Audio-Based Multimedia Event Detection with DNNs and Sparse Sampling

Khalid Ashraf[1], Benjamin Elizalde[2], Forrest Iandola[1], Matthew Moskewicz[1],
Julia Bernd[2],Gerald Friedland[2], Kurt Keutzer[1]
[1]EECS Department, University of California, Berkeley, CA
[2]International Computer Science Institute, Berkeley, CA
{ashrafkhalid,forresti,moskewcz,keutzer}@berkeley.edu
{benmael,jbernd,fractor}@icsi.berkeley.edu

ABSTRACT

This paper presents advances in analyzing audio content information to detect events in videos, such as a parade or a birthday party. We developed a set of tools for audio processing within the predominantly vision-focused deep neural network (DNN) framework Caffe. Using these tools, we show, for the first time, the potential of using only a DNN for audio-based multimedia event detection. Training DNNs for event detection using the entire audio track from each video causes a computational bottleneck. Here, we address this problem by developing a sparse audio frame-sampling method that improves event-detection speed and accuracy. We achieved a 10 percentage-point improvement in event-classification accuracy, with a 200x reduction in the number of training input examples as compared to using the entire track. This reduction in input feature volume led to a 16x reduction in the size of the DNN architecture and a 300x reduction in training time. We applied our method using the recently released YLI-MED dataset and compared our results with a state-of-the-art system and with results reported in the literature for TRECVID MED. Our results show much higher MAP scores compared to a baseline i-vector system—at a significantly reduced computational cost. The speed improvement is relevant for processing videos on a large scale, and could enable more effective deployment in mobile systems.

Categories and Subject Descriptors

G.3 [**PROBABILITY AND STATISTICS**]: Statistical software; Time series analysis

Keywords

Multimedia Event Detection; Audio; Video; Deep Neural Networks; Caffe

ICMR '15 June 23 - 26, 2015, Shanghai, China

DOI: http://dx.doi.org/10.1145/2671188.2749396.

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 someone making a sandwich, by using the content characteristics of the video. Much recent work in this area is related to NIST's annual TRECVID Multimedia Event Detection evaluation, and results in that benchmark define the state of the art.

Multimedia event detection based on audio has been approached in a variety of ways. Many of the most successful recent approaches [6, 8, 2] rely mainly on a combination of low-level features, especially Mel Frequency Cepstral Coefficients (MFCCs), followed by a Bag of Words. The final detection in these approaches is computed using Support Vector Machines (SVMs). Additionally, recent MED research has increasingly revolved around semantic or humanly explainable approaches. As a result, the focus in audio has shifted toward detecting identifiable audio concepts such as laughter or clapping. For example, a TNET-based [10] deep neural network (DNN) has been employed for an audio concept–classification step and complemented with Hidden Markov Models (HMMs) for audio-based event detection [4]. However, a review of recent work shows that DNNs have been explored largely for the computer vision aspect of MED and for fusion steps. Despite their high accuracy in other domains, DNNs alone have not yet been used for audio-based video-event detection.

Last year, the computer-vision DNN framework Caffe won the ACM Multimedia 2014 Open Source Software competition. It is a clean and modifiable framework for state-of-the-art deep learning [5], so we decided to leverage this efficient, well-known tool for our task. In this work, we developed tools based on Caffe that can be used for audio processing. Then we showed, for the first time, the potential of using only DNNs (without HMMs) for the audio-based video-event detection task. Training DNNs for event detection using the entire audio track from each video is computationally expensive. We addressed this problem by developing a sparse audio frame–sampling method that optimizes the representation of the audio file for event-detection performance. We achieved a notable event-classification accuracy improvement (10 percentage points), with a 200x reduction in the number of training input examples required as compared to dense sampling. This reduction in input feature volume en-

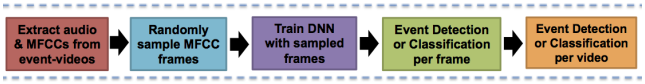

Figure 1: Pipeline for audio-based multimedia event detection or classification using a DNN.

abled a 16x reduction in the size of the DNN architecture and a 300x reduction in training time. We applied our methods to the recently released open-source YLI-MED dataset, and compared our results with a state-of-the-art system and with results reported in the literature for TRECVID MED 2013. We achieve higher MAP scores than a baseline i-vector system for audio-based video-event detection.

Our methodology and tools are described in Sec. 2, along with the comparison system. The datasets are described in Sec. 3. In Sec. 4, we present our experiments and highlight the most relevant results. Sec. 5 summarizes our conclusions.

