrix $\mathbf{A}$ to bring some information about class labels in addition to reconstruction information.

First note that there has been several works on supervised dictionary learning that is related to the one we propose. Notably, Zhang et al [24] introduced an approach that jointly learns a representation and a classifier from the representation. Jiang et al. [25] considered a different approach in which they enforce the learned codes to carry discriminative information. Our work extends their methodology to non-negative matrix factorization providing rationale on why the resulting problem is still an NMF.

For our supervised NMF approach, similarly to Jiang et al. [25], we introduce a matrix $\mathbf{C}$ of size $K \times L$, K being the number of dictionary elements and L the number of elements to decompose (the columns of $\mathbf{S}$). The objective of $\mathbf{C}$ is to drive the coefficients in the matrix $\mathbf{A}$ to be aligned, in some sense to be defined, to class labels. We achieve this goal by considering that a given dictionary element should be preferrably used only for approximating signals of a given class.

This matrix $\mathbf{C}$ is built according to the following way. For a sake of clarity, suppose that K is a multiple of the number m of class, and that the $(c-1)\frac{K}{m}+1$ to $c\frac{K}{m}$ dictionary elements are related to class c. For all $i \in [(c-1)\frac{K}{m}+1, c\frac{K}{m}]$, we fix each entry $c_{i,j}$ of $\mathbf{C}$ so that $c_{i,j}=1$ if the signal $\mathbf{s}_j$ belongs to class c, meaning that the i-th dictionary element is somewhat "assigned" to that class. As an example, if we have a problem with 6 signals ordered in classes, 3 different classes and 3 dictionary elements to be learn, $\mathbf{C}$ writes

$$\mathbf{C} = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix}$$

The first dictionary element (row 1) is devoted to signals from the first class, whereas the second and thirdh dictionary elements are related to the second and third classes. Hence, $\mathbf{C}$ is a rank m matrix which bears class information owing to the assignment of a given dictionary element to a given class.

Since our objective is to induce codes in matrix $\mathbf{A}$ to bring both generative and discriminative information, the supervised NMF problem we want to solve is now

$$\min_{\mathbf{D}\succeq 0, \mathbf{A}\succeq 0, \mathbf{X}\succeq 0} \mathcal{D}(\mathbf{S}, \mathbf{DA}) + \lambda \mathcal{D}_d(\mathbf{C}, \mathbf{XA}) \qquad (2)$$

where $\mathbf{X}$ is a matrix of size $K \times K$ and $\mathcal{D}_d$ is another divergence measure. Note that the objective value of this optimization problem balances two terms weighted by $\lambda \geq 0$. The first term aims at reconstructing each signal as a positive combination of the dictionary elements. The second term goal is to make coefficients in the matrix $\mathbf{A}$ to be aligned with the label information brought by matrix $\mathbf{C}$.

Indeed, as described in the example above, we can note that if K is equal to m then, we are in the situation where a single dictionary element is "assigned" to one class. Suppose now that $\mathcal{D}_d$ is the Euclidean divergence, then the second term of Equation 2 can be written as $\|\mathbf{C} - \mathbf{XA}\|_F^2$. If we focus only on this term (without taking into account the generative aspect of NMF), we are actually looking for a rank-m factorization of the matrix $\mathbf{C}$. Since the basis elements of the column space of $\mathbf{C}$ are the canonical vectors of $\mathbb{R}^K$, the solution of this rank-m factorization is the matrix $\mathbf{X}$ composed of the canonical vectors in $\mathbb{R}^K$ and $\mathbf{A}$ is exactly the matrix $\mathbf{C}$:

$$\mathbf{C} = \underbrace{\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}}_{\mathbf{X}} \underbrace{\begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix}}_{\mathbf{A}}$$

As such, $\mathbf{A}$ carries label informations of each signal $\mathbf{s}_i$. This clarifies the compromise imposed by the objective function in Equation 2 between the alignment of $\mathbf{A}$ with the supervision matrix $\mathbf{C}$ and the generative information carried by $\mathbf{DA}$.

B. Algorithms

There exists a flurry of algorithms for solving NMF problems ($\lambda = 0$ in Equation 2). Depending on the divergence $\mathcal{D}$ considered, one can apply a multiplicative update strategy [26] which alternates between the optimization of $\mathbf{D}$ and $\mathbf{A}$. Lin [27] has also introduced a general projected gradient algorithm that applies to the Euclidean divergence. For more details about algorithmic development for NMF, interested readers can refer to [28].

Because the objective function in Equation (2) is notwell defined for the Kullback-Leibler or the Itakura-Saito divergences (for instance when $s_{i,j}=0$ or $(\mathbf{DA})_{i,j}=0$), we discuss in the sequel, about classical NMF algorithms that can be extended to our supervised NMF setting, when $\mathcal{D}$ and $\mathcal{D}_d$ are the Euclidean divergence.

Since we are considering Euclidean divergence, the objective function $f(\mathbf{A}, \mathbf{D}, \mathbf{X})$ can be written as :

$$\begin{aligned} f(\mathbf{A}, \mathbf{D}, \mathbf{X}) &= \frac{1}{2}\|\mathbf{S} - \mathbf{D}\mathbf{A}\|_F^2 + \frac{\lambda}{2}\|\mathbf{C} - \mathbf{X}\mathbf{A}\|_F^2 \\ &= \frac{1}{2}\|\tilde{\mathbf{S}} - \tilde{\mathbf{D}}\mathbf{A}\|_F^2 \end{aligned}$$

where $\tilde{\mathbf{S}} = \begin{pmatrix} \mathbf{S} \\ \sqrt{\lambda}\mathbf{C} \end{pmatrix}$ and $\tilde{\mathbf{D}} = \begin{pmatrix} \mathbf{D} \\ \sqrt{\lambda}\mathbf{X} \end{pmatrix}$. From this reformulation of the objective function, we show that in our case, our supervised NMF problem boils down to be a classical NMF but with augmented matrices. One simple way to solve (2) is to consider projected gradient approaches as described by Lin [27]. When the number of signals to be decomposed is very large, one possible method for solving Equation 2 is to employ an online matrix factorization algorithm such as the one proposed by Mairal et al. [29]. This last algorithm is the one we used in our experimental analysis for learning feature from mel-frequency representations.

