Expert Systems With Applications 114 (2018) 334–344

Contents lists available at ScienceDirect

Expert Systems With Applications

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

Speech-music discrimination using deep visual feature extractors

Michalis Papakostas[a,b,*], Theodoros Giannakopoulos[a]

[a] Institute of Informatics and Telecommunications, National Center for Scientific Research "Demokritos", Athens 153 41, Greece
[b] Department of Computer Science, University of Texas at Arlington, Arlington, TX 760 19, USA

ARTICLE INFO

Article history:
Received 3 January 2018
Revised 11 May 2018
Accepted 12 May 2018
Available online 19 May 2018

Keywords:
CNNs
Speech-music discrimination
Transfer learning
Audio analysis

ABSTRACT

Speech music discrimination is a traditional task in audio analytics, useful for a wide range of applications, such as automatic speech recognition and radio broadcast monitoring, that focuses on segmenting audio streams and classifying each segment as either speech or music. In this paper we investigate the capabilities of Convolutional Neural Networks (CNNs) with regards to the speech - music discrimination task. Instead of representing the audio content using handcrafted audio features, as traditional methods do, we use deep structures to learn visual feature dependencies as they appear on the spectrogram domain (i.e. train a CNN using audio spectrograms as input images). The main contribution of our work focuses on the potentials of using pre-trained deep architectures along with transfer-learning to train robust audio classifiers for the particular task of speech music discrimination. We highlight the supremacy of the proposed methods, compared both to the typical audio-based and deep-learning methods that adopt handcrafted features, and we evaluate our system in terms of classification success and run-time execution. To our knowledge this is the first work that investigates CNNs for the task of speech music discrimination and the first that exploits transfer learning across very different domains for audio modeling using deep-learning in general. In particular, we fine-tune a deep architecture originally trained for the Imagenet classification task, using a relatively small amount of data (almost 80 min of training audio samples) along with data augmentation. We evaluate our system through extensive experimentation against three different datasets. Firstly we experiment on a real-world dataset of more than 10 h of uninterrupted radio broadcasts and secondly, for comparison purposes, we evaluate our best method on two publicly available datasets that were designed specifically for the task of speech-music discrimination. Our results indicate that CNNs can significantly outperform current state-of-the-art in terms of performance especially when transfer learning is applied, in all three test-datasets. All the discussed methods, along with the whole experimental setup and the respective datasets, are openly provided for reproduction and further experimentation.

Published by Elsevier Ltd.

1. Introduction

Speech-music discrimination, i.e. the task of segmenting an uninterrupted audio stream and classifying each segment as either speech or music, is very important in various audio analysis applications such as, radio broadcast monitoring, multimedia indexing, audio coding and automatic speech recognition. During the last two decades, several research efforts have tried to tackle the problem by proposing different technical approaches (El-Maleh, Klein, Petrucci, & Kabal, 2000; Pikrakis, Giannakopoulos, & Theodoridis, 2007; Saunders, 1996). However, speech-music discrimination still remains quite a challenging task for the community of audio analysis as most proposed systems fail to perform sufficiently under real-time scenarios, where the two classes overlap or alternate each other continuously.[1]

One of the earliest works in speech music discrimination (Panagiotakis & Tziritas, 2005) proposed a two-stage (segmentation, classification) speech-music discrimination approach, based on signal energy and the zero crossing rate. A combination of Hidden Markov models (HMMs) and multilayer perceptrons (MLPs) had been proposed by Ajmera, McCowan, and Bourlard (2003) using features from the cepstrum domain. In other researches (Jarina, 2001; Jarina, O'Connor, Marlow, & Murphy, 2002) , speech music discrimination was applied directly on the MPEG encoded bitstream, avoiding the computationally expensive decoding-encoding

* Corresponding author at: Department of Computer Science, University of Texas at Arlington, Arlington, TX 760 19, USA.
E-mail addresses: michalis.papakostas@mavs.uta.edu.com (M. Papakostas), tyiannak@gmail.com (T. Giannakopoulos).

https://doi.org/10.1016/j.eswa.2018.05.016
0957-4174/Published by Elsevier Ltd.

[1] https://github.com/MikeMpapa/CNNs-Speech-Music-Discrimination.

process. Dynamic programming and Bayesian Networks were adopted by Pikrakis et al. (2007) using typical short-term features. Kim et al. (2015) proposed ensembles of biased classifiers for automatically classifying the input signals into speech, singing, or instrumental categories, while empirical mode decomposition (EMD) was adopted by Khonglah, Sharma, and Prasanna (2015). Wu, Yan, Deng, and Wang (2010) discussed a method where a set of invariant audio features were extracted to perform Speech-Music classification by building a hierarchical structure of decision trees, and refining the classification result using a strategy for context-based state transform (ST). Sell and Clark (2014) deployed chroma features extracted from each audio sample and a GMM with a single component was trained on those features. The research performed by Didiot, Illina, Fohr, and Mella (2010) proposed a wavelet-based signal decomposition in a novel effort to tackle the problem. Finally Nilufar, Ray, Molla, and Hirose (2012) presented a multiple kernel learning approach in order to select the optimal sub-bands to discriminate the audio signals.

As with many other application domains, audio analysis has also been significantly benefited by the groundbreaking solutions that *deep learning* has offered to the machine learning community. Even though most research has focused on speech recognition (Dahl, Yu, Deng, & Acero, 2012; Hinton et al., 2012) with rather impressive results, there are still many areas under the umbrella of audio signal analysis that are still considered very challenging. The research discussed by Lee, Pham, Largman, and Ng (2009), proposed a pioneering approach on audio-feature extraction using convolutional deep belief networks (CDBNs) and illustrated results on 5 different application domains showing very promising results for the time, both in speech and music analysis. Scardapane, Comminiello, Scarpiniti, and Uncini (2013) applied an Extreme Learning Machine (ELM) for different tasks of music classification trained on traditional audio features. Their results were compared with a traditional feed-forward neural network showing superior performance on the task. The authors report results also on the task of speech-music discrimination, which is the target application of this paper, showing significantly inferior results compared to the ones illustrated in this work. Another paper that tackled the same problem is Pikrakis and Theodoridis (2014), where Restricted Boltzmann Machines (RBMs) were applied. RBMs highlight the great potentials of deep architectures on the task not only in terms of performance but also in terms of robustness and generalizability. Inspired by the outstanding results of convolutional neural networks (CNNs) in the domain of computer vision, initially illustrated by Krizhevsky, Sutskever, and Hinton (2012), there are a few research efforts that focus on tackling different audio analysis problems by representing the audio signal as a frequency image. In several other papers, such as Schlüter and Böck (2013a,b) and Grill and Schluter (2015), the authors used CNNs as the major component for the task of music onset detection, while Choi, Fazekas, and Sandler (2016),Zhang, Evangelopoulos, Voinea, Rosasco, and Poggio (2014) and Deng, Abdel-Hamid, and Yu (2013) exploit frequency images as input to deep learning structures (mainly CNNs) on the tasks of music classification and phone recognition. In a similar manner, Huang, Dong, Mao, and Zhan (2014) and Papakostas et al. (2017) deployed CNNs on frequency images for the task of emotion recognition from speech. In general, such deep-architectures and especially deep CNNs are well known for their ability to autonomously learn highly-invariant feature representations, extracted from complex images.

In this work, we have adopted CNNs on the task of speech - music discrimination. In contrast to all the previous works on the task, we exploit the highly invariant capabilities of CNNs on raw spectrogram images. The main novelty of our approach focuses on exploiting a deep pre-trained network (Donahue et al., 2015) and transfer learning for parameter tuning. We use a relatively small amount of training data (40 min of speech and 40 min of music training audio samples) to fine-tune the parameters of a deep-architecture initially trained on the Imagenet 1000-class image classification task. We extensively evaluate the potentials of CNNs on the task of speech music discrimination and we highlight the advantages in performance when transfer learning is applied. To our knowledge this is the first work that exploits CNNs on the task of speech music discrimination and also the first work that uses transfer-learning from a computer-vision to an audio modeling domain. Extensive experimentation and comparison to a wide range of traditional hand-crafted and deep-learning methodologies on audio features proves the supremacy of using CNNs with transfer learning on the task. Our results significantly outperform current state-of-the-art (Pikrakis et al., 2007; Pikrakis & Theodoridis, 2014) and show that CNNs can provide extremely robust solutions in the speech - music discrimination task.

