IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Detection of Advertisement Video Shots among Normal Shots Using MFCC Features of Audio

Soumya Majumdar, K. Sreenivasa Rao

PhD Scholar, Professor
Department of Computer Science & Engineering
Indian Institute of Technology Kharagpur, Kharagpur, India

Abstract: Scene boundary detection is crucial in applications like scene segmentation and video skimming, but the presence of ad clips makes it difficult. There are some existing methods based on visual, audio-visual and audio-only features. The existing audio-only feature-based method depends on short silences between program and ad or between ads. Still, short silences can also be present inside the program, affecting performance. So we proposed a detection method for ad shots using MFCC features. MFCC is a basic speech feature based on short-term spectral. We used the average of MFCC for a shot and determined the threshold for detection purposes. We used four episodes of The Big Bang Theory as our dataset. Our method is the first method that uses MFCC features for ad detection and explores the relation between video contents and MFCC features.

Index Terms **-Advertisement detection, Mel frequency cepstral coefficient, Static Threshold, Dynamic Threshold.**

I. INTRODUCTION

Scene boundary detection is an important job which is used in many applications like scene segmentation, video skimming etc. But the presence of advertisements clips between videos makes scene boundary detection jobs difficult for given videos. Presence of advertisement clips also makes activity detection from video scenes more difficult. For this reason, advertisement clips should be avoided during activity detection.

There are three existing methods for advertisement detection in a video: using visual features, using audio-visual features and using audio-only features. An advertisement detector system is proposed by Sadlier et. al. [1] that automatically detects the commercial breaks from the bitstream of digitally captured television broadcasts using audio-visual features. They used two audio-visual properties: advertisement breaks during/ between television programmes are typically recognised by (i) series of 'black' video frames (ii) depression in audio volume. They separate each advertisement from one another by recurrently occurring before and after each advertisement using these two properties mentioned before. Their approach failed in two cases: (i) occurrence of individual advertisements longer than 90 secs in duration and (ii) ad-breaks consisting of less than 3advertisement. An audio-only method for advertisement detection in broadcast television content is proposed by Ramires et. al. [2] which depends on the detection of short silences which exist at the boundaries between the program and advertising or advertisements themselves. Their method got 89.3 % accuracy on the Portuguese TV dataset. But this method has some drawbacks, for example- short silences can also be present inside the program and will affect the overall performance.

In our work, we aim to design an audio-only method for detection of video shots containing advertisements. Our method will work on given video shots and classify them as advertisement/non-advertisements. Our method is also helpful for extraction of advertisement segments from the given video.

Mel-frequency cepstral coefficient is a basic music feature. It is a short-term spectral based feature[5]. They are a set of perceptually motivated features that have been widely used in speech recognition. They provide a compact representation of the spectral envelope in such a way that most of the signal energy is concentrated in coefficients[5]. But MFCCs are not robust to noise because performance of MFCCs in presence of additive noise has not always been good in comparison to other features.

In our audio-based approach, we used an average of Mel-frequency cepstral coefficients throughout the length of the audio of video shots as the feature. We determined a threshold for detection of advertisement shots. If the average is greater than the threshold, then the corresponding shot contains advertisement.

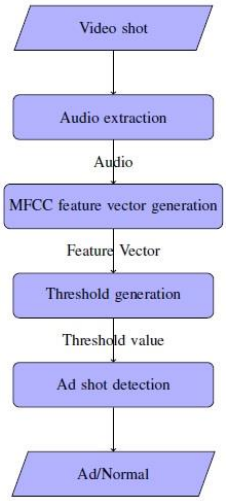


Fig 1. Framework of our approach

II. CONTRIBUTION

There are some ad detection approaches like ADNet [3] which uses only visual features from an image and using dB as an audio feature [2]. Our contributions over them are:
(i) Our method is based on MFCC features from the audio part of the video part. MFCC features are applied on many applications, but first time it is applied for ad detection purposes.
(ii) Our method also explored the relation between video contents (ad/non-ad) and MFCC coefficients

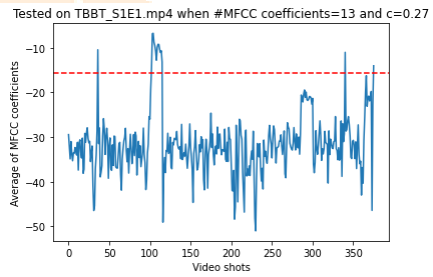


Fig 2. Average of MFCC coefficients of all video shots including in a test video. It is observed here that some shots have values comparatively greater than other values and they contain advertisements. In this figure, a threshold is indicated by a red line. Our approach considers the video shots having values above the red line as an ad shot.