III. Machine Learning Pipeline for ASC

We have introduced in the above section, a novel methodology for learning features based on a supervised non-negative matrix factorization. In what follows, we describe our machine learning pipeline for audio scene classification and show how the two supervised feature learning methods based on supervised NMF and ConvNet are used for solving this task.

A. Time-frequency representation of acoustic scenes

For learning features, we have at our disposal recordings related to acoustic scenes. Each of these recordings is associated to a class label describing the environment where it has been acquired. The first transformations we apply to each acoustic scene signal are the following

- the stereo signal is averaged over the two channels and normalized to unit energy.
- a log mel-frequency representation is obtained from this signal. The frequency span ranges from 0 to the half of the sampling rate. The number of spectral bands we considered is 70 and they are computed over windows of size 25 ms with hops of 10 ms. At this point, 15-s and 30-s length acoustic scenes can be represented as a matrix of size respectively 70×1495 and 70×2998. In the sequel, we will assume that we deal with matrix of size 70×2998.

Examples of log mel-frequency representation of different acoustic scenes are depicted in Figure 1. We can note that for these examples, classes are visually distinguishable and that the different acquisition process used for the two datasets have strong impacts on mel-frequency amplitudes.

B. Supervised NMF-based feature learning

Our objective in this part is to learn discriminative features from the time-frequency representation by leveraging on labeled examples. Suppose that we have n of these training examples each represented as a 70×2998 matrix. The idea behind supervised NMF, in this context, is to learn a decomposition of each time-slice and of course, the underlying weights of this decomposition should carry both generative and discriminative information.

1) Factorization: Hence, in our case, the matrix $\mathbf{S}$ we want to learn a factorization of is the matrix obtained from the concatenation of the mel-frequency representations of all signals, leading to a matrix of size $70 \times (2998 \times n)$, $\mathbf{D}$ is the matrix containing the discriminative dictionary elements and $\mathbf{A}$ is the code matrix allowing to reconstruct $\mathbf{S}$ from $\mathbf{D}$. The number K of dictionary elements is an hyperparameter which effect will be investigated in the experimental study. Because the number of vectors to decompose is of the order of 2.5 millions, we have used the online algorithm of Mairal et al. [29] for solving the problem in Equation (2).

For feature extraction purposes, once the dictionary $\mathbf{D}$ is learned, time-frequency representation of each acoustic scene is decomposed on the non-negative dictionary elements by proceeding slice per slice resulting in a matrix $\mathbf{A}$ of size $K \times 2998$ representing the acoustic scene over the dictionary. Note that this step does not require the label to be known and it is thus suitable for decomposing signals at the testing phase.

2) Pooling: The pooling step aims at creating a sketch of the matrix $\mathbf{A}$ by computing some statistics. These statistics are afterwards used as feature vector for a classifier. In our approach, these statistics are obtained through an integration over the temporal context of the acoustic scene. For instance, we have considered a sum pooling leading thus to a feature vector f_k of size K with :

$$f_k = \sum_{j=1}^{2998} a_{k,j} \quad \forall k \in 1, \cdots, K$$

We have also investigated a temporal maximum pooling

$$f_k = \max_j a_{k,j} \quad \forall k \in 1, \cdots, K$$

as well as the concatenation of a max pooling with a temporal average and standard deviation over $\mathbf{A}$, leading thus to a feature vector of size $3K$.

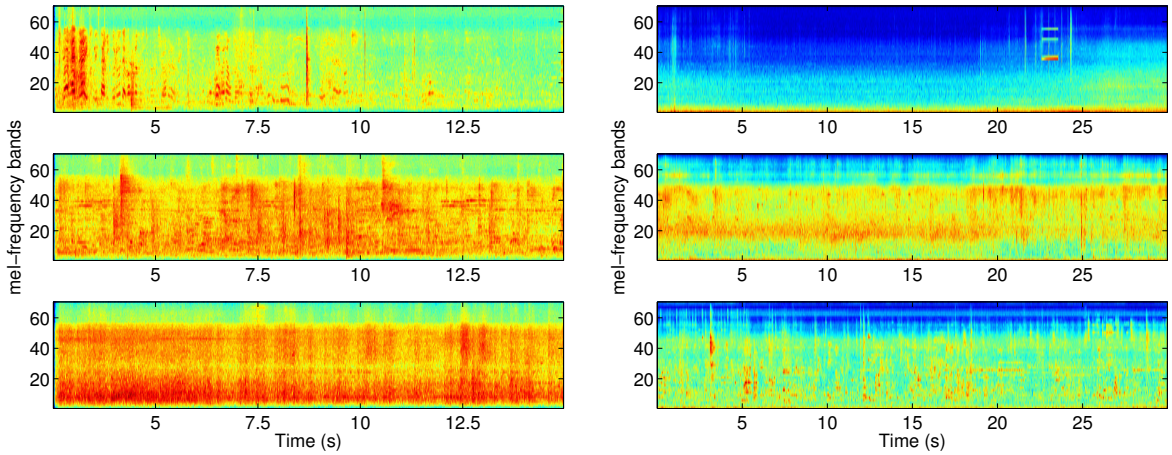

Fig. 1. Examples of mel-frequency representations of acoustic scenes. (left). From top to bottom, *bus*, *billiard pool hall* and *train* acoustic scenes from the LITIS Rouen-15 dataset. (right) *car*, *beach*, and *cafe* acoustic scenes from the DCASE16 dataset.

