Insights into Audio-Based Multimedia Event Classification with Neural Networks

Mirco Ravanelli
Fondazione Bruno Kessler
Via Sommarive, 18
Trento, 38122, Italy
mravanelli@fbk.eu

Benjamin Elizalde
International Computer Science Institute
1947 Center Street
Berkeley, CA 94704, USA
benmael@icsi.berkeley.edu

Julia Bernd
International Computer Science Institute
1947 Center Street
Berkeley, CA 94704, USA
jbernd@icsi.berkeley.edu

Gerald Friedland
International Computer Science Institute
1947 Center Street
Berkeley, CA 94704, USA
fractor@icsi.berkeley.edu

ABSTRACT

Multimedia Event Detection (MED) aims to identify events—also called scenes—in videos, such as a flash mob or a wedding ceremony. Audio content information complements cues such as visual content and text. In this paper, we explore the optimization of neural networks (NNs) for audio-based multimedia event classification, and discuss some insights towards more effectively using this paradigm for MED. We explore different architectures, in terms of number of layers and number of neurons. We also assess the performance impact of pre-training with Restricted Boltzmann Machines (RBMs) in contrast with random initialization, and explore the effect of varying the context window for the input to the NNs. Lastly, we compare the performance of Hidden Markov Models (HMMs) with a discriminative classifier for the event classification. We used the publicly available event-annotated YLI-MED dataset. Our results showed a performance improvement of more than 6% absolute accuracy compared to the latest results reported in the literature. Interestingly, these results were obtained with a single-layer neural network with random initialization, suggesting that standard approaches with deep learning and RBM pre-training are not fully adequate to address the high-level video event-classification task.

Categories and Subject Descriptors

I.5 [**PATTERN RECOGNITION**]: Models; Neural nets

Keywords

Multimedia Event Classification; Multimedia Event Detection; Audio; Video; Web Video; Context Windows; Neural Networks; Hidden Markov Models.

MMCommons'15, October 30, 2015, Brisbane, Australia.

DOI: http://dx.doi.org/10.1145/2814815.2814816.

1. INTRODUCTION

Web videos have significant visual and audio content that is not fully described in their textual metadata. It is therefore necessary to develop tools that can automatically analyze that content. Multimedia event detection (MED) aims to identify the event(s) depicted in a user-generated video, such as a flash mob or a person feeding an animal, by using the content characteristics of the video.[1]

Multimedia event detection based on audio has been approached in a variety of ways—as can be seen, for example, by looking at the range of approaches used by participants in the AASP Audio (only) Scene Challenge [8] for classifying 10 different scenes. The results showed the great potential of low-level feature exploration. Many of the most successful recent approaches using audio for MED [10, 14, 4] rely on intermediate features created by a combination of low-level features, mainly Mel Frequency Cepstral Coefficients (MFCCs), followed by a Bag of (Audio) Words or a Fisher Vector encoding. The final detection in these approaches is computed using Support Vector Machines (SVMs). Some approaches focus on *audio concepts* at the semantic or humanly explainable level of analysis, including several deep neural network (DNN) approaches (e.g., [16] and [6]). In another recent example, Elizalde *et al.* (2014) use a hierarchical deep neural network (H-DNN) trained on audio concepts and complemented with a Hidden Markov Model (HMM) layer to perform audio-based video event detection or in other words a binary classification, does the video belong to an event class or not [7]. However, our main goal in the work reported in this paper was multi-class classification, or in other words to which of this event classes the video belongs.

Despite the widespread success of NNs, there is little published research on applying NNs to such a high-level, abstract task as video event classification. In one exception, Ashraf *et al.* [1] employ a DNN trained on events using only sparsely sampled audio features, to avoid training with an entire audio track. The approach reduced the training bottleneck and presented promising results.

[1]The term *event* is used in multiple ways in the video-analysis literature. The community around the TRECVID MED evaluation uses *event* to refer to a higher-level semantic abstraction, and uses the term *concept* for more specific, concrete targets such as clapping or laughing. Other researchers use the term *scenes* to refer to (more or less) what TRECVID calls *events*, and call *events* or *sounds* what TRECVID calls *concepts*. Here we use the TRECVID terminology.

However, further exploration of NNs is required to better exploit the audio cues.