2. DESCRIPTION OF SYSTEMS

In Sec. 2.1, we describe the feature extraction step that begins the pipeline for both systems we tested in this work. In Sec. 2.2, we describe how we used the deep neural network framework Caffe for audio-based video event detection and classification. In Sec. 2.3, we describe the i-vector detection system, which we used as a baseline for comparison.

2.1 Audio Preprocessing

In both tasks, the first step was to extract MFCC features from the audio track. We decided to use this standard preprocessing feature in these initial experiments for the sake of comparability. For the DNN, we used a total of 14 feature dimensions, including energy, while for the i-vector, we used 20, with delta and double delta for a total of 60 dimensions. Each feature frame was computed using a 25 ms Hamming window with a stride size of 10ms per frame shift. After a mean and variance normalization step, we applied a context window of 49 consecutive frames (centered at the 25^{th} frame). We next performed a feature size reduction step in which the Discrete Cosine Transform (DCT) was applied to decorrelate the information in time. The total NN input feature consists of 462 frames.

2.2 Audio Processing Using Caffe

Our full pipeline for event detection and classification using a DNN is depicted in Figure 1. In sum, the MFCC features are sparsely sampled and fed into the neural net (more details on sampling are given in Sec. 4.1). Each sample training frame is labeled with an event name. The DNN performs classification or detection on the test MFCC frames, to identify the video as belonging to a particular event. For classification, the DNN is trained on all the events and performs multi-class classification to identify videos as depicting one of the events. In detection, the DNN performs binary classification between a given event and the negative event. There is thus one binary detector for each event category.

Caffe [5] is essentially oriented toward vision applications. Our first step in using Caffe as the basis for a video-analysis framework was to develop or incorporate tools specifically for analyzing audio signals; these tools are open-source and publicly available.[1] We incorporated a context window for capturing correlation in the temporal dimension, a Discrete Cosine Transform to decorrelate the data, and a Restricted Boltzmann Machine (RBM) for pre-training the DNN.

[1] `https://github.com/ashrafk/audioCaffeInitial`

2.3 The i-Vector System

I-vector-based systems have been previously used for audio-based event detection in the TRECVID MED evaluations [11, 3], and thus provide a helpful baseline for comparison. The i-vector can be thought of as a low-dimensional representation of the identity of each event class. A log-likelihood ratio for similarity between test data and event classes is computed using a generative Probabilistic Linear Discriminant Analysis (pLDA). The Within-Class Covariance Normalization (WCCN) and pLDA system components normalize for the within- and between-class i-vector scatter of the events. This accounts for cases where examples of the same event have distinctive audio profiles, and where different events have similar audio elements. Details on the system used for these experiments are in Elizalde et al. 2013 [3].

3. CORPORA USED

The TRECVID MED dataset [9] used in much previous MED work is comprised of web videos labeled for the events depicted; however, use is restricted to evaluation-related research. Therefore, for these experiments we used YLI-MED, which is inspired by TRECVID MED and is annotated for some of the same events [1]. YLI-MED is drawn from the Yahoo Flickr Creative Commons 100 Million (YFCC100M) dataset.YLI-MED Version 1 contains 10 events as well as non-event videos. For training, there are 1000 event videos (100 per) and 5,000 non-event videos. For testing, there are 823 event videos and 43,638 non-event videos.

4. EXPERIMENTS & RESULTS

In this section, we present multimedia event detection and classification results obtained using our adapted Caffe framework vs. a baseline system. First, using the YLI-MED dataset, we studied event *classification*, tuning our audio–frame sampling method (Sec. 4.1) and neural net architecture (Sec. 4.2) for classification speed and accuracy. In Sec. 4.3, we describe how we included YLI-MED non-event background samples in the training and testing sets to study event *detection*. We compared our event-detection results with other results in the literature; we also used an i-vector system trained on TRECVID MED and tested on YLI-MED to show the comparability between the two datasets (Sec. 4.3).

4.1 Sparse Audio Sampling for Speed and Accuracy

The main bottleneck in performing DNN-based video event detection is at the input stage. The extracted audio features are represented as frames.Web videos are usually between 2 and 3 minutes long, and thus processing thousands of videos—each including thousands of frames—is slow. Moreover, global optimization of a DNN with a stochastic gradient descent method is difficult with such a high number of example frames.