3) Classifier: After unit-norm normalization, these feature vectors extracted from the training examples are fed to a Gaussian kernel SVM classifier for learning a decision function. We used a *one-vs-one* multi-class strategy based the *LibSVM* software [30]. The C parameter of the SVM is selected among 8 values logarithmically scaled between 0.01 and 1000 while the parameter σ of the Gaussian kernel $k(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{\|\mathbf{x}-\mathbf{x}'\|_2^2}{2\sigma^2}\right)$ is chosen among the values $[0.5,\ 1,\ 5,\ 10,\ 20,\ 30,\ 50,\ 70,\ 100,\ 120]$. Depending on the dataset, model selection is performed according to a validation set or according to a cross-validation procedure.

C. ConvNet-based feature learning

The second approach we have investigated for supervised feature learning is the acclaimed and award-winning ConvNet model. In the next paragraphs, we describe how we have trained this model and the architecture we have explored.

Our ConvNet model is trained end-to-end from the mel-frequency representations to class posterior probabilities. Similarly to the supervised NMF approach, we fed to a ConvNet, a matrix of size 70×2998 as a single example from which the mean matrix over all training examples has been subtracted. We have investigated several ConvNet architectures using a trial-and-error approach. Typically, the best performing ones have two convolutional and one fully connected layers. We can note that these models are not very deep which can be explained by the fact that deeper models tend to overfit in our small-scale setting. Because of this context, we have also investigated the use of dropout [31] as a regularizing term. More details about these architectures will be provided in the experimental study.

Regarding the implementation details, the network architectures have been trained using stochastic mini-batch gradient descent based on back-propagation with momentum. Mini-batch size has been set to 5, while momentum to 0.9, the weight decay to 0.0005 and the learning rate to 10^{-4}. We have implemented these models using the Torch library [32]. The maximal number of epochs that has been used is 150 for models with small value of dropout and 300 for the other ones.

IV. Experimental results

This section presents the experimental study that we have carried out for evaluating the different supervised feature learning approaches we propose. We start by presenting the datasets we have considered and by describing the experimental setup. After presenting the baseline results obtained by state-of-the-art hand-crafted features, we analyze in details the performance obtained by our supervised non-negative matrix factorization method and as well as those obtained by the ConvNet architectures we have investigated.

A. Experimental settings

We describe in this subsection the datasets that have been used for carrying out our analysis. Specific experimental set-ups are also detailed

1) Datasets: Recently, Rakotomamonjy et al. [2] have introduced a publicly available dataset for acoustic scene classification. This dataset is one of the largest dataset available both in terms of number of classes and in terms of minutes of recordings. However, it presents an

TABLE I
SUMMARY OF DATASETS. FOR EACH DATASET AND FOLD, WE PROVIDE THE NUMBER OF EXAMPLES IN THE TRAINING, VALIDATION AND TEST SETS. WE ALSO GIVE THE MINIMAL AND MAXIMAL NUMBER OF EXAMPLES PER CLASS.

Rouen-15 : 19 classes				
fold	# examples	learning	validation	test
1	total	4241	574	1297
	min / class	36	4	9
	max / class	392	59	151
2	total	4316	537	1259
	min / class	35	3	11
	max / class	387	53	122
3	total	4275	597	1240
	min / class	38	6	5
	max / class	396	59	116
4	total	4323	629	1160
	min / class	35	4	10
	max / class	392	72	155
5	total	4287	538	1287
	min / class	36	3	10
	max / class	378	64	157

DCASE16 : 15 classes				
fold	# examples	learning	validation	test
1	total	880	290	0
	min / class	56	18	0
	max / class	60	22	0
2	total	880	290	0
	min / class	55	18	0
	max / class	60	23	0
3	total	872	298	0
	min / class	55	18	0
	max / class	60	23	0
4	total	878	292	0
	min / class	56	18	0
	max / class	60	22	0

TABLE II
ROUEN-15. COMPARING PERFORMANCE OF DIFFERENT BASELINE FEATURES BASED ON A *Average precision* CRITERION.

fold	RQA-MFCC	PSD	HOG	HOG+PSD
1	64.43	69.34	75.19	78.90
2	67.71	64.52	71.85	76.13
3	64.70	69.22	77.54	76.18
4	69.97	69.62	78.30	80.28
5	71.56	62.11	73.61	74.72
mean	67.68	66.96	75.30	**77.24**

important flaw that turns the results over-optimistic. The dataset suffers from the so-called "album effect" problem in music retrieval. Indeed, when splitting the recorded files into 30s examples, we did not pay attention that cuts from the same recording may fall into the training and test sets. This strong resemblance, at least due to background sounds, of some train and test examples make the classification problem easier to solve. This

TABLE III
DCASE16. COMPARING PERFORMANCE OF DIFFERENT BASELINE FEATURES BASED ON A *accuracy* CRITERION.

fold	RQA-MFCC	PSD	HOG	HOG+PSD
1	63.79	64.48	76.21	78.62
2	64.83	55.17	73.10	72.07
3	71.81	65.10	76.17	77.52
4	67.81	58.22	74.66	74.66
mean	67.06	60.74	75.04	**75.72**

last point is corroborated by one result of Bisot et al. [17] that show that a simple kernel PCA yields to features that are extremely competitive. In addition to this flaw, the current version of this dataset does not provide any validation set. Consequently, most results (including those we provide in [2]) are optimistic in the sense that they have been selected according to the best performing parameters on the test set. We use in this paper a corrected version of this dataset that we denoted as LITIS Rouen-15, which is based on examples of 15s long. In total, we have 6112 examples that have been separated into a training, validation and test sets. These splits have been performed five times and for each split, we were careful to have examples from same recording into the same set. Figure 1 displays 3 examples of acoustic scenes based on their time mel-frequency representations. Details on the classes and acquisition process can be found in [2].

The second dataset we have considered is the DCASE16 Task 1 development set. This dataset is composed of 1170 of 30s-length examples separated in a training and validation set. More details on this dataset can be found in [3]. Examples of time mel-frequency representations of these signals are depicted in Figure 1.

A quantitative summary of these datasets is presented in Table I. We present in there the repartition of the examples over the sets as well as the balance of classes. We can note that the DCASE16 is better balanced than the Rouen-15 dataset in which the largest class is 10 times more represented than the smallest one.