In this paper, we describe work that explores the optimization of NNs for audio-based multimedia event classification, and discuss some insights for more effective application of this paradigm. We explore different architectures, including different numbers of layers and numbers of neurons; the performance impact of pre-training with an RBM [9] as opposed to random initialization; varying the length of the context windows in the NN input; and adding HMMs [15] for the event-classification step. The system architecture is described in Sec. 2, and the corpus and metrics are described in Sec. 3. In Sec. 4, we present our experiments and highlight the most relevant results. Sec. 5 summarizes our conclusions.

2. SYSTEM OVERVIEW

In this section, we provide an overview of our audio-based multimedia event classification system, whose main stages are depicted in Fig. 1. First, the audio track is extracted from the videos, then low-level features are computed. Next, utilizing an NN, a video-level event prediction is made for each frame. Lastly, a video-level event classification is made based on all the predictions from the whole audio track. The next sections provide a more detailed description of these stages.

2.1 Audio-Track and Feature Extraction

In the first step, the audio track is extracted from the videos. Then, as in Elizalde *et al.* 2014 [7], we employ standard MFCCs as the input audio features. The MFCCs include 13 dimensions plus log-energy, using a 25ms Hamming window with a stride size of 10ms per frame shift. After a mean and variance normalization step, we apply a context window that considers a fixed number of consecutive frames. (Significant benefits from using a context window are reported in Sec. 4.3.)

We chose MFCCs as the starting point because they are used in the state-of-the-art approaches described in Sec. 1. Though exploration of other low-level features is not in the scope of this work, they could well benefit system performance.

2.2 Frame-Level Predictions

The audio features represented by the context windows are then fed to a standard feed-forward neural network. The hidden layers are composed of sigmoid-based neurons, while the output layer is based on softmax activation functions. Sec. 4 describes in more detail the baseline architecture and the optimization process for the number of hidden neurons and hidden layers and for the context window size.

The NN training uses the standard back-propagation algorithm. The pre-training phases are based on the GPU version of the TNet toolkit [18]. The experiments are based on either a random initialization of the NN parameters or an RBM-based pre-training. (The role of pre-training is described in Sec. 4.2.)

The NN performs a prediction for each frame of each audio recording. The output predictions are the probabilities corresponding to any of the event classes. An interesting aspect, which makes this task particularly challenging, is that in this case the NN has to perform a frame-level prediction from the low-level features to very high-level semantic categories (e.g., *birthday party*, *parade*, *person landing a fish*), using only a limited number of frames. NNs have more traditionally been applied to audio for classification of phones [3] and audio concepts [16], which are typically less abstract.

2.3 The Video-Level Event Classifier

Given the frame-level predictions for each audio recording, an overall event classification is performed for each track. In this work, we compared two methods. The first is a discriminative classifier, which uses a cumulative probability to determine if a test file belongs to a target event, as shown in Eq. 1.

$$\hat{E} = \arg\max_{E} \prod_{i=1}^{N} P(E|x_i) \quad (1)$$

In Eq. 1, $\hat{E}$ is the estimated video event, $P(E|x_i)$ is the output of the neural network given the current feature x_i, and N is the number of features.

The second classifier is based on Hidden Markov Models, which are generative statistical models widely used for temporal pattern-detection tasks such as gesture recognition, genomics, and handwriting, as well as speech recognition [15]. Our Gaussian Mixture Model (GMM)/HMM classifier is similar to the one described in Elizalde *et al.* 2014 [7]. However, while the input features used by Elizalde *et al.* are derived from the acoustic concepts, here the features are the frame-level event predictions discussed in Sec. 2.2.

3. CORPUS AND EVALUATION METRICS

3.1 The Video Corpus

We used the YLI-MED corpus, which was inspired by TRECVID MED and is annotated for some of the same events [2]. YLI-MED is drawn from the Yahoo Flickr Creative Commons 100 Million (YFCC100M) dataset; it is the only publicly available web-video dataset annotated for events. (The popular TRECVID MED dataset created by the Linguistic Data Consortium (LDC) is available only to participants in the TRECVID MED evaluation.[2]) YLI-MED Version 1 contains 1823 videos classified into 10 events, listed in Tab. 3, as well as more than 48,000 non-event videos.

For the experiments described here, we used only the videos categorized as depicting an event. We split the dataset into three chunks. We used 1000 videos for training ($\sim$100 for each event). We employed a dev set of 319 videos for tuning the free parameters of the proposed architecture, such as number of hidden layers, number of hidden units, and length of context window. Finally, we used a test set of 480 videos for evaluation.