2. CNNs for audio classification

Convolutional Neural Networks are probably the most popular modeling technique in computer vision related problems nowadays. Their ability to capture and represent robust and invariant features across millions of images has provided breakthrough results in some of the most traditional computer-vision problems such as activity or facial-expression recognition(Lopes, de Aguiar, De Souza, & Oliveira-Santos, 2017; Yang, Nguyen, San, Li, & Krishnaswamy, 2015). Despite their proven value in capturing features from multi-dimensional spaces, research that exploits CNN classifiers for non-vision problems and especially audio, has been very recently introduced and in a very limited amount of applications - mainly related to music classification or emotion recognition from speech (Lee, Park, Kim, & Nam, 2018; Papakostas et al., 2017). The aforementioned results highlight the great potentials of CNNs in modeling audio signals and indicate their potential superiority against traditional audio classifiers in several non-trivial tasks.

In our case we use CNNs as a classification method to classify raw spectrograms, with minimum data pre-processing into Speech or Music samples. Our approach is shown below in the form of pseudo-code. In the rest of the section we discuss in depth implementation details and we show how CNNs can be designed and exploited to capture audio related features for the problem of speech-music discrimination.

2.1. Training dataset and augmentation

All the evaluated methods have been trained using a set of pre-segmented audio samples each one belonging to any of the two classes (speech or music). In particular, the training data consists of 750 samples containing speech and 731 samples containing music. The average duration of a music sample equals 3.2 s while the total duration of all 750 music samples is 2428 s (40.5 min). On the other hand the average duration of a speech sample in our training set is 3.1 s and in total the duration of all 731 speech samples is 2237 s (37.3 min). All the samples from both classes were processed in a sampling frequency of 16000 Hz and where mono-channel audio samples. These speech and music segments have been gathered from several sources such as movies, youtube videos or radio-shows and have been manually annotated for the purposes of the work presented in Pikrakis et al. (2007).

Deep learning techniques, in most cases, require huge amounts of training data, in order to achieve satisfactory classification performance rates and avoid overfitting. In cases that the original data size is limited, data augmentation is required to overcome this data

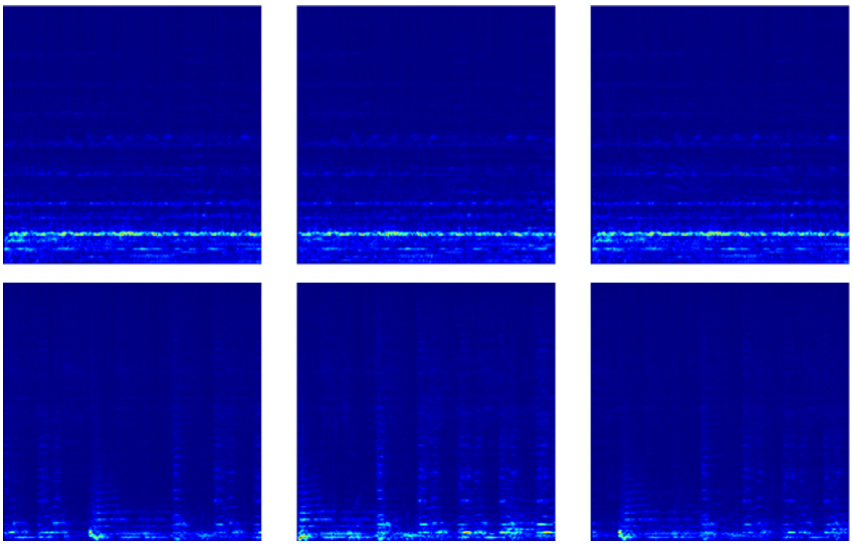


Fig. 1. Examples of part of the augmentation process for a music (upper row) and a speech (lower row) sample. The augmentation process generates 9 new spectrograms by adding background noise at three different levels and by applying 3 different crops (9 augmentation results in overall).

scarcity problem. Data augmentation is defined as a series of deformations applied on the annotated training samples which results in new additional training data (Simard, Steinkraus, & Platt, 2003). In most computer vision applications that utilize deep learning for classification, data augmentation is achieved through image deformations such as horizontally flipping, random crops and color jittering. In our case, before extracting the spectrogram of each training sample we add a random background sound (playing the role of noise) in three different Signal-To-Noise ratios (5, 4 and 3) and for three different crops of the original audio sample. If we also include the original (no noise) training sample, this means that this data augmentation procedure achieves a $3 \times 3 = 9$ dataset increase.

After the data augmentation we end up with 7500 music samples (750 original samples and 6750 new samples created after data augmentation) and 7262 speech samples (731 original samples and 6531 samples created after data augmentation). In total the duration of the augmented music class equals 12,587 s (210 min) while the average duration of a music sample is 1.7 s. Similarly the total duration of the augmented speech class is 11,161 s (186 min) while the average duration of a speech sample equals 1.5 s. Fig. 1 presents an example of two original (no noise) signals.

2.2. Audio segment representation

Each audio stream is broken to overlapping mid-term segments of 2.4 s length, while a 1 s step is used (i.e. almost 60% overlap). For each segment, the spectrogram is extracted, using 20 ms short-term window size and 15 ms step (25% overlap). This spectrogram is first interpolated, using linear-mapping, to match the target input of the CNN classifier and then is fed into the network for classification. In the rest of the paper we show two different ways of how CNNs can be adopted for the task of speech-music discrimination and we thoroughly discuss pros and cons of each different approach.

2.3. Using CNNs to classify audio segments

As recent literature has shown, deep hierarchical visual feature extractors can significantly outperform shallow classifiers trained on hand-crafted features and are more robust and generalizable when facing problems that include significant levels of inherent noise. To classify an unknown audio segment to either speech or music, we utilize two different CNN classifiers that differ primarily in their size.

Big CNN: The first one performs upon RGB-pseudo-colored frequency images, corresponding to the spectrograms of each audio segment, as described above. The color-map matches frequency values with a different color according to their intensity. Higher frequency values are mapped with brighter colors while lower frequencies with darker ones. The reason to do so was to be able to exploit the pre-trained CNN architecture for fine-tuning, which was originally designed to accept input images of three channels. The architecture of this deep CNN (Fig. 2) was initially proposed in Donahue et al. (2015). The model is mainly based on the Caffenet (Jia et al., 2014) reference model, which is similar to the original AlexNet (Krizhevsky et al., 2012)) and the network proposed in Zeiler and Fergus (2014). The network architecture consists of two convolution layers with stride of 2 and kernel sizes[2] equal to 7 and 5 respectively, followed by max pooling layers. Then a convolution layer with three filters of kernel size equal to 3 is applied, followed again by a max pooling layer. The next two layers of the network are fully connected layers with dropout, followed by a fully connected layer and a softmax classifier, that shapes the final probability distribution. All max pooling layers have kernel size equal to 3 and stride equal to 2. For all the layers we use the ReLu as our activation function.

The output of the network is a probability distribution on our target classes, while the output vector of the semifinal fully connected layer has size equal to 4096. Initial learning rate is set to 0.001 , and decreases after 700 iterations by a factor of 10. Since training from scratch such a big CNN structure as the one proposed by Donahue et al. (2015), requires millions of images thus, having very high computational demands, we used transfer learning to fine-tune the parameters of a pre-trained model. The original CNN was trained on the 1.2M images of the ILSVRC-2012 (Russakovsky et al., 2015) classification training subset of the ImageNet (Deng et al., 2009) dataset. Following this approach, we manage to decrease the required training time and to avoid over-

[2] By kernel size we refer to the size of each dimension of the kernel. All kernels are square matrices.