2) Evaluation set-ups: We have considered two different ways, according to the dataset for evaluating performance.

In the LITIS Rouen-15 dataset, since a validation set is available, model selection and classifier hyperparameter selection have to be selected based on this set. Hence, in all results we present in the sequel, the performances on the test set correspond to models which have performed the best on the validation set.

For the DCASE16 Task 1 dataset, since we do not have a validation set, model selection and hyperparameter

selection have been done *a posteriori* through cross-validation. This means that the results we present are the best averaged over the fold ones among all models and hyperparameters that have been evaluated.

Details on the different models of SNMF or ConvNet that have been evaluated will be given in the sequel. When features extracted from these models are fed to a Gaussian kernel SVM classifier, they are normalized so as to have zero mean and unit variance on the training set. The test set has also been normalized accordingly. As an evaluation criterion, because the number of examples per class are different in the LITIS Rouen-15 dataset, we have used the mean average precision as a criterion whereas accuracy is considered for the DCASE16 dataset.

B. Baseline results

Our first results depict how baseline features perform on the two dataset at hand. These features are recurrence quantitative analysis (RQA) extracted from MFCC, power spectral density and HOG extracted from mel-frequency representations. Results on LITIS Rouen-15 are shown in Table II. We note that MFCC coupled with a recurrence quantitative analysis and power spectral density [11] yield to similar performance of about 67% of average precision. Using histogram of gradient, a richer feature, improves performance up to 75%. Coupling HOG and power spectral density further boosts the mean average precision with a gain of 2%. This last point clarifies how PSD and HOG complement each other, the first one capturing how spectral energy is spread along frequency while the second one captures variation of spectral energy. This also confirms the results presented in [11].

Results in Table III for the DCASE16 dataset back up these findings. Although performance of PSD is far worse than those of other features and the fusion of HOG and PSD leads only to marginal gain in performance, we note the same trend in performances.

C. Supervised NMF

In this part, we discuss the performances of our supervised NMF approach. Note that in order to capture larger temporal patterns in the time-frequency representation, we have used this supervised NMF on consecutive frames. This is performed by simply concatenating these frames into a single vector and by applying NMF on these vectors. This operation is known as *shingling* [33].

Before discussing quantitative results, we present in Figure 2 a representation of the features that have been extracted from the validation set and obtained owing to the dictionary learned by SNMF as well as the best number of dictionary. In this figure, we present for both datasets, a 2D multidimensional scaling representation [34], denoted as *t-sne*, of the obtained features. As detailed above, the considered features are obtained by pooling using the best pooling function, the codes yielded by approximation of all the frames. The low-dimensional representation we achieve for the Rouen-15 dataset show that the features bring some discriminative information. Indeed, despite the fact that 900-dimensional feature vectors (best pooling is the concatenation of max, average and standard deviation and dictionary size 300) are projected into a 2-dimensional space, some cluster of classes are well-preserved. We can for instance note that examples from the *kid game hall* and *train* classes are well-clustered. At the contrary some other classes, like *restaurant* and *metro-paris* are spread. The feature vector projection for the DCASE16 dataset consolidates finding that some classes are well-clustered (*car*, *office*) while some are spread (*library* and *metro station*).

One interesting thing can also be noted from these plots : examples of some classes follow a multi-modal representation in this 2D projection (see for instance the *car* and *high-speed train* classes for Rouen-15 and the

TABLE IV
ROUEN-15. SUPERVISED NMF RESULTS WITH MODEL SELECTION PERFORMED ON POOLING, NUMBER OF DICTIONARY, SHINGLE SIZE AND CLASSIFIER PARAMETERS.

fold	pooling	#atoms	shingle	val	test
1	maxave	200	0	76.93	75.83
2	sum	400	1	77.18	72.09
3	sum	200	0	82.72	81.97
4	sum	400	0	84.06	81.14
5	sum	200	0	76.86	75.81
mean				**79.55**	**77.37**

TABLE V
DCASE16. SUPERVISED NMF: RESULTS WITH MODEL SELECTION ON CLASSIFIER PARAMETERS. (TOP) SHINGLE = 0 (BOTTOM) SHINGLE = 1.

dico	sum	max	maxave
50	62.27	39.25	63.79
100	60.77	30.74	59.22
200	72.98	69.24	76.89
300	76.20	75.55	**79.74**
400	71.11	62.75	71.88

dico	sum	max	maxave
50	68.45	50.44	71.45
100	57.42	32.48	57.85
200	65.05	22.29	64.52
300	75.38	54.18	73.77
400	76.93	65.63	77.26

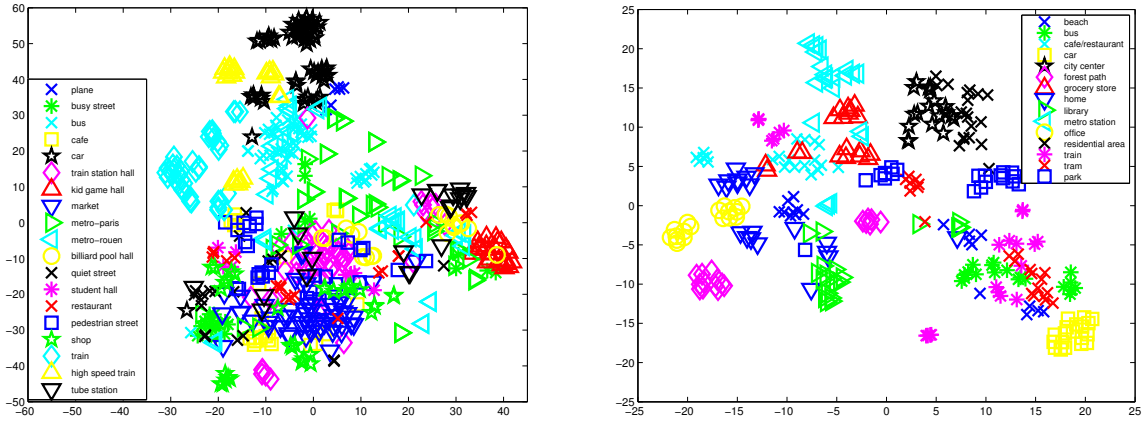

Fig. 2. Example of projection of features resulting from supervised NMF left) Rouen-15 dataset. right) DCASE16. For both datasets, the examples come from the validation set of the third fold.

forest path and *park* classes for DCASE16). This may suggest a certain inability of the learned features to be robust to strong variabilities present in recording of some classes.