3.2 Evaluation Metrics

We employed two standard metrics to evaluate classifications for each event category: classification accuracy (ACC) and average precision (AP). ACC is computed by dividing the total number of event-video files that are correctly classified, called True Positives (TP), by the total number of event-video files. The AP, as shown in Eq. 2, is calculated from the true positives (TP) and the false positives (FP), where the latter is the total number of event-video files that are classified as belonging to the reference event category but actually belong to a different event category. The mean average precision (MAP) is calculated by averaging the AP scores across the 10 events.

[2]Most previous work on audio-based MED has used the TRECVID MED data set. The TRECVID MED corpus is not publicly available, so we are unable to make a direct comparison between the datasets. However, Bernd *et al.* 2015 [2] compared results for audio-based experiments with YLI-MED with published results for a similar TRECVID MED selection. They found that the MED overall accuracy for that system was roughly comparable for the two datasets.

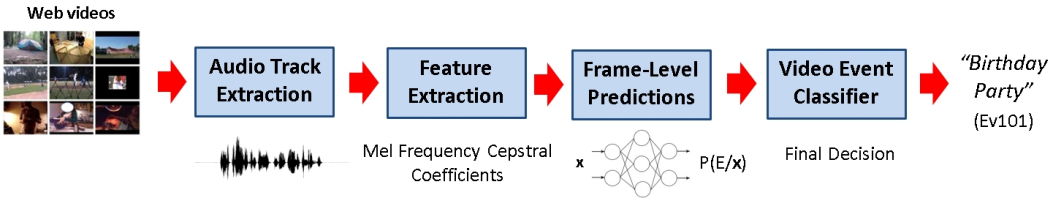

Figure 1: Pipeline for the audio-based multimedia event classification system.

$$AP = \frac{TP}{TP + FP} \tag{2}$$

4. PARAMETERS AND RESULTS

In this section, we describe the baseline systems and the evolution of our system design.

4.1 The Baseline DNN

Tab. 1 reports results from approaches to audio-based multimedia event classification that have previously been evaluated on the YLI-MED dataset. It also reports results from our baseline system modeled on those approaches.

Baseline	Ref	ACC(%)
DNN with Caffe: sparse sampling	[1]	37.40
DNN with Caffe: all frames	[1]	27.40
DNN with TNet: all frames (our baseline)	New	28.50

Table 1: Event-classification accuracy of reference systems and our system, all trained and tested on the YLI-MED video corpus.

Ashraf *et al.* [1] compared variations on two basic approaches: using sparse sampling of audio frames and using all frames. The sparse-sampling system produced higher accuracy results on the YLI-MED dataset, with a context window of 49 consecutive frames (24 before and 24 after the current one). The topology (600:600:10) was a DNN of 2 RBM pre-trained hidden layers and a softmax output layer. The discriminative event-classification approach used in that work is explained in Sec. 2.3.

The Caffe-based DNN system that used all the frames as input (representing the entire audio recording) involved two hidden layers and one softmax layer (2000:2000:2000:10); a context window of 49 consecutive frames; RBM pre-training; and the same discriminative event classifier.

As a starting architecture, we largely reproduced the DNN described in the previous paragraph. The performance we obtained differs by about 1%, perhaps because we used TNet while Ashraf *et al.* used the Caffe framework. Note that this solution leads to an initial performance far lower than that of the sparse-sampling system, mainly due to suboptimal choices for the main neural-network free parameters.

In the following subsections, we describe how we progressively optimized our baseline DNN by exploiting the dev set to design a neural network more suitable for the target task.

4.2 The Role of RBM Pre-Training

Previous work in deep learning has found that RBM pre-training leads to better initialization of the DNN parameters, and thus to substantial performance improvements, especially when the training dataset is small. In particular, the RBM technique has enabled demonstrable improvements in character recognition [9], object recognition [13], information retrieval [17], and speech recognition [11, 5], just to name a few. However, pre-training has not yet been explored in the literature on our task, so we experimented with its effects. Tab. 2 compares the baseline DNN system (with RBM pre-training) with the same system using a conventional random initialization of the NN parameters.

Initialization	ACC(%)		MAP(%)	
	DEV	TEST	DEV	TEST
RBM pre-training	24.13	28.50	19.78	22.40
Random initialization	36.36	38.12	30.83	31.74

Table 2: Video classification performance with RBM pre-training vs. random initialization of the NN parameters.