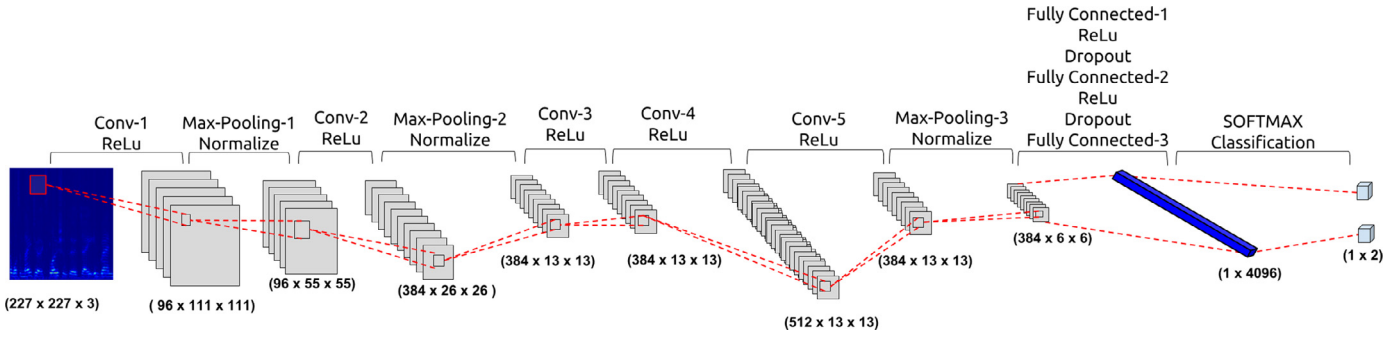


Fig. 2. Caffenet: CNN architecture proposed in Donahue et al. (2015).

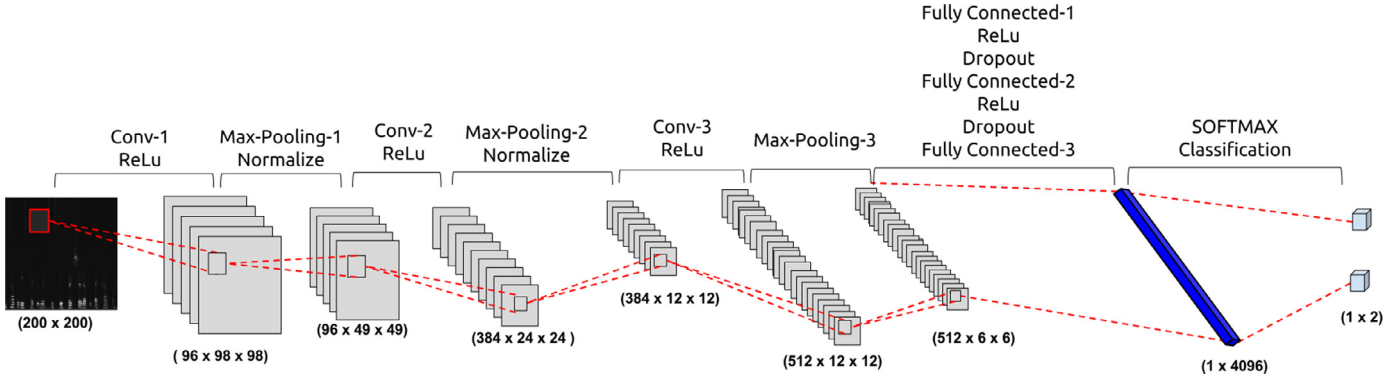


Fig. 3. CNN_SM: architecture proposed in this work.

fitting our classifier by ensuring a good weight initialization, given the relatively small amount of available training data. It is important to note that the network used for weight initialization was pre-trained on a dataset (*ImageNet*) completely irrelevant to our target data, proving the high invariance of CNN features and the importance of having a robust weight initialization. Finally, the input to the network corresponds to 227×227 RGB-pseudo-colored spectrogram images and their mirrors. Table 1, illustrates the effect of transfer-learning on the original kernels.

Small CNN: The second CNN structure (Fig. 3) is a smaller architecture that has been designed for the needs of the discussed problem. Despite the very good results provided by the first method, Caffenet was originally designed to tackle the Imagenet recognition task which is a problem significantly more complex than speech-music discrimination. In contrast to our target domain, Imagenet is a 1000-class image recognition task where many thousands of different features must be observed and captured by the classifier. However in our scenario, in all possible cases, the displayed patterns are much more simplistic compared to the Imagenet problem (Fig. 1). Traditional computer-vision issues such as background and lighting variations do not apply in our domain thus simplifying significantly our classification task. In our scenario noise mainly occurs by background sounds -which can be considered as visual occlusion in our case-, which in most cases (since they are in the background) do not affect much the dominant signal in the frequency-domain (mostly low frequencies are impacted). Moreover the original Caffenet was designed to function on raw RGB images; a source of information which is redundant for any audio-based classification problem. Spectrograms have by default two information channels and thus in order to make them fit in the original Caffenet we had to augment a third dimension using the pseudo-coloring approach described in the previous paragraph. Thus, in an effort to avoid unwanted computations and inspired by the proven value of Caffenet's architecture we decomposed the initial model to a smaller one in order to evaluate the ability of Deep Convolutional Classifiers to model the problem in general. Taking into account all the aforementioned observations and through extensive experimentation we ended up with a smaller architecture with a reduced number of convolutional layers. In total, the new network consists of two convolution layers less than the previous model with three consecutive pairs of convolution and pooling layers. The first convolution layer has a kernel size equal to 5 and the following two convolution layers have kernels with size equal to 3. All intermediate max-pooling layers have kernels equal to 3. Kernel stride in all layers is equal to 2. As activation the ReLu function is deployed once again. All fully-connected layers remain also untouched as well as learning rate and learning decay rate.

We evaluated two versions of this smaller CNN; one that operates again on 227×227 pseudo-colored RGB Images and one that operates on the default gray-scale spectrogram representation with smaller size equal to 200×200. For convenience reasons, we will refer to this model for now on as CN_SM. The grayscale version of CNN_SM is designed in an effort to reduce the computational complexity of the network by getting rid of some redundant computations including the extra information channel of the input but also some of the layers included in the original Caffenet. We reduce the image in an effort to investigate if the input size affects the final outcome. In both cases CNN_SM is trained from scratch in an end-to-end fashion. CNN_SM exploits on the one hand the core components of Caffenet while on the other hand has a reduced total number of parameters by a factor equal to 25% (almost 550.000 parameters less) compared to the total number of parameters that were on the initial Caffenet. As

Table 1
Convolution filters randomly picked from each layer. The first column illustrates the weights of the pre-trained Caffenet network before transfer-learning takes place. The second column displays the updated weights after fine-tuning. Lastly, third column displays how the values of each kernel have been shifted during the retraining process. *max* and *min* refer to the maximum and minimum weight values occurred in the two kernels, while *min_diff* refers to the mean shift value across all weights of each kernel.