Answering the question "What is this video about?" requires listening to multiple segments of the video. We were interested in seeing if we could achieve a comparable result by using arbitrary segments of the video rather than the

Method	Per-Frame Accuracy(%)	Video-Level Accuracy(%)
DNN(2000:2000:2000:10), All Frames Input	18.3	27.4
DNN(2000:2000:2000:10), 100 Frames Input	28.6	36.8
DNN(600:600:10), 100 Frames Input	29.3	37.4

Table 1: Accuracy of selected DNN-based event-classification configurations tested in this work.

whole. It turns out that it is possible to achieve not only improved efficiency but also improved accuracy. The DNN can find correlations among these arbitrarily sampled points, and from that, represent the entire event.

For each training and test video in the YLI-MED dataset, we sampled a fixed number of frames in each trial, varying the frame separation based on the video length. In our experiments, we varied the number of sampled frames per video from 20 to 150. We found that 100 frames per video gave the highest neural net per-frame accuracy, at 28.6%, with a video-level event classification accuracy of 36.8%. In contrast, when densely feeding the entire audio track into the DNN, the per-frame accuracy was only 18.3%, which translated to a video-level classification accuracy of 27.4%.

4.2 Optimizing the Neural Net Architecture

Having selected an efficient and accurate frame sampling method, we moved on to exploring neural network architectures. Starting with a 400:400:10 network (where 400:400:10 means 400d inner product → sigmoid → 400d inner product → sigmoid → 10d softmax), we initially increased the number of hidden units by increments of 100 in the layers up to 800. We found that a 600:600:10 neural net gave the highest per-frame accuracy, at 29.3%, which resulted in 37.4% accuracy on video-level event classification. These results are summarized in Table 1. We also varied the depth of the neural net. However, adding another layer did not improve the accuracy significantly.

4.3 Comparing Methods for Event Detection

We used the sparse 100-frame audio sampling and the DNN architecture described in Secs. 4.1 and 4.2 to perform event detection. For binary detection, we replaced the 10d softmax layer with a 2d softmax layer (600:600:2). In this section, we describe the detection methods and results, and compare them with those described in previous literature.

4.3.1 MAP scores

For detection, the DNN was trained on the individual YLI-MED event videos (100 per) and a similar number of negatives. Once the DNN was trained to achieve the highest cross-validation accuracy, posterior probabilities were calculated for the test positive examples for each event and for the full set of ˜43,600 test negative examples. Cumulative probability was used to determine whether each test file belonged to each target event. Average precision (AP) for each event was calculated from the true positives (TP) and false positives (FP) among the full 44K test set. The mean average precision (MAP) is calculated by averaging the AP scores for individual events over all the 10 YLI-MED events. The resulting MAP scores are shown in Table 2.

Event Category	i-vector TV/YL	i-vector YL/YL	Caffe YL/YL
Birthday Party	0.31	0.37	1.10
Flash Mob	0.22	0.12	0.89
Getting a Vehicle Unstuck	0.07	0.12	0.61
Parade	0.21	0.32	1.62
Person Attempting a Board Trick	0.20	0.23	1.34
Pers. Grooming an Animal	0.09	0.11	1.12
Person Hand-Feeding an Animal	0.22	0.28	1.71
Person Landing a Fish	0.05	0.10	0.67
Wedding Ceremony	0.25	0.32	0.92
Working on a Woodworking Project	0.11	0.19	1.81
Overall MAP	0.17	0.22	1.18

Table 2: Event-detection MAP percentage scores for: the i-vector system trained on TRECVID MED (col1), and trained on YLI-MED (col2), both tested on YLI-MED; Caffe-based DNN system (col3) trained and tested on YLI-MED.

Most previous work on audio-based multimedia event detection has used the TRECVID MED dataset. The highest reported audio-based MAP score for the TRECVID MED 2013 dataset is 14.6% [7]. TRECVID MED data is not publicly available, so we were unable to make a direct comparison with the state of the art. However, we had previously trained an i-vector system on the TRECVID MED 2013 data (see Sec. 2.3), achieving a MAP of 7% testing it on that data [3]. For the current project, we used this TRECVID MED-trained system and tested it on YLI-MED, to assess the comparability of the two datasets, before using YLI-MED to compare the i-vector and DNN detection approaches. Under these conditions, i-vector achieved a MAP score of 0.17%.[2] Then we both trained and tested the i-vector system on the YLI-MED dataset. The MAP score in this case improved to 0.22%.

In comparison, the DNN event-detection system with sampled-MFCC input achieved a MAP of 1.1% on the YLI-MED, at a much lower computation cost (see Sec. 4.3.2). It is exciting to observe such improvement in MAP scores compared to the i-vector system on the same dataset, at a significantly lower computational cost. Also, it should be noted that the DNN-based detection was performed on a raw input signal, without segmentation. In contrast, the state-of-the-art result [7] was obtained after performing unsupervised clustering on the input audio, and also with many fewer negative examples (25k in TRECVID 2013 compared to 43K in YLI-MED). Our sparsely sampled DNN method has proven its potential by showing improved accuracy results compared to full audio samples, and we anticipate that fine-tuning the accuracy by segmenting the input audio will be a very promising area for future research.