The RBM pre-training employed in this work initializes weights in the first two hidden layers via a Gaussian-Bernoulli RBM using a learning rate of 0.005 for 10 pre-training epochs. The remaining RBMs are Bernoulli-Bernoulli and use a learning rate of 0.05 for 5 pre-training epochs. The following supervised fine-tuning phase involves a stochastic gradient descent to optimize the cross-entropy loss function. Ben I modified it The learning rate is kept fixed at 0.005 as long as the single-epoch increment in dev-set frame accuracy is higher than 0.5%. For subsequent epochs, the learning rate is halved until the increment in dev-set frame accuracy is less than the stopping threshold of 0.1%. NN weights and biases are updated per blocks of 1024 frames.

Surprisingly, performance is improved by replacing the standard RBM with a simple, conventional random initialization of the NN parameters. To further confirm this finding, we also varied the values for the main parameters involved in the RBM pre-training (e.g., learning rates), but the trend is the same as that shown in Tab. 2. We believe this result can be attributed to the fact that an *event* is defined at a high level of abstraction, unlike the targets for more well-explored tasks such as speech recognition. The RBM pre-training could be acting as an unsupervised hierarchical feature detector, where the neural network progressively explores higher-level features. In this case, the features discovered via the RBM do not seem to help in explaining and representing such high-level events, so the training algorithm converges to a poor local optimum.

4.3 The Role of the Context Window

For an audio-based multimedia event classification task, one might intuitively expect that a long context window would better capture the long duration of a high-level semantic event. After all, for a human being, we could reasonably expect that a classification would be more accurate if it were performed after listening to several seconds of the audio track. However, there is a complication: for machine learning, increasing the context window would increase the input dimensionality of the NN, possibly causing dimensionality problems. In this experiment, we tested some techniques to

Events	Ev101	Ev102	Ev103	Ev104	Ev105	Ev106	Ev107	Ev108	Ev109	Ev110	ACC(%)	AP(%)
Ev101: Birthday Party	**53**	3	0	6	0	2	4	2	***9***	1	*66.2*	65.4
Ev102: Flash Mob	3	**10**	0	***6***	0	1	0	1	5	1	37.0	26.3
Ev103: Getting a Vehicle Unstuck	0	0	**6**	3	1	0	3	2	***5***	0	30.0	24.0
Ev104: Parade	5	***15***	4	**54**	0	0	1	2	3	0	*64.3*	61.4
Ev105: Person Attempting a Board Trick	1	1	7	3	**4**	4	8	8	***10***	3	*8.2*	44.4
Ev106: Person Grooming an Animal	2	0	1	1	0	**4**	5	2	***6***	5	*15.4*	14.3
Ev107: Person Hand-Feeding an Animal	7	1	4	6	2	4	**20**	6	***13***	0	31.7	37.0
Ev108: Person Landing a Fish	0	1	2	1	2	***7***	1	**10**	3	0	37.0	25.6
Ev109: Wedding Ceremony	***9***	6	1	4	0	1	***9***	5	**41**	3	51.9	41.8
Ev110: Working on a Woodworking Project	1	1	0	4	0	***5***	3	1	3	**7**	28.0	35.0

Table 3: Confusion matrix for the test set on the optimized system. Boldface indicates correct classifications and bold italics indicates the event with the most confusion. The two highest and two lowest accuracy scores (ACC) are italicized.

	ACC(%)		**MAP(%)**	
Context Window	DEV	TEST	DEV	TEST
Default window: 0.49s	36.36	38.12	30.82	31.74
Optimized window: 0.71s	41.04	41.06	38.19	33.89

Table 4: Video classification performance with the default vs. the optimized context window.

circumvent this chicken-and-egg problem. Similarly to [16], we investigated a solution based on hierarchical deep neural networks, employing several cascade neural networks able to progressively explore a wider context, of more than 10 seconds. However, despite the complexity and the great potential of such a system, the performance was not particularly encouraging. We plan to further explore and analyze the possibilities for this approach in the near future, but for this work, we decided to first explore the classification potential of a single neural network able to employ a limited local-time context of N consecutive frames.

Tab. 4 compares the best system described in Sec. 4.2, based on the default context window of 49 consecutive frames, with the same system based on a context window optimized on the dev set. To optimize, we explored contexts ranging from 5 to 101 frames, which correspond to about 1 sec.

In the AASP Audio (only) Scene Challenge [8], several researchers suggested 1 second as an optimum segment. The best dev-set performance is obtained with a context window of 71 consecutive frames (35 before and 35 after the reference frame, adding up to about 0.7 seconds). Longer context windows perform worse than the optimal one, due to the progressive impact of the dimensionality problems.