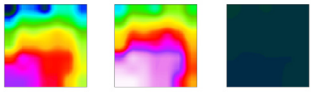
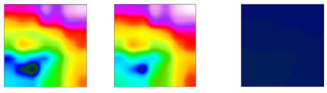
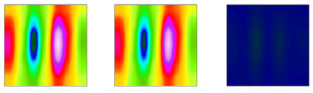
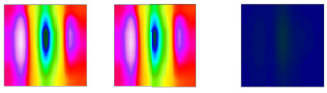
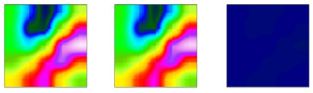
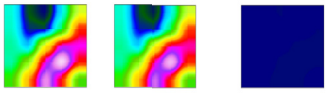
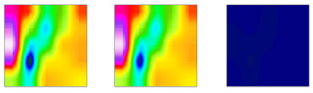
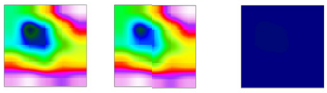
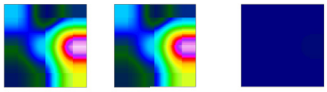
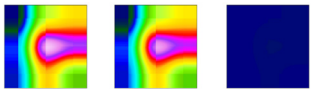
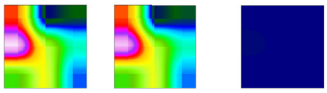
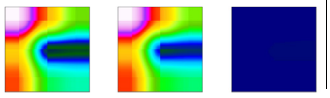
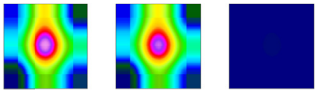
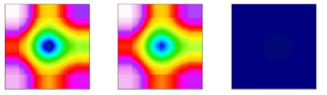
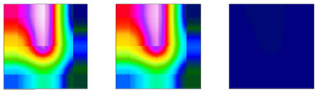
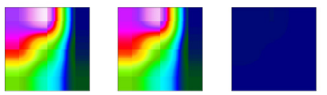
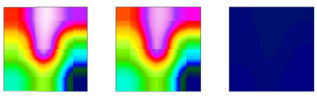
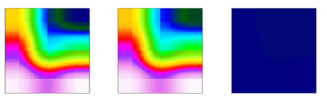
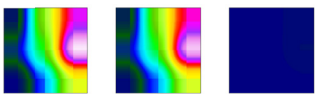
Colormap						
Layer	Initial	Fine-tuned	Difference	Initial	Fine-tuned	Difference
1st Convolutional Layer						
	max=0.0529, min=-0.0386, mean_diff=0.0155			max=0.0553, min=-0.1039, mean_diff=0.0074		
						
	max=0.3058, min=-0.3580, mean_diff=0.0046			max=0.2088, min=-0.3622, mean_diff=0.0042		
2nd Convolutional Layer						
	max=0.0727, min=-0.06132, mean_diff=0.0013			max=0.0534, min=-0.0457, mean_diff=0.0011		
						
	max=0.0534, min=-0.0870, mean_diff=0.0010			max=-0.0068, min=-0.0465, mean_diff=0.0009		
3rd Convolutional Layer						
	max=0.0100, min=-0.0053, mean_diff=0.0014			max=0.0663, min=-0132.0457, mean_diff=0.0007		
						
	max=0.0962, min=-0.0161, mean_diff=0.0016			max=0.0579, min=-0.0238, mean_diff=0.0008		
4th Convolutional Layer						
	max=0.0026, min=-0.0513, mean_diff=0.0011			max=0.0610, min=-0.0007, mean_diff=0.0007		
						
	max=-0.0072, min=-0.0657, mean_diff=0.0010			max=0.0629, min=-0.0217, mean_diff=0.0011		
5th Convolutional Layer						
	max=0.0631, min=-0.0059, mean_diff=0.0010			max=0.0776, min=-0.0089, mean_diff=0.0018		
						
	max=-0.0038, min=-0.0548, mean_diff=0.0010			max=0.0660, min=-0.0147, mean_diff=0.0009		

Table 2
Experimental results of the proposed method and comparisons to other methodologies, with and without post-processing on D1. We mainly focus on the average F1 measure as the final evaluation metric, due to its ability to be robust against unbalanced datasets, however, we report that the *overall classification accuracy* for the best two methods was: 96.6% for CNN_SM and 96.8% for the Caffenet-S, I, A method. For abbreviation purposes we define the following notations; Sp: Speech, Mu: Music, Rec: Recall, Prec: Precission, Av: Average.

No post-processing											
	Audio-based classifiers					Image-based classifiers					
	GMM	RF	GB	ET	SVM	SVM	Caffenet S	Caffenet S,I	Caffenet S,I,A	CNN_SM	CNN_SM A
Sp Rec	92.4	90.7	90.4	92.3	92.6	85.7	89.3	88.7	**93.7**	90.9	91.5
Sp Pre	79.5	82.7	82	80.9	77.4	83.6	89.4	**96.6**	93.3	91.1	90.8
Mu Rec	90.1	92.5	92.1	91.3	89.3	93.2	95.7	**98.8**	97.2	95.3	95.2
Mu Pre	95.7	96.2	96	96.8	96.8	94.2	95.7	95.7	**97.6**	96.2	95.8
Sp F1	85.5	86.5	86	86.2	84.3	84.6	89.3	92.5	**93.5**	91	91.1
Mu F1	92.8	94.3	94	94	92.9	93.7	95.7	97.2	**97.4**	95.7	95.5
Av F1	89.2	90.4	90	90.1	88.6	89.1	92.5	94.8	**95.4**	93.4	93.3
Post-processing segmentation											
	Audio-based classifiers					Image-based classifiers					
	GMM	RF	GB	ET	SVM	SVM	Caffenet S	Caffenet S,I	Caffenet S,I,A	CNN_SM	CNN_SM A
Sp Rec	92.4	90.3	90.3	92.3	92.8	85.8	89.7	88.9	**93.9**	92	92
Sp Pre	81.2	83.6	83.7	82	79.2	87.7	92.2	**97.3**	94.9	95.2	95.5
Mu Rec	90.8	93	93	92	90.7	95	97	**99**	98.1	98.2	98.4
Mu Pre	96.1	96	96	96.8	96.9	94.3	95.9	95.8	**97.6**	96.5	96.8
Sp F1	86.4	86.8	86.9	86.9	85.4	86.7	90.9	92.9	**94.4**	93.6	93.7
Mu F1	93.4	94.5	94.5	94.3	93.7	94.6	96.4	97.4	**97.8**	97.3	97.6
Av F1	89.9	90.7	90.7	90.6	89.6	90.7	93.7	95.1	**96.1**	95.5	95.6

our findings indicate in Section 3.4 CNN_SM can provide results that are slightly worse compared to the first method in a significantly shorter amount of time without the need of data augmentation. However using transfer-learning based on the pretrained architecture of Caffenet still remains the most accurate and robust method. In both scenarios though, it is obvious that CNNs can depict significant differences between the two classes even when the available data are limited showing state-of-the-art results on the task.

The input data-layer in both cases are in a batch form of 128 spectrogram images. At the end of each epoch[3] (about 16 iterations without data augmentation and 156 iterations when augmenting the initial training-set) we reshuffle the training data aiming to capture more robust feature representations.

2.4. Implementation details

For our experiments we used the *BVLC Caffe* (Jia et al., 2014) deep-learning framework and the system was evaluated on a Tesla K40c GPU, donated by NVIDIA. The code for this project is publicly available online.[4] In addition, the pyAudioAnalysis library (Giannakopoulos, 2015) for basic IO audio handling, segmentation and audio feature extraction was used.

3. Experimental evaluation

3.1. Evaluation datasets

In order to evaluate the performance of the proposed methodology against the traditional machine-learning methods shown in Table 2 a dataset (*D1*) of real recordings from several BBC radio broadcasts has been used. 33 separate uninterrupted radio streams of 10 min to 1 h length each have been manually annotated originally for the purposes of the research presented in Pikrakis et al. (2007). The total duration of the dataset is more than 10 h (almost 620 min).

In addition, we have evaluated our method on two additional open-access datasets for comparison purposes with the work done by Pikrakis and Theodoridis (2014), which also used deep-learning, and specifically RBMs, on traditional audio features (MFCCs and DFT coefficients).

Dataset *D2* originally appeared in Scheirer and Slaney (1997) and was subsequently refined in Williams and Ellis (1999). This corpus is a relatively small collection of 240 randomly chosen extracts from radio recordings. Each resulting file is 15 s long and stored in WAVE format. Original sampling frequency is 22050 Hz and all samples are single channel waves. For the purposes of this study we reduce the sampling frequency to 16000Hz in order to match our current implementation. The dataset is partitioned by its creators into a training subset and a test subset. However, in order to have a fair comparison against the methods proposed by Pikrakis and Theodoridis (2014) we ignored the initial data partitioning scheme. Our final D1 test dataset consists of two classes pure music (101 files) and pure speech (80 files with male, female and conversational speech).