Numerical results for the Rouen-15 dataset are presented in Table IV where for each fold, we present the best performing model, according to the validation set, with respects to the number of dictionary atoms, pooling, and shingling. We have made this last parameter, denoting the number of frames appended to the current one, varying from 0 to 2. The overall performance we achieve using these features is just slightly better than the one obtained with HOG+PSD. We remark that for some folds, improvements are significant while for some other (fold 2), the model clearly overfits.

Table V shows results for the DCASE16 dataset. For this problem, supervised NMF performs far better than HOG+PSD with an overall gain of 4%. Interestingly, we note that for both problems taking into account consecutive frames through *shingling* does not help in learning better features. One point that also needs to be highlighted is the important impact of the number of atoms in the dictionary and the pooling type on the performance. Indeed, details of the obtained results for Rouen-15 depicted in Table VI also support the evidence of strong variances in the results with performances ranging from 11% using max pooling to 81.9% using sum pooling.

D. ConvNet-based feature learning

We now present the results we obtain using convolutional neural networks for supervised feature learning.

As we have already stated, we have explored a large amount of models with varying number of convolutional layers, number of filters in these layers, size of kernels for the filters, type and size of pooling and type of non-linearity. For each fold, we have trained each of these models for 5 times with different initialization of the ConvNet weights. Our typical best performing architecture for both datasets are presented in Table VII. These architectures are of course not necessarily optimal and it is highly probable that models performing better according to the validation set can be found.

At first, we visualize how features learned by ConvNets carry discriminative information. Again, we have used the *t-sne* multi-dimensional scaling algorithm for projecting ConvNet features onto a 2D-dimensional space. These features have been obtained by removing the last fully-connected layer of one of the best performing architecture. Figure 3 depicts these 2D projection for the Rouen-15 and the DCASE16 dataset. For the Rouen-15 dataset, for some classes like the *car* or *high-speed train*, we can clearly note the benefits of ConvNets compared to supervised NMF. Their features are clearly better clustered. For the DCASE16 dataset, we remark that examples from same classes are well gathered together although some classes like *office* or *home* still present multi-modality aspects.

The model described in Table VII achieves a performance without dropout of about 77% which is similar to the performance of supervised NMF. We report in Table VIII, for each fold the best performing model according to the best initialization weights and the best value of $p \in [0.25, 0.375, 0.5, 0.6, 0.7, 0.8]$. Model selection is based on the average precision achieved on the validation set. We note in these results that the best performing model are those with larger probability of dropping out the weights. Owing to this regularization technique,

TABLE VI

ROUEN-15. SUPERVISED NMF RESULTS WITH MODEL SELECTION ON CLASSIFIER PARAMETERS. EACH TABLE CORRESPONDS TO A SINGLE TYPE OF POOLING AND DEPICTS TEST PERFORMANCE FOR DIFFERENT DICTIONARY SIZES AND DIFFERENT SIZE OF SHINGLE. RESULTS ARE FOR FOLD 3.

maxave

	Shringle		
#atoms	0	1	2
50	68.54	71.00	74.51
100	64.81	59.09	68.14
200	77.80	64.21	47.28
400	73.50	75.54	69.48

max

	Shringle		
#atoms	0	1	2
50	27.47	43.28	43.86
100	20.59	20.55	30.07
200	65.98	11.96	12.16
400	57.78	57.35	32.43

sum

	Shringle		
#atoms	0	1	2
50	68.12	71.46	72.45
100	64.13	60.35	67.57
200	**81.97**	61.64	36.66
400	67.12	77.24	73.42

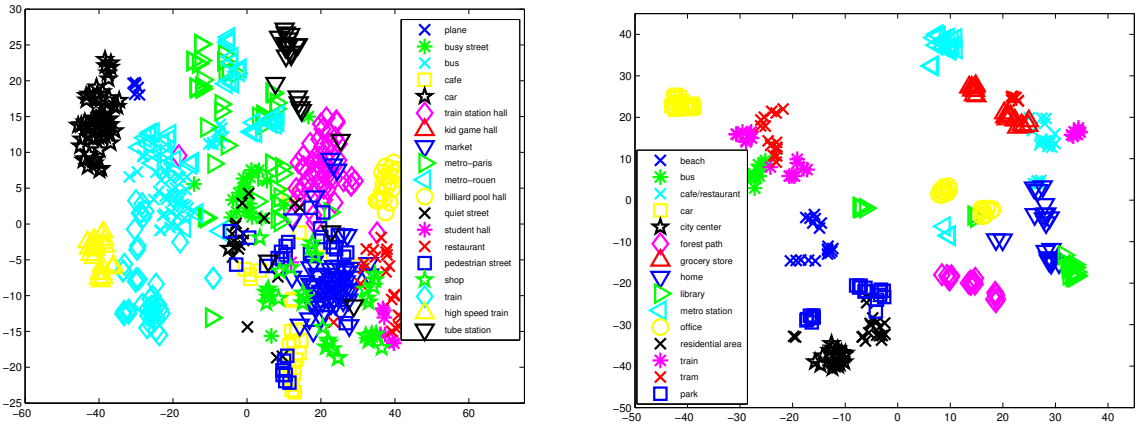

Fig. 3. Example of projection of features resulting from convolutional neural networks. left) Rouen-15 dataset. right) DCASE16. For both datasets, the examples come from the validation set of the third fold.