4.4 Architecture Optimization

For the experiments described in the previous subsections, we used a topology of 2000:2000:2000:10. We next explored optimization of the number of hidden layers (ranging from 1 to 5) and the number of neurons per hidden layers (ranging from 500 to 8000), performed on the dev set. Tab. 5 compares the best system so far with the same system with an architecture optimized on the dev set.

	ACC(%)		**MAP(%)**	
Architecture	DEV	TEST	DEV	TEST
Default: 2000:2000:2000:10	41.04	41.06	33.19	33.89
Optimized: 6000:10	45.45	43.54	41.48	37.53

Table 5: Video classification performance with the default vs. the optimized architecture.

The best performance, 43.54% accuracy, is obtained with a single-layer NN composed of 6000 neurons. This is an improvement over our baseline of 28.50% and even over the 37.40% achieved by the NN with sparse sampling [1]—an absolute improvement of 6.14%. Unlike other tasks where deeper and wider architectures have been used [13, 17, 11], in this case no benefits are observed by concatenating several hidden layers, since a single-layer NN performs better. This may suggest that discussion about the choice between deep vs. wide networks [12, 19] may need to be revisited. In any case, the architecture proposed here may be particularly valuable for reducing computational complexity in large-scale web-video analysis.

Tab. 3 breaks down the results achieved by the optimized system, as a confusion matrix of per-event classifications, along with computed scores. The best accuracies are obtained for Ev101 *Birthday Party* and Ev104 *Parade*. The lowest-scoring events are Ev105 *Person Attempting a Board Trick* and Ev106 *Person Grooming an Animal*. These results are not consistent with either of the systems described by Ashraf *et al.* [1] (sparse-sampling or all-frames), which achieved the highest accuracies for Ev110 *Working on a Woodworking Project* and Ev107 *Feeding an Animal* and the lowest for Ev103 *Getting a Vehicle Unstuck* and Ev108 *Landing a Fish*.

In other words, different architectures have variable effects on acoustic discrimination across different events. This suggests that future work should pursue analyzing and optimizing architectures for individual events (or subgroups of events) in order to further our understanding of the relationship between the NN architecture and the event.

4.5 Comparison with HMMs

This section compares the discriminative classifier used so far with an HMM-based classifier similar to that described by Elizalde *et al.* [7]. We optimized the topology of the HMMs on the dev set, resulting in a model based on 3 fully-connected states and 16 gaussians with a diagonal covariance matrix for each event. The decoding graph is composed of all the video-level event models in parallel. No transitions between different events are allowed within a given video file, forcing the system to choose one of the 10 event categories.

	ACC(%)		**MAP(%)**	
Classifier	DEV	TEST	DEV	TEST
Discriminative	45.45	43.54	41.48	37.53
HMM-based	46.08	40.42	39.73	35.46

Table 6: Performance of the disciminative vs. the HMM-based classifier.

Results on the test set show that the discriminative approach performs better than a more costly HMM-based solution. This may suggest that the information carried by the frame-level event predictions is already sufficiently distinctive.

5. CONCLUSIONS AND FUTURE WORK

This paper describes research exploring the use of neural networks for audio-based multimedia event detection. Specifically, we studied different NN architectures, the impact of context information, and the role of RBM pre-training, and we compared discriminative vs. HMM-based classifiers.

Results, reported on the YLI-MED corpus, show a consistent performance improvement over the latest approaches described in the literature. A context window of about 0.7 seconds seemed to capture discriminative information about the event. Most interestingly, our results suggest that deep learning is not necessarily the best approach for this high-level abstract task. In this case, a simple single-layer NN with a conventional random initialization yielded better results than deeper architectures using RBM pre-training.

Future directions we hope to explore include investigating other architectures, including convolutional and recurrent NNs, as well as event-based customization of NNs.

6. ACKNOWLEDGMENTS

This work was supported in part by the National Science Foundation under Award #1251276 (SMASH: Scalable Multimedia content AnalysiS in a High-level language), and by Lawrence Livermore National Laboratory, operated by Lawrence Livermore National Security, LLC, for the U.S. Department of Energy, National Nuclear Security Administration, under Contract DE-AC52-07NA27344. We also gratefully acknowledge the support of NVIDIA Corporation with the donation of a Tesla K40 GPU used for this research.

7. REFERENCES

[1] K. Ashraf, B. Elizalde, F. Iandola, M. Moskewicz, G. Friedland, K. Keutzer, and J. Bernd. Audio-based multimedia event detection with DNNs and sparse sampling. In *Proceedings of the 5th ACM International Conference on Multimedia Retrieval (ICMR '15)*, 2015.