Dataset *D3* is available via the Marsyas website (Tzanetakis & Cook, 2000). It consists of a total of 120 tracks, evenly distributed among the classes of music and speech. Each track is 30 s long and also stored in WAVE format. As in D2 we reduced the original sampling frequency from 22050 Hz to 16000 Hz. All audio samples are single channel wav files. The music class covers a wide variety of music genres and as in D2 some of the music samples are purely instrumental. The speech class contains both male and female speakers and in some cases dialogue.

3.2. Performance measures

Let *CM* be the confusion matrix, i.e. a 2×2 matrix, since 2 is the number of classes in our case. The rows and columns refer to the true (ground truth) and predicted class labels of the dataset, respectively. In other words, each element, $CM(i, j)$, stands for the number of samples of class i that were assigned to class j. The diagonal of the confusion matrix captures the correct classification decisions ($i = j$). The following three class-specific measures that

[3] A single epoch consists of a forward pass and a backward pass of all the training samples. The number of iterations is the number of passes, each pass using [batch size] number of examples. A pass consists of both the forward and back-word propagation.

[4] https://github.com/MikeMpapa/CNNs-Speech-Music-Discrimination.

describe how well the classification algorithm performs on each class have been extracted.

- the class *recall, Re*(i), which is defined as the proportion of data with true class label i that were correctly assigned to class i: $Re(i) = \frac{CM(i,i)}{\sum_{m=1}^{N_c} CM(i,m)}$, where $\sum_{m=1}^{N_c} CM(i,m)$ is the total number of samples that are known to belong to class i.
- the class *precision*, i.e. the fraction of samples that were correctly classified to class i if we take into account the total number of samples that were classified to that class: $Pr(i) = \frac{CM(i,i)}{\sum_{m=1}^{N_c} CM(m,i)}$.
- the F_1-measure which is the harmonic mean of the precision and recall values: $F_1(i) = \frac{2Re(i)Pr(i)}{Pr(i)+Re(i)}$

Note that, the confusion matrix, and therefore all adopted performance measures, has been extracted on a 1 s segment basis.

3.3. Results

The results of the proposed method, along with the compared methodologies on the D1 test dataset, are presented in Table 2. In general, two types of classifiers have been evaluated: (a) audio classifiers based on low-level audio features (b) classifiers applied on the spectrogram images. In particular, the following methods are evaluated (we also present the abbreviations in the following list):

1. **Audio-based classifiers**:
   (a) RF: Random Forests
   (b) GB: Gradient Boosting
   (c) ET: Extra Trees
   (d) SVM: Support vector machines
   (e) GMM+HMM: Gaussian Mixture Models + Hidden Markov Models
   In order to extract hand crafted audio features he have used the pyAudioAnalysis library (Giannakopoulos, 2015) which computes several time, spectral and cepstral domain audio features such as zero crossing rate, spectral centroid and MFCCs. Towards this end, a short-term windowing is applied, and for each short-term window (frame) 34 features are computed. Then for each segment two feature statistics are extracted, namely the mean and standard deviation, leading to a 34 * 2 = 68 feature statistic representation for each audio segment. This final representation is used as a feature vector to classify unknown audio segments to either speech or music. More details can be found at Giannakopoulos (2015).
2. **Image-based classifiers**. The following image classifiers have been directly applied on the spectrogram images for comparison reasons:
   (a) SVM: for comparison reasons we have also evaluated an image classifier applied on the spectrograms using typical visual features: *Histograms of Oriented Gradients, Local Binary Patterns, Grayscale and color histograms*. To train the SVM classifier each image was decomposed to a 4x4 grid and from each block (4x4=16 blocks in total) a feature vector was extracted. The final feature vector describing the whole image was the concatenation of each individual feature vector extracted from each of the 16 blocks.
   (b) Caffenet-S: this CNN uses the structure of the Imagenet CNN proposed by Donahue et al. (2015), but it is trained directly on the training samples of the speech - music discrimination task
   (c) Caffenet-S, I: this is the same network, however the weights of the Imagenet CNN are also used for initialization in the training phase of the speech-music classifier
   (d) Caffenet-S, I, A: this is the same network (with weight initialization), but training is performed using the augmented data of speech - music
   (e) CNN_SM: this is the smaller CNN proposed in Section 2.3 to discriminate speech and music segments, which is trained from scratch. For fair comparison against our Caffenet implementation the results shown in Table 2 are with CNN_SM functioning on 227×227 pseudo-colored RGB images. It has to be noted that no significant differences in performance were observed when altering the type and shape of input to 200x200 grayscale (see Section 3.4).
   (f) CNN_SM-A: this is the CNN of (e) trained using augmented data.

Additionally, the evaluation has been conducted with and without post processing of the single classification decisions. We have selected a simple but effective median filtering on the extracted classification labels as a post-processing method. We have also conducted experiments with supervised smoothing approaches (e.g. HMM), but no further improvement was observed. The presented results have been produced after the application of a median filter of a 11 s window.

The following conclusions are directly drawn from the results of Table 2:

- All CNN-based methods outperform both audio-based classifiers and the image SVM-classifier based on handcrafted features by 3% to 6% in Average F1.
- Caffenet-S, I, A and CNN_SM are the best classifiers. In particular, for the post-processing case, they achieve an average F1 measure almost equal to 96%. Additionally, experiments proved that the overall classification accuracy was 96.6% for CNN_SM and 96.8% for the Caffenet-S, I, A method.
- The original Caffenet architecture is better than the audio classifiers but almost 3% worse than our Caffenet-S, I approach (with and without augmentation). This means that, using Imagenet for weight initialization is equally important to using its structure. It is therefore obvious that adopting transfer learning to fine-tune the parameters of a pre-trained model from a totally different domain (image retrieval) leads to significant performance boosting.
- The simple post-processing step achieves a performance boosting from 0.5% to 1.5% in terms of average F1. A higher boosting is achieved for the CNN_SM method compared to the Caffenet-S, I, A approach. This obviously indicates that this method returns more misclassifications that are easily corrected by simple smoothing methods.
- CNN_SM is also slightly better than Caffenet-S, although both networks are learnt from scratch. This indicates that the network simplicity can, under certain circumstances, lead to better performance.
- Data augmentation does not always lead to better results: in the case of the best model (Caffenet-S, I, A), data augmentation adds 1% to the final F1 measure, however this is not the case for the CNN_SM network.

In Fig. 4 we demonstrate the average F1 measure of various experimental setups of the two CNN architectures, as the number of training iteration increases when evaluated on D1 test dataset. By taking a closer look at Fig. 4 we can educe the following immediate observations that complement the aforementioned conclusions:

- Using Imagenet as a weight initialization mechanism significantly improved both classification performance and convergence rate by a minimum of 2.1% (Caffenet-S vs Caffenet-S, I at 1500 iterations) to a maximum of 5.2% (Caffenet-S vs Caffenet-S, I at 500 iterations).

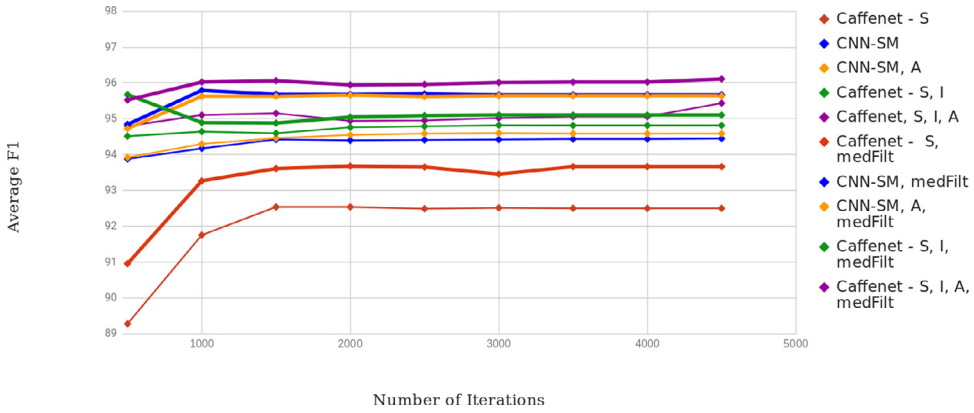


Fig. 4. Average F1 convergence of different experimental setups with and without post processing and for different number of training iteration.