III. METHOD

Our aim is to propose an audio-only approach for detecting ad-shots. We noticed that during the ad, audio tracks were changed. So we thought of catching it using some features. So we extracted MFCC features from each video shot. Those MFCC values are in the vast range, so we decided to determine maximum, minimum and average of those values. It is noticed that maximum and minimum values didn't bear any resemblance with ad-shots, but average values are getting high during ad-shots.

Our approach consists of four steps.

3.1 Audio extraction from video

For each video shot, the audio will be extracted. (Each video shot is in ".mp4" format and extracted audio files will be saved in ".wav" format.)

3.2 MFCC feature vector generation

For each audio clip, mfcc features will be extracted. Each feature vector contains coefficients per length for that audio clip. The number of coefficients can be considered as 13. Average, maximum and minimum of all of these coefficients are evaluated.

3.3 Threshold generation

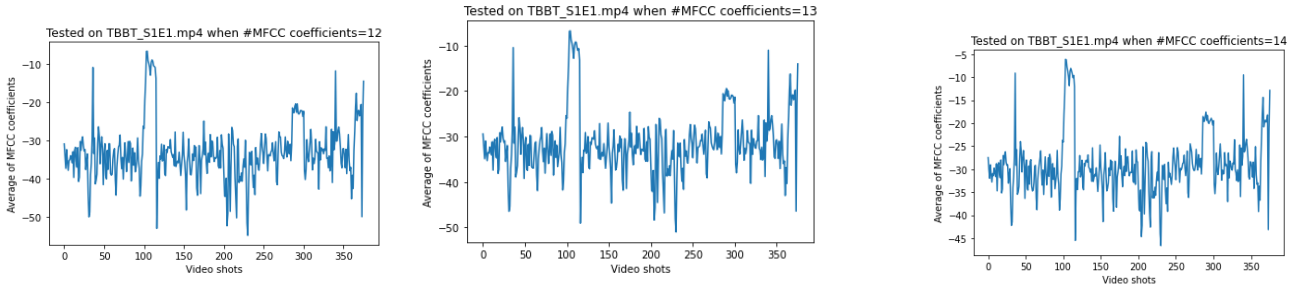
The dynamic threshold is considered instead of a static threshold because the static threshold can lead towards less accurate results. So we evaluated a threshold formula which will depend upon the input video shots. The threshold for each video will be determined using the following formula:

$$T = (M+m)*c \quad \ldots\ldots\ldots\ldots\ldots(1)$$

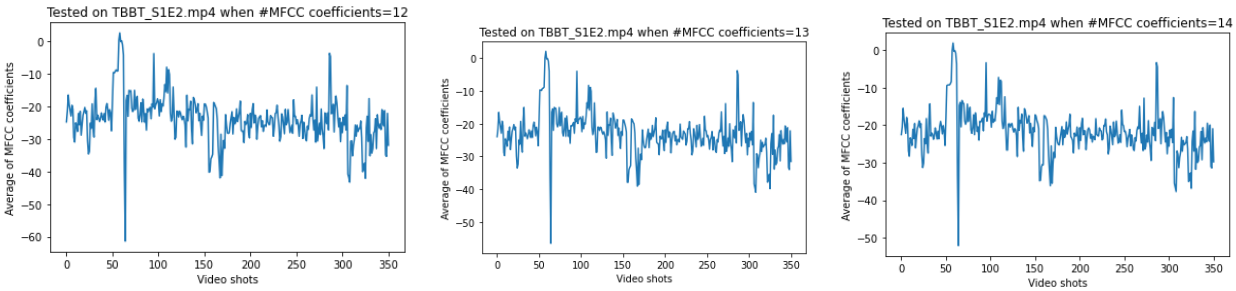
Where T is the desired threshold, M is the maximum coefficient value, m is the minimum coefficient value and c is a small constant value less than 1. Value of c can be in the range of 0.24-0.28.