[2] J. Bernd, D. Borth, B. Elizalde, G. Friedland, H. Gallagher, L. Gottlieb, A. Janin, S. Karabashlieva, J. Takahashi, and J. Won. The YLI-MED corpus: Characteristics, procedures, and plans. *ICSI Technical Report TR-15-001*, 2015.

[3] H. Bourlard and N. Morgan. Continuous speech recognition by connectionist statistical methods. *IEEE Transactions on Neural Networks*, 4(6):893–909, 1993.

[4] H. Cheng, J. Liu, S. Ali, O. Javed, Q. Yu, A. Tamrakar, A. Divakaran, H. S. Sawhney, R. Manmatha, J. Allan, A. Hauptmann, M. Shah, S. Bhattacharya, A. Dehghan, G. Friedland, B. M. Elizalde, T. Darrell, M. Witbrock, and J. Curtis. SRI-Sarnoff AURORA system at TRECVID 2012: Multimedia event detection and recounting. In *Proceedings of TRECVID 2012*. NIST, USA, 2012.

[5] G. E. Dahl, D. Yu, L. Deng, and A. Acero. Context-dependent pre-trained deep neural networks for large vocabulary speech recognition. In *in IEEE Transactions on Audio, Speech, and Language Processing*, volume 20, 2012.

[6] B. Elizalde, M. Ravanelli, and G. Friedland. Audio concept ranking for video event detection on user-generated content. In *Proceedings of SLAM@INTERSPEECH*, 2013.

[7] B. Elizalde, M. Ravanelli, and G. Friedland. Audio-concept features and hidden Markov models for multimedia event detection. In *Proceedings of SLAM@INTERSPEECH*, 2014.

[8] D. Giannoulis, E. Benetos, D. Stowell, M. Rossignol, M. Lagrange, and M. D. Plumbley. Detection and classification of acoustic scenes and events: an IEEE AASP challenge. In *2013 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pages 1–4. IEEE, 2013.

[9] G. E. Hinton, S. Osindero, and Y.-W. Teh. A fast learning algorithm for deep belief nets. *Neural Computation*, 18:1527–1554, 2006.

[10] Z. Lan, L. Jiang, S.-I. Yu, C. Gao, S. Rawat, Y. Cai, S. Xu, H. Shen, X. Li, Y. Wang, W. Sze, Y. Yan, Z. Ma, N. Ballas, D. Meng, W. Tong, Y. Yang, S. Burger, F. Metze, R. Singh, B. Raj, R. Stern, T. Mitamura, E. Nyberg, and A. Hauptmann. Informedia @ TRECVID 2013. In *Proceedings of TRECVID 2013*. NIST, USA, 2013.

[11] A. Mohamed, G. E. Dahl, and G. E. Hinton. Deep belief networks for phone recognition. In *NIPS Workshop on Deep Learning for Speech Recognition and Related Applications*, Vancouver,Canada, 2009.

[12] N. Morgan. Deep and wide: Multiple layers in automatic speech recognition. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(1):7–13, 2012.

[13] V. Nair and G. E. Hinton. 3-D object recognition with deep belief nets. In *Advances in Neural Information Processing Systems*, 2009.

[14] P. Natarajan, P. Natarajan, S. Wu, X. Zhuang, A. Vazquez Reina, S. N. Vitaladevuni, K. Tsourides, C. Andersen, R. Prasad, G. Ye, D. Liu, S.-F. Chang, I. Saleemi, M. Shah, Y. Ng, B. White, L. Davis, A. Gupta, and I. Haritaoglu. BBN VISER TRECVID 2012 multimedia event detection and multimedia event recounting systems. In *Proceedings of TRECVID 2012*. NIST, USA, 2012.

[15] L. R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. In *Proceedings of the IEEE*, pages 257–286, 1989.

[16] M. Ravanelli, B. Elizalde, K. Ni, and G. Friedland. Audio concept classification with hierarchical deep neural networks. In *Proceedings of EUSIPCO*, 2014.

[17] R. Salakhutdinov and G. E. Hinton. Semantic hashing. In *International Journal of Approximate Reasoning*, 2009.

[18] K. Vesely, L. Burget, and F. Grezl. Parallel training of neural networks for speech recognition. In *Proceedings of INTERSPEECH*, 2010.

[19] O. Vinyals and N. Morgan. Deep vs. wide: Depth on a budget for robust speech recognition. In *Proceedings of INTERSPEECH*, 2013.