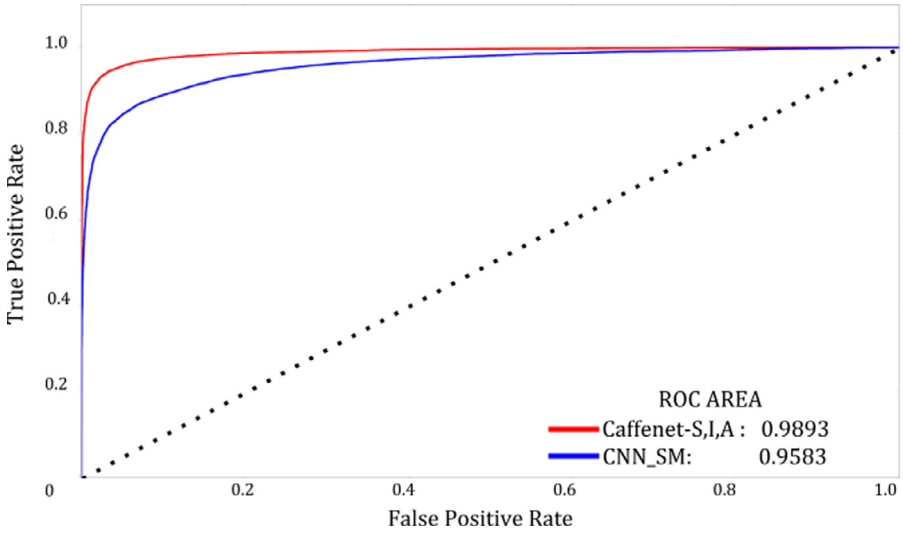


Fig. 5. ROC curves of the two best methods (Caffenet-S, I, A and CNN_SM) when evaluated on D1.

- CNN_SM architecture converged much faster to higher performance scores compared to its deeper-counterpart achieving to constantly outperform both the base and the post-processed models by a maximum of 4.6% (CNN_SM vs Caffenet-S at 500 iterations) and 2.9% (CNN_SM vs Caffenet-S,medFilt at 500 iterations) respectively. CNN_SM post-processed version lead to even further improvement of the result by almost 1% which, reflects a maximum total improvement of 3.9% (CNN_SM,medFilt vs Caffenet-S, medFilt at 500 iterations). However training a model from scratch seems to always be inferior compared to the transfer-learning approach.
- In all cases, as noted in the conclusions of Table 2, the median filtering post-processing leads to a significant increase in the final performance. With regards to the number of training iterations, the classification performance raised between a minimum of 0.25% (Caffenet-S, I vs Caffenet-S, I, medFilt at 1000 iterations) and a maximum of 1.68% (Caffenet-S vs Caffenet-S, medFilt at 500 iterations).

In Fig. 5 we show the ROC curves of the two best methods (Caffenet -S,I,A and CNN_SM) when evaluated on D1. Judging from the presented results we argue that in general CNNs can significantly outperform traditional methods on the task. However when transfer-learning is applied we observe a boost in performance equal to 3% when we compare the areas under the ROC curves of the two classifiers from almost 96% to 99%.

3.4. Comparison to other methods

One of the first efforts on speech - music discrimination is reported in Saunders (1996) were the authors achieved a classification accuracy of 96%, by adopting simple time domain features, evaluated on a real-time monitoring application of a specific radio station for 2 h of recording data. In El-Maleh et al. (2000), almost 20 min of audio data were used for training and testing purposes. The authors reported that on a short-term basis the overall accuracy was around 80%. When a mid-term window was used (1 s long), the accuracy rose to approximately 95.9%. In Carey, Parris, and Lloyd-Thomas (1999), for training and testing the classifier almost 4500 segments (10 s long each) of speech and 3000 for music (10 s length again) were. The reported experiments showed that the error rate ranges from 1.2 to 6%, however the assumption of homogeneous audio segments of quite a long duration (i.e., 10 s) is a simplified version of the problem. In later approaches such as Pikrakis et al. (2007) and Giannakopoulos, Pikrakis, and Theodoridis (2006), the authors deployed more sophisticated techniques such as dynamic programming and Bayesian networks. Those two works were evaluated on the same data (D1 test dataset) as the proposed method in this paper. They report an accuracy around 95.5%, which is comparable to current state-of-the-art approaches on such a large and diverse amount of the test data. In more detail, Table 3 presents the exact comparisons between the method proposed in Pikrakis et al. (2007) and the two dominant methods presented here, in terms of all the performance measures (speech and

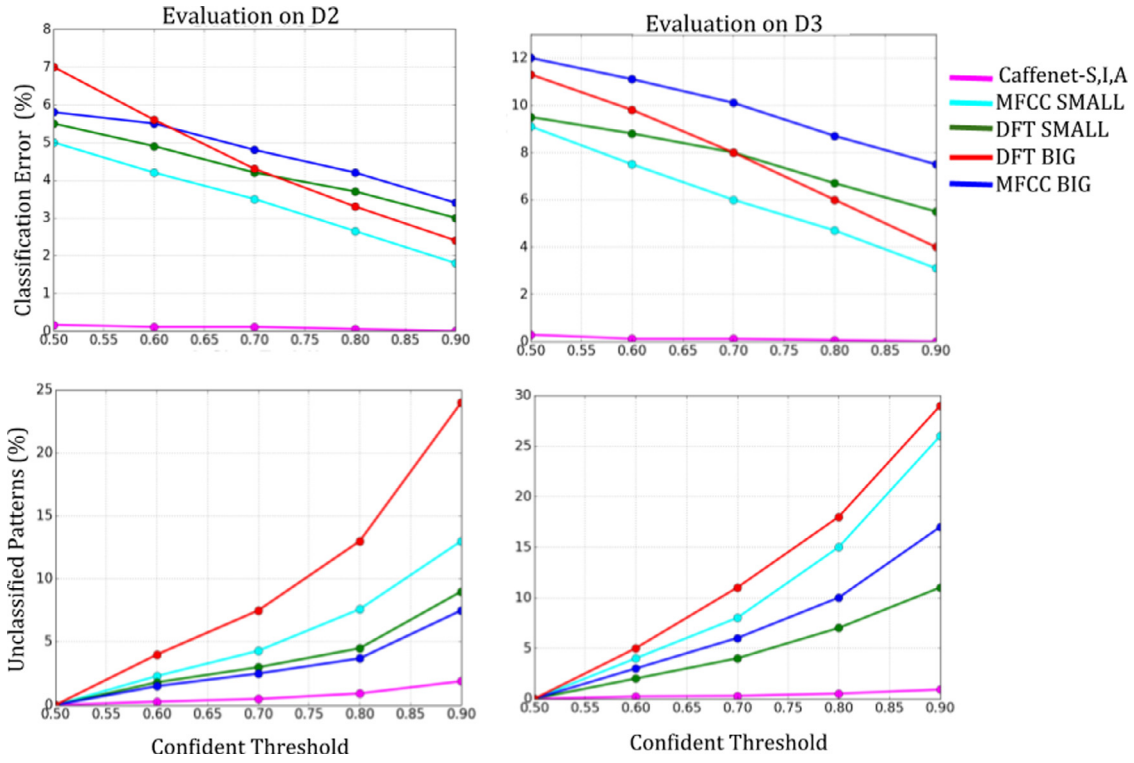


Fig. 6. Evaluation of Caffenet-S, I, A against the RBM-based deep-learning methods proposed by Pikrakis and Theodoridis (2014) that function on MFCC and DFT coefficients. Graphs on the left show evaluation on D2 test-dataset while graphs on the right refer on D3.