4.3.2 Speed and memory

Irrespective of dataset, the Caffe framework consistently gave a ˜1.5x speedup on a Tesla K40 GPU, compared to a TNET-based DNN [10]. The next level of speedup came

[2]It is not known how much overlap there may be between the training and test sets across the two datasets.

Method	Train Time (hr)	Test Time (hr)	Training Speedup	Model Size (MB)
i-Vector	2.71	7.8	1x	5100
DNN: Full	10.33	NA	0.26x	48
DNN: Sparse Sampled	0.034	0.748	78.4x	3

Table 3: Speed and memory required for the event-detection methods compared in this work.

from representing the input video events with sparsely sampled frames of the MFCC features. Sparse frame sampling reduces the training- and test- data volume significantly. A typical video in the YLI-MED dataset is around 3 minutes long, so there are about 18,000 MFCC frames per video. Sampling only 100 audio frames per video resulted in about a 200x reduction in input frame numbers, while still achieving high classification accuracy results.

Reducing the number of input frames in turn reduces the size of the DNN required to obtain high accuracy. For example, as shown in Table 3, with the full-length audio track, the best DNN architecture is 2000:2000:2000:10 (to achieve 18.3% per-frame classification accuracy), while sparse sampling of the input frames gave an optimum per-frame accuracy with a 600:600:10 DNN. The DNN with dense audio sampling takes up about 48MB of storage, whereas the DNN with sparse frame input takes only about 3MB. In contrast, the baseline i-vector system takes 5.1GB—that is, 1700x more than the sampled DNN system. The smaller DNN facilitated a 300x and 78.4x training-time speedup compared to the full-input DNN and the i-vector system, respectively. We could not perform a complete test of the full-input DNN due to the prohibitively long run time, but an overall 10x testing speedup was observed for the sampled DNN compared to the i-vector system.

5. CONCLUSIONS

In summary, we have built several tools for audio analysis within the DNN framework Caffe, which we have made publicly available. We explored, for the first time, the potential of using DNNs without HMMs for audio-based video-event detection. We achieved a 10 percentage-point improvement in event-classification accuracy by optimizing sampling and DNN topology, with a 200x reduction in the number of input frames as compared with using all of the frames. This reduction in input frame numbers resulted in a 16x reduction in the size of the DNN required for maximum accuracy. The combination of input-data volume and DNN size reduction gave an overall 300x and 78.4x speedup in training time compared to the full-audio DNN and i-vector systems respectively. When evaluated on the newly released YLI-MED dataset, our DNN system with sparse frame sampling showed higher accuracy than the baseline i-vector results, at a significantly lower computation cost. These improvements in speed and accuracy are relevant for processing videos at a larger scale, as well as for potential deployment of video analysis on mobile platforms.

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). It was also supported 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.

7. REFERENCES

[1] 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). *arXiv:1503.04250*, 2015.

[2] 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, et al. SRI-Sarnoff AURORA system at TRECVID 2012: Multimedia event detection and recounting. In *Proceedings of TRECVID 2012*. NIST, USA, 2012.

[3] B. Elizalde, H. Lei, and G. Friedland. An i-vector representation of acoustic environments for audio-based video event detection on user generated content. In *ISM*, 2013.

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

[5] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014.

[6] 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, et al. Informedia @ TRECVID 2013. In *Proceedings of TRECVID 2013*. NIST, USA, 2013.

[7] F. Metze, S. Rawat, and Y. Wang. Improved audio features for large-scale multimedia event detection. *ICME*, 2014.

[8] 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. Shahand, Y. Ng, et al. BBN VISER TRECVID 2012 multimedia event detection and multimedia event recounting systems. In *Proceedings of TRECVID 2012*. NIST, USA, 2012.

[9] P. Over, G. Awad, M. Michel, J. Fiscus, G. Sanders, W. Kraaij, A. F. Smeaton, and G. Quénot. TRECVID 2013 – an overview of the goals, tasks, data, evaluation mechanisms and metrics. In *Proceedings of TRECVID 2013*. NIST, USA, 2013.

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

[11] X. Zhuang, S. Tsakalidis, S. Wu, P. Natarajan, R. Prasad, and P. Natarajan. Compact audio representation for event detection in consumer media. In *INTERSPEECH*, 2012.