TABLE VII

TYPICAL ARCHITECTURE FOR OUR CONVNETS FOR BOTH DATASETS. THE PARAMETER p OF THE DROPOUT IS ADJUSTED ACCORDING TO A MODEL SELECTION PROCEDURE. FOR CONVOLUTIONAL LAYERS, THE FIRST NUMBER DEPICTS THE NUMBER OF FILTERS. PRODUCTS BETWEEN PARENTHESES DESCRIBES SIZE OF CONVOLUTIONAL KERNELS OR SIZE OF POOLING.

Type	Rouen-15	DCASE16
Input	70×1495	70×2998
Conv	256 - (5×21)	256 - (5×15)
ReLU		
MaxPool	(1×21)	(1×21)
Conv	512 - (3×11)	(3×7)
ReLU		
MaxPool	(1×21)	
FC	$512 \times 34 \times 4$	$512 \times 34 \times 4$
Dropout	p	p
FC	200	200
FC	19	15
Output	softmax	softmax

the average performance of the ConvNets now rise up to 79.5%, with smallest and largest performances over the fold reaching respectively 73% and 83%. In Table IX, we have depicted for each fold, the averaged-over-initialization, performance of the model that yields the best performance. We can note that for these models, variation of performances does not exceed 1.7%. It seems thus that ConvNets are more sensitive to variation due to folds that due to weight initializations.

Exploration of ConvNet architectures has also been carried out for the DCASE16 dataset. Performances of some of these models are reported in Table X. The models numbered from 1 to 8 are all models that do not use dropout. The best performing one, the sixth, is the one described in Table VII. The others are variants of this one. For instance, the first one uses larger convolutional kernels similar to those used for the Rouen-15 dataset, the second ones use 3 convolutional layers, etc... We can note that most of these models overfit as they reach their best performances on the validation set quite rapidly with respects to the maximal number (150) of epochs. Models numbered 9 to 11 correspond to model 6 for which we have applied dropout on the first fully-connected layer (models 9 to 11). Models 12 to 15 are variants of model 6 with dropout on the last convolutional layer and the first fully-connected layer (model 12 to 15). We can note the strong regularization effect of dropout yielding

TABLE VIII
ROUEN-15. CONVNET RESULTS. FOR EACH FOLD, WE REPORT THE RESULTS OF THE MODEL DESCRIBED IN TABLE VII THAT ACHIEVES THE BEST *average precision* ON THE VALIDATION SET, ACCORDING TO THE DROPOUT PARAMETER p. PERFORMANCE IS EVALUATED BASED ON AN *Average Precision* CRITERION.

Fold	Dropout p	init	validation set	test set
1	0.5	2	80.75	80.49
2	0.7	1	81.39	80.25
3	0.8	5	81.98	80.83
4	0.8	5	86.73	83.10
5	0.8	5	76.08	72.97
mean			**81.39**	**79.53**

TABLE IX
ROUEN-15. LOOKING AT THE VARIATION OF PERFORMANCE ACROSS INITIALIZATIONS.

Fold	Dropout p	validation set	test set
1	0.5	78.95	79.44 $\pm$ 1.50
2	0.7	80.80	78.90 $\pm$ 1.44
3	0.8	80.43	80.05 $\pm$ 1.69
4	0.8	85.67	81.43 $\pm$ 1.56
5	0.8	74.40	72.48 $\pm$ 1.00
mean		80.11	78.46

to a boost of performances. Dropout on the FC layer increase performance of 4%. With further dropout on the convolution layer, we achieve a global performance of 79.65%.

When comparing performances of the two supervised feature learning approaches, a slight advantage goes to ConvNets. Indeed, they perform about 2% better than supervised NMF on the Rouen-15 dataset while they are nearly on par on DCASE16.

E. Enriching learned features

Supervised matrix factorization followed by pooling or a convolutional neural networks are optimized to detect specific patterns in time-frequency representations of acoustic scenes. As such, they may lack in uncovering discriminative patterns that are not related to time-frequency structures.

Based on this rationale, we have considered enriching features extracted from supervised matrix factorization and ConvNets with other ones that have been recently deployed for acoustic scene classification problems. In a very basic way, we have computed histogram of gradient features on the time-frequency representations, power spectral density and recurrence quantitative analysis features and concatenated them to the supervised matrix factorization features or to the ConvNet features obtained by suppressing the last FC layer and the softmax. We have fed them to a classifier that has been trained following the same protocol that has been used for SNMF.

TABLE X
DCASE16. PERFORMANCE OF DIFFERENT CONVNET MODELS IN TERM OF ACCURACY. WE ALSO REPORT THE NUMBER OF ITERATIONS NEEDED FOR YIELDING THE BEST ACCURACY ON EACH FOLD'S VALIDATION SET. THE MOST-RIGHT NUMBER IN THE DROPOUT COLUMN DEPICTS THE PROBABILITY p OF DROPOUT FOR THE FC LAYER WHILE THE MOST-LEFT ONE IS THE ONE FOR CONVOLUTIONAL LAYER.

Model	Dropout	Accuracy	best models
1	-	73.85	19-39-39-51
2	-	68.72	73-64-46-45
3	-	69.41	67-47-40-77
4	-	68.27	83-78-86-75
5	-	74.27	29-21-27-17
6	-	74.61	36-33-58-33
7	-	73.16	42-23-24-32
8	-	73.00	27-47-50-55
9	0.7	76.65	83-63-52-47
10	0.3	77.68	49-52-152-177
11	0.9	78.78	164-156-152-150
12	0.5/0.7	78.45	62-119-88-96
13	0.5/0.8	**79.65**	138-171-277-202
14	0.5/0.9	78.29	159-237-244-175
15	0.8/0.8	77.61	156-199-143-177

TABLE XI
ENRICHING CONVNETS AND SNMF FEATURES WITH SOME HAND-CRAFTED ONES.