Table 3
Comparison between the two CNN methods presented in this paper (Caffenet-S, I, A and CNN_SM both with postprocessing) and the work presented by Pikrakis et al. (2007) on audio-based features, where a Dynamic Programming (DP) approach on a Bayesian Network was deployed. Evaluation is being done on the D1 dataset. As in Table 2 the following notations are used; Sp: Speech, Mu: Music, Rec: Recall, Prec: Precission, Av: Average.

	DP (Pikrakis et al., 2007)	Caffenet-S, I, A	CNN_SM
Sp Rec	89.2	**93.9**	**92.1**
Sp Pre	**95.8**	94.9	95.4
Mu Rec	98.3	**98.1**	**98.4**
Mu Pre	95.6	**97.6**	**96.9**
Sp F1	92.4	**94.4**	**94.3**
Mu F1	96.9	**97.8**	**97.6**
Av F1	94.7	**96.1**	**96**

Algorithm 1 CNNs for speech-music discrimination.

1. $target\_height, target\_width \leftarrow$ CNN input-image height & width
2. $m\_win, m\_step \leftarrow$ length & step of mid-term window
3. $s\_win, s\_step \leftarrow$ length & step of short-term window
4. $median\_win \leftarrow$ size of median filter for post-processing
5. **for** $i = 0; i < length(audio\_file); step\_i = s\_step$ **do**
6. $audio\_segment \leftarrow audio\_file[i : m\_win]$
7. $spectrogram \leftarrow ABS\{FFT\{audio\_segment, s\_win, s\_step\}\}$
8. $resized\_spec \leftarrow LINEAR\_INTERPOLATION$ $\{spectroogram, target\_height, target\_width\}$
9. $raw\_prediction \leftarrow CNN\_Classifier(resized\_spec)$
10. $filtered\_prediction \leftarrow median\_filter(raw\_prediction, median\_win)$

music recall and precision). In this case, to demonstrate the robustness of CNNs on the task the results associated with CNN_SM in Table 3 were estimated on the simplified image inputs (gray-scale 200×200). As it is easily observable by comparing results in both Tables 2 and 3, CNNs almost in all cases outperform the method presented by Pikrakis et al. (2007), with Caffenet-S, I, A showing again superior performance

For further experimentation we compared our dominant approach (CNN with transfer learning) against the deep learning methods proposed by Pikrakis and Theodoridis (2014) when evaluated on datasets D2 and D3 (Fig. 6). In order to have a fair comparison between the different methods we show results in a similar manner as it was done by Pikrakis and Theodoridis (2014). We present performance measurements using a confidence threshold, T_h. The goal of the threshold is to reject any classification decision if the estimated posterior probability of the winning class fails to exceed T_h. As complementary information, we also provide the percentage of patterns that have been left unclassified. Pikrakis and Theodoridis (2014) have experimented using different deep-learning methods based on MFCC and DFT coefficients. The results shown on Fig. 6 were directly derived by that publication. Our method is the one referred as *Caffenet-S, I, A*. It has to be noted, that the evaluation procedure has been carried out on a segment basis, i.e. each segment of every file in the dataset has been classified separately.

As our results indicate CNNs with transfer learning significantly outperformed all methods described by Pikrakis and Theodoridis (2014) showing a reduction in classification error of 5% and 9% on D2 and D3 respectively compared to the best deep-learning methods with RBMs that operate on MFCC and DFT coefficient features. In addition the proposed method shows a significant improvement in the confidence levels of each decision by reducing the number of unclassified patterns by almost 5% and 10% on D2 and D3 respectively.

3.5. Computational demands

We have evaluated the required computational demands for the two proposed speech music classification schemes, namely

Table 4
Execution Time (in minutes) on a GPU Tesla K40c of the two best methods for the whole testing dataset. Note that the size of this dataset is 620 min.

	Time in min		
Method	Total	Feature Ext.	CNN test
Caffenet - S, I, A	35.2	9.3	26
CNN_SM	23.5	9.3	14.3

the CNN_SM and the Caffenet network. As discussed above, the CNN_SM model is able to offer directly comparable results to its deeper counterpart when both methods where trained from scratch. At the same time, as Table 4 indicates, it requires less computational time to complete the whole evaluation process. In particular, with regards to the overall execution time required by the two models, the CNN_SM model achieves a relative computational reduction almost equal to 33%. If, however, we focus on the classification step alone, i.e. if we exclude the fixed feature extraction demand and the post-processing procedures (9.3 min for the whole dataset), the relative computational decrease is 45%. Note that, given the overall duration of the testing dataset (620 min), the CNN Caffenet-I method requires 4.2% of the true audio duration, while the CNN_SM method only 2.3% of the real audio time. This means, for example, that it takes almost a minute to classify one hour of audio data.

That is due to the decrease in the number of parameters that need to be learned and to the reduction of input's dimensionality. Based on the performance results presented in the previous subsection, it turns out that, given the simplistic visual fluctuations that spectrogram-images consist of, in terms of visual-shapes and colors, fewer parameters can be sufficient to model an audio problem with a relatively small amount of target classes, even if the available data are significantly few (around 750 samples per class).

4. Conclusions

In this work, we have examined the task of speech-music discrimination by modeling deep visual features from raw spectrogram representations. In particular, we have utilized different CNN-based approaches for audio classification and we compare our methods against both traditional and deep-learning based techniques that operate on handcrafted audio features. We show that CNNs, applied on raw spectrograms, provide state-of-the-art results and, more importantly, that transfer-learning can give us the advantage of exploiting very deep architectures as an initialization point to train robust classifiers on the task even when the original training data are relatively limited. In particular our best classifier exploits the Caffenet, a pre-trained CNN architecture proposed by Donahue et al. (2015). The method performs fine-tuning of the pre-initialized weights of the network according to the speech-music discrimination task. The network was initially trained on the vast Imagenet dataset and transferring knowledge from a different domain has been proven beneficial, despite the great irrelevance of our target application to the initial Imagenet classification task. As shown through extensive experimentation against three different datasets, parameter-tuning can provide a sturdy weight initialization that helps the network become more flexible and adapt faster and with higher performance to the requirements of the new domain. Secondarily we show that CNNs in general can provide a featureless approach - in terms of hand-crafted feature engineering that can still lead to dominant results on the task. All CNN approaches significantly outperformed traditional machine-learning and deep-learning methods on the task that are based on handcrafted audio features.

To summarize, our experimentation has proven that CNNs are a very efficient method to better discriminate between speech and music, compared to typical methods that operate on the audio domain through handcrafted audio features. Utilizing transfer learning in CNNs boosts the classification performance (from 93.7% to 95.1%) on our primary evaluation dataset, and reduces the classification error on our additional evaluation datasets D2 and D3 by 5% and 9% respectively. In addition, using CNNs and transfer learning lead to higher levels of confidence in the final decision against previously published deep-learning methods on D2 and D3 that were based on traditional audio features (MFCCs, DFT-based features).

Therefore, the key hypothesis answered in that work is that CNNs applied on raw spectrogram audio representations can provide state-of-the-art results on speech-music discrimination. Moreover transfer-learning can provide a great boost in performance and can give us the advantage to exploit very deep architectures even if the available amount of data are relatively limited to train such deep models. All methods and techniques extracted from this work are publically available, along with the implemented open-source methodologies,[5] and can be used either for further experimentation on the task or as weight initialization mechanisms to other similar classification problems.

Acknowledgments

We would like to thank NVIDIA for donating the GPU Tesla K40c.

References

Ajmera, J., McCowan, I., & Bourlard, H. (2003). Speech/music segmentation using entropy and dynamism features in a hmm classification framework. *Speech Communication, 40*(3), 351–363.

Carey, M. J., Parris, E. S., & Lloyd-Thomas, H. (1999). A comparison of features for speech, music discrimination. In *Acoustics, speech, and signal processing, 1999. proceedings., 1999 IEEE international conference on: Vol. 1* (pp. 149–152). IEEE.