	Rouen15		DCASE 16
	Average Precision		Accuracy
Features	val. set	test set	cross-val
cnn	80.73	80.98	78.46
cnn+rqa	**81.94**	**81.79**	78.96
cnn+psd	79.00	78.00	73.05
cnn+hog	81.08	80.19	80.93
nmf	79.26	78.17	79.74
nmf+rqa	80.24	78.99	80.08
nmf+psd	73.57	72.56	75.72
nmf+hog	79.24	78.03	**81.19**
cnn+nmf	79.26	78.17	78.37
cnn+nmf+hog	79.24	78.03	80.93

Results for both datasets are presented in Table XI. We can note that some hand-crafted features are indeed good complement to SNMF and ConvNet features. For the Rouen-15 dataset, ConvNet models yield to a performance of 79.53%. Using them as features fed to an SVMs rises performance to almost 81%. This means that in some situations, these features benefit from a large-margin classifier instead of a softmax logistic regression. We can note that combination with RQA features further enhance performance with a maximum of 81.79% average precision on the test set. However, for this dataset, any combination with supervised NMF features does

not have any positive impact on performances. At the contrary, for the DCASE16 dataset, the combination of HOG and SNMF achieves performance of 81.19% which is our overall best, while combination of HOG and CNN reaches 80.98%. Interestingly, combining ConvNets and SNMF features yield to no gain for both datasets. As they both aim at capturing relevant discriminative patterns in a mel-frequency representation, it can be understood that they carry redundant information.

F. Open questions

The results we presented above show that supervised feature learning is highly competitive when it comes to unearth discriminative information in acoustic scenes. Proving this fact in a small-scale setting context is an important contribution of this work. While proposing novel feature is interesting per se, we believe that some of our results raise questions that need to be addressed for further improving audio scene classification methods.

1) Feature combination: According to the results we obtained by concatenating hand-crafted features to the learned ones, the question we want to ask is the following : does the need for combining features intrinsic to the ASC problem or it is relevant in this work due to lack of training data?

Indeed, on one hand, one may think that with a larger training set, most variabilities in the acoustic scenes will be well represented in training examples and thus can be captured by the learned features without the need for feature combination. On the other hand, it is also plausible that the patterns learned by the ConvNets or SNMF capture some specific characteristics of audio scenes but miss to highlight features related to recurrence pattern, low-energy but discriminative events etc...

To answer this question, we advocate again the release of larger and larger datasets by combining efforts of different groups while keeping developing features that complement those learned by ConvNets.

2) Mismatch between probability distributions: Results in Table VIII and IX for the Rouen-15 dataset suggest that variabilities in performances come more from distribution of the examples in the fold than from the random initialization associated to the ConvNet models. For the DCASE16 dataset, Figure 3 suggests that ConvNets are capable of learning discriminative features of acoustic scenes as most classes are well clustered. However, this visual inspection of the features does not translate in performance better than 80% of accuracy. These observations on both datasets support the conjecture that there exists a problem in the acoustic scene classification problem that goes beyond the question

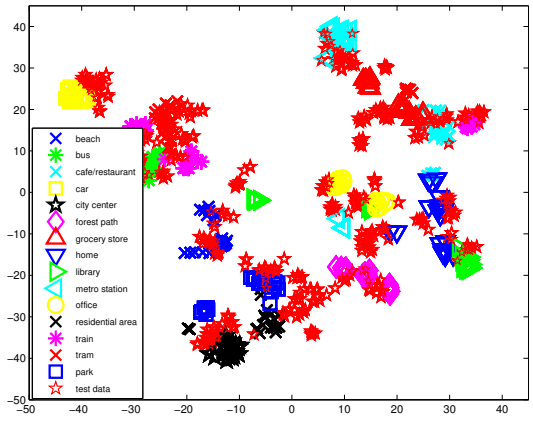

Fig. 4. Illustrating mismatch between validation set and test set probability distributions on the DCASE16 dataset. We have plotted in this Figure, the 2D t-sne projection of ConvNet features of validation and test set of fold 3.

of designing or learning discriminative features. And the problem is related to the probability distribution of training, validation and test examples. Figure 4 brings evidence for this strong statement of ours. As in the right panel of Figure 3, we have reproduced in there the 2D projection of the ConvNet features for the DCASE16 third fold validation set. In addition, we have also plotted the projection of the features from the test set. One important thing to note is that although classes are unknown, there is a clear mismatch between the distribution of the validation and test examples. Indeed, one would have expected a majority of the test examples to be located in regions of high-density of validation examples. Strong mismatches can for instance noted for *city center* or in-between *forest path* and *residential area*. Mismatch with probably smaller impact on performances can be highlighted for the *car* or *metro station*.

In summary, we clearly blame this mismatch in distribution for spoiling the discriminative power of ConvNets features (and probably other ones). However, we also believe that this mismatch is a natural but difficult challenge posed by acoustic scene classification problems. Indeed, it is understandable that due to different background sounds, different volumes of sound or due to the intrinsic variability of a given environmental or urban sounds, this mismatch occurs.

To address this mismatch problem, we thus have to design or learn features that are invariant or robust enough to these acoustic scene variabilities. This problem is also known in the machine learning community as the *domain adaptation* or *transfer learning* problem and we believe that it is possible and probably mandatory to take inspiration from related works [35], [36], [37] for further

progress in the acoustic scene classification problem.

V. Conclusion

We have investigated in this paper two methodologies for supervised feature learning for the acoustic scene classification problem. One of the approaches that we have explored is based on convolutional neural networks. The second one is based on a novel model that we have developed and it is a supervised extension of non-negative matrix factorization. We have evaluated the performance of these two approaches on two datasets : the DCASE16 acoustic scene classification problem and a corrected and enhanced version of our LITIS Rouen dataset.

We have carried out a large body of numerical analyses through which we have shown that the proposed supervised learning feature approaches are highly competitive even though the small-scale setting of the datasets. More interestingly, our results helped us pointing out two open questions that we believe need to be addressed for further breakthroughs. The *domain adaptation* problem is an important and difficult one and we plan to concentrate our future efforts in developing novel approaches able to circumvent this problem in the context of acoustic scene classification.

References

[1] D. Giannoulis, E. Benetos, D. Stowell, M. Rossignol, and M. Lagrange, "Detection and classification of acoustic scenes and events: an ieee aasp challenge," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 2013.