Choi, K., Fazekas, G., & Sandler, M. (2016). Explaining deep convolutional neural networks on music classification. arXiv:1607.02444.

Dahl, G. E., Yu, D., Deng, L., & Acero, A. (2012). Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. *IEEE Transactions on Audio, Speech, and Language Processing, 20*(1), 30–42.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *Computer vision and pattern recognition, 2009. CVPR 2009. IEEE conference on* (pp. 248–255). IEEE.

Deng, L., Abdel-Hamid, O., & Yu, D. (2013). A deep convolutional neural network using heterogeneous pooling for trading acoustic invariance with phonetic confusion. In *2013 IEEE international conference on acoustics, speech and signal processing* (pp. 6669–6673). IEEE.

Didiot, E., Illina, I., Fohr, D., & Mella, O. (2010). A wavelet-based parameterization for speech/music discrimination. *Computer Speech & Language, 24*(2), 341–357.

Donahue, J., Anne Hendricks, L., Guadarrama, S., Rohrbach, M., Venugopalan, S., Saenko, K., & Darrell, T. (2015). Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2625–2634).

El-Maleh, K., Klein, M., Petrucci, G., & Kabal, P. (2000). Speech/music discrimination for multimedia applications. In *Acoustics, speech, and signal processing, 2000. ICASSP0. proceedings. 2000 IEEE international conference on: Vol. 6* (pp. 2445–2448). IEEE.

Giannakopoulos, T. (2015). Pyaudioanalysis: An open-source python library for audio signal analysis. *PloS One, 10*(12), e0144610.

Giannakopoulos, T., Pikrakis, A., & Theodoridis, S. (2006). A speech/music discriminator for radio recordings using bayesian networks. *2006 IEEE international conference on acoustics speech and signal processing proceedings*: Vol. 5. IEEE (pp. V–V).

Grill, T., & Schluter, J. (2015). Music boundary detection using neural networks on spectrograms and self-similarity lag matrices. In *Signal processing conference (EUSIPCO), 2015 23rd European* (pp. 1296–1300). IEEE.

Hinton, G., Deng, L., Yu, D., Dahl, G. E., Mohamed, A.-r., Jaitly, N., et al. (2012). Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine, 29*(6), 82–97.

Huang, Z., Dong, M., Mao, Q., & Zhan, Y. (2014). Speech emotion recognition using cnn. In *Proceedings of the 22nd ACM international conference on multimedia* (pp. 801–804). ACM.

[5] https://github.com/MikeMpapa/CNNs-Speech-Music-Discrimination.

Jarina, M. N. O. N. M. S. R. (2001). Speech-music discrimination from mpeg-1 bitstream. In *WSES international conference on speech, signal and image processing* (pp. 174–178).

Jarina, R., O'Connor, N., Marlow, S., & Murphy, N. (2002). Rhythm detection for speech-music discrimination in mpeg compressed domain. In *Digital signal processing, 2002. DSP 2002. 2002 14th international conference on: Vol. 1* (pp. 129–132). IEEE.

Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., et al. (2014). Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the 22nd ACM international conference on multimedia* (pp. 675–678). ACM.

Khonglah, B. K., Sharma, R., & Prasanna, S. M. (2015). Speech vs music discrimination using empirical mode decomposition. In *Communications (NCC), 2015 twenty first national conference on* (pp. 1–6). IEEE.

Kim, K., Baijal, A., Ko, B.-S., Lee, S., Hwang, I., & Kim, Y. (2015). Speech music discrimination using an ensemble of biased classifiers. In *Audio engineering society convention: 139*. Audio Engineering Society.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (pp. 1097–1105).

Lee, H., Pham, P., Largman, Y., & Ng, A. Y. (2009). Unsupervised feature learning for audio classification using convolutional deep belief networks. In *Advances in neural information processing systems* (pp. 1096–1104).

Lee, J., Park, J., Kim, K. L., & Nam, J. (2018). Samplecnn: End-to-end deep convolutional neural networks using very small filters for music classification. *Applied Sciences, 8*(1), 150.

Lopes, A. T., de Aguiar, E., De Souza, A. F., & Oliveira-Santos, T. (2017). Facial expression recognition with convolutional neural networks: Coping with few data and the training sample order. *Pattern Recognition, 61*, 610–628.

Nilufar, S., Ray, N., Molla, M. I., & Hirose, K. (2012). Spectrogram based features selection using multiple kernel learning for speech/music discrimination. In *2012 ieee international conference on acoustics, speech and signal processing (ICASSP)* (pp. 501–504). IEEE.

Panagiotakis, C., & Tziritas, G. (2005). A speech/music discriminator based on rms and zero-crossings. *IEEE Transactions on Multimedia, 7*(1), 155–166.

Papakostas, M., Spyrou, E., Giannakopoulos, T., Siantikos, G., Sgouropoulos, D., Mylonas, P., & Makedon, F. (2017). Deep visual attributes vs. hand-crafted audio features on multidomain speech emotion recognition. *Computation, 5*(2), 26.

Pikrakis, A., Giannakopoulos, T., & Theodoridis, S. (2007). A dynamic programming approach to speech/music discrimination of radio recordings. In *Signal processing conference, 2007 15th European* (pp. 1226–1230). IEEE.

Pikrakis, A., & Theodoridis, S. (2014). Speech-music discrimination: A deep learning perspective. In *2014 22nd European signal processing conference (EUSIPCO)* (pp. 616–620). IEEE.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., et al. (2015). Imagenet large scale visual recognition challenge. *International Journal of Computer Vision, 115*(3), 211–252.

Saunders, J. (1996). Real-time discrimination of broadcast speech/music. In *ICASSP: 96* (pp. 993–996).

Scardapane, S., Comminiello, D., Scarpiniti, M., & Uncini, A. (2013). Music classification using extreme learning machines. In *2013 8th international symposium on image and signal processing and analysis (ISPA)* (pp. 377–381). IEEE.

Scheirer, E., & Slaney, M. (1997). Construction and evaluation of a robust multifeature speech/music discriminator. In *Acoustics, speech, and signal processing, 1997. ICASSP-97., 1997 IEEE international conference on: Vol. 2* (pp. 1331–1334). IEEE.

Schlüter, J., & Böck, S. (2013a). Cnn-based audio onset detection mirex submission.

Schlüter, J., & Böck, S. (2013b). Musical onset detection with convolutional neural networks. *6th international workshop on machine learning and music (MML), Prague, Czech Republic*.

Sell, G., & Clark, P. (2014). Music tonality features for speech/music discrimination. In *2014 IEEE international conference on acoustics, speech and signal processing (ICASSP)* (pp. 2489–2493). IEEE.

Simard, P. Y., Steinkraus, D., & Platt, J. C. (2003). Best practices for convolutional neural networks applied to visual document analysis. In *ICDAR: 3* (pp. 958–962).

Tzanetakis, G., & Cook, P. (2000). Marsyas: A framework for audio analysis. *Organised Sound, 4*(3), 169–175.

Williams, G., & Ellis, D. P. (1999). Speech/music discrimination based on posterior probability features. In *Sixth European conference on speech communication and technology*.

Wu, Q., Yan, Q., Deng, H., & Wang, J. (2010). A combination of data mining method with decision trees building for speech/music discrimination. *Computer Speech & Language, 24*(2), 257–272.

Yang, J., Nguyen, M. N., San, P. P., Li, X., & Krishnaswamy, S. (2015). Deep convolutional neural networks on multichannel time series for human activity recognition. In *IJCAI* (pp. 3995–4001).

Zeiler, M. D., & Fergus, R. (2014). Visualizing and understanding convolutional networks. In *European conference on computer vision* (pp. 818–833). Springer.

Zhang, C., Evangelopoulos, G., Voinea, S., Rosasco, L., & Poggio, T. (2014). A deep representation for invariance and music classification. In *2014 IEEE international conference on acoustics, speech and signal processing (ICASSP)* (pp. 6984–6988). IEEE.