[2] A. Rakotomamonjy and G. Gasso, "Histogram of gradients of time-frequency representations for audio scene classification," *Audio, Speech, and Language Processing, IEEE/ACM Transactions on*, vol. 23, no. 1, pp. 142–153, Jan 2015.

[3] A. Mesaros, T. Heittola, and T. Virtanen, "TUT database for acoustic scene classification and sound event detection," in *24th European Signal Processing Conference 2016 (EUSIPCO 2016)*, Budapest, Hungary, 2016.

[4] J. Deng, W. Dong, R. Socher, L. J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, June 2009, pp. 248–255.

[5] L. Ma, B. Milner, and D. Smith, "Acoustic environment classification," *ACM Transactions on Speech and Language Processing*, vol. 3, 2006.

[6] B. Cauchi, "Non-negative matrix factorization applied to auditory scenes classification," Master's thesis, Master ATIAM, Université Pierre et Marie Curie, 2011.

[7] P. Hu, W. Liu, and W. Jiang, "Combining frame and segment based models for environmental sound classification," in *Proceedings of 13th Annual Conference of the International Speech Communication Association*, 2012.

[8] S. Chu, S. Narayan, and C. J. Kuo, "Environment sound recognition with time-frequency audio features," *IEEE Trans. on Audio, Speech and Language Processing*, vol. 17, no. 6, pp. 1142–1158, 2009.

[9] R. Mogi and H. Kasai, "Noise-robust environmental sound classification method based on combination of ica and mp features," *Artificial Intelligence Research*, vol. 2, no. 1, p. p107, 2012.

[10] G. Roma, W. Nogueira, and P. Herrera, "Recurrence quantification analysis features for environmental sound recognition," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 2013.

[11] V. Bisot, S. Essid, and G. Richard, "Hog and subband power distribution image features for acoustic scene classification," in *Signal Processing Conference (EUSIPCO), 2015 23rd European*. IEEE, 2015, pp. 719–723.

[12] H. Lim, M. J. Kim, and H. Kim, "Robust sound event classification using lbp-hog based bag-of-audio-words feature representation," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[13] J. Dennis, H. Tran, and E. Chng, "Image feature representation of the subband power distribution for robust sound event classification," *IEEE Trans on Audio, Speech and Language Processing*, vol. 21, no. 2, pp. 367–377, 2013.

[14] G. Yu and J. Slotine, "Audio classification from time-frequency texture," in *in Proceedings of IEEE International Conference in Acoustics, Speech and Signal Processing*, 2009.

[15] C. V. Cotton and D. P. Ellis, "Spectral vs. spectro-temporal features for acoustic event detection," in *Applications of Signal Processing to Audio and Acoustics (WASPAA), 2011 IEEE Workshop on*. IEEE, 2011, pp. 69–72.

[16] E. Benetos, M. Lagrange, and S. Dixon, "Characterisation of acoustic scenes using a temporally-constrained shift-invariant model," in *Proceedings of the fifteenth International Conference on Digital Audio effects*, 2012.

[17] V. Bisot, R. Serizel, S. Essid, *et al.*, "Acoustic scene classification with matrix factorization for unsupervised feature learning," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 6445–6449.

[18] K. Lee, Z. Hyung, and J. Nam, "Acoustic scene classification using sparse feature learning and event based pooling," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 2013.

[19] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[20] D. Lee and H. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, pp. 788–791, 1999.

[21] T. Hofmann, "Probabilistic latent semantic indexing," in *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 1999, pp. 50–57.

[22] P. Hoyer, "Non-negative matrix factorization with sparseness constraints," *Journal of Machine Learning Research*, vol. 5, pp. 1457–1469, 2004.

[23] C. Févotte and J. Idier, "Algorithms for nonnegative matrix factorization with the β-divergence," *Neural Computation*, vol. 23, no. 9, pp. 2421–2456, 2011.

[24] Q. Zhang and B. Li, "Discriminative k-svd for dictionary learning in face recognition," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2010, pp. 2691–2698.

[25] Z. Jiang, Z. Lin, and L. S. Davis, "Label consistent k-svd: Learning a discriminative dictionary for recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 11, pp. 2651–2664, 2013.

[26] D. Lee and H. Seung, "Algorithms for non-negative matrix factorization," in *Advances in Neural Information Processing Systems*, 2001, pp. 556–562.

[27] C.-J. Lin, "Projected gradient methods for nonnegative matrix factorization," *Neural computation*, vol. 19, no. 10, pp. 2756–2779, 2007.

[28] J. Kim, Y. He, and H. Park, "Algorithms for nonnegative matrix and tensor factorizations: a unified view based on block coordinate descent framework," *Journal of Global Optimization*, vol. 58, no. 2, pp. 285–319, 2014.

[29] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online learning for matrix factorization and sparse coding," *Journal of Machine Learning Research*, vol. 11, pp. 19–60, 2010.
[30] C.-C. Chang and C.-J. Lin, *LIBSVM: a library for support vector machines*, 2001, software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.
[31] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting." *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.
[32] R. Collobert, K. Kavukcuoglu, and C. Farabet, "Torch7: A matlab-like environment for machine learning," in *BigLearn, NIPS Workshop*, no. EPFL-CONF-192376, 2011.
[33] J. Salamon and J. P. Bello., "Unsupervised feature learning for urban sound classification," in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. Brisbane, Australia: IEEE Computer Society, 2015.
[34] L. Van der Maaten and G. Hinton, "Visualizing non-metric similarities in multiple maps," *Machine learning*, vol. 87, no. 1, pp. 33–55, 2012.
[35] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.
[36] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang, "Domain adaptation via transfer component analysis," *IEEE Transactions on Neural Networks*, vol. 22, no. 2, pp. 199–210, 2011.
[37] R. Gopalan, R. Li, and R. Chellappa, "Domain adaptation for object recognition: An unsupervised approach," in *2011 international conference on computer vision*. IEEE, 2011, pp. 999–1006.