3.4 Ad shot detection

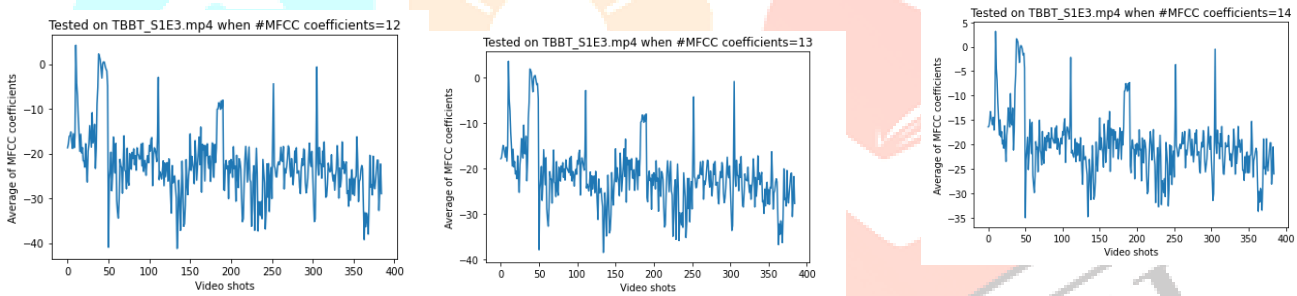
If the average value is greater than T, then that video shot will be classified as an ad shot. Otherwise, it will be classified as a non-ad shot.



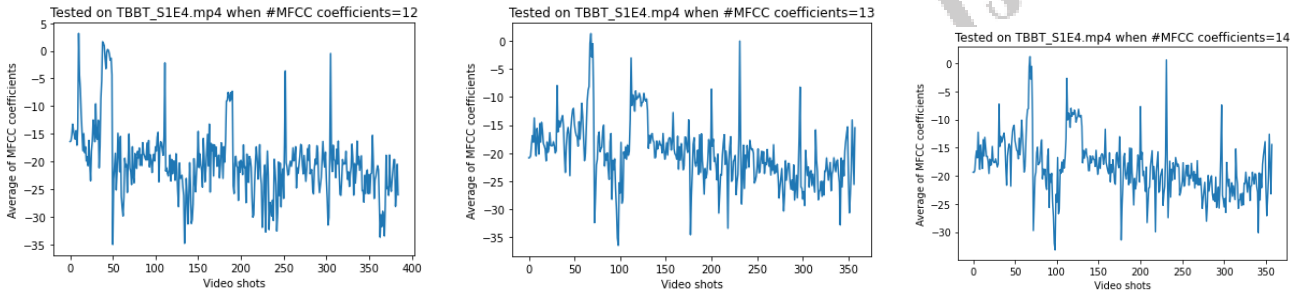
(a) Test video: TTBT_S01E01.mp4



(b) Test video: TTBT_S01E02.mp4



(c)Test video: TTBT_S01E03.mp4



(d) Test video: TTBT_S01E04.mp4

Fig 3. The average of MFCC values of all video shots are plotted in case of number of extracted MFCC values are 12/13/14 for test videos (a) TTBT_S01E01.mp4 (b) TTBT_S01E02.mp4 (c) TTBT_S01E03.mp4 (d) TTBT_S01E04.mp4

IV. RESULTS AND DISCUSSION

4.1 Experimental Setup

We evaluated our approach over an annotated dataset. Our dataset is four episodes of an American sitcom named The Big Bang Theory which contain advertisements. Each episode length is around 22 minutes including advertisements. The advertisements in the video dataset are annotated at the shot level. Each video is divided into shots and the shots are annotated as advertisement or non-advertisements.

We performed the MFCC extraction using a python library called librosa. During MFCC extraction, Sample rate was 22.05 kHz. Type 2 Discrete Cosine Transformation(DCT) and Orthogonal Normalization was used.

4.2 Results

Our performance metrics are F-score and Accuracy.

$$Precision=Tp/Tp+Fp \quad (4.1)$$

$$Recall=Tp/Tp+Fn \quad (4.2)$$

Where Tp= True positive, Fp= False positive, Fn= False negative

$$F\text{-score}= (2*Precision*Recall)/(Precision+Recall) \quad (4.3)$$

$$Accuracy=(Tp+Tn)/(Tp+Tn+Fp+Fn) \quad (4.4)$$

Where Tp= True positive, Tn= True negative, Fp= False positive, Fn= False negative

Table 4.1 shows average F-score and Accuracy over Dataset

Constant Value	F-score	Accuracy
0.24	0.7992	0.9832
0.245	0.8215	**0.9845**
0.25	0.8245	0.9839
0.2525	**0.8275**	0.9839
0.255	0.8142	0.9832
0.26	0.8164	0.9832
0.265	0.8164	0.9825
0.27	0.8004	0.9811
0.275	0.7883	0.9798
0.28	0.7721	0.9773

Table 4.1: Average F-score and Accuracy over dataset

From table 4.2, it is cleared that our approach gives the best performance when the value of the positive constant c proposed by us is in the range 0.245-0.25252. One performance metric F-score gives the best result at 0.2525 and another performance metric Accuracy gives the best result at 0.245.

Table 4.2 shows Precision, Recall, F-score and Accuracy on each video of the dataset

Constant Value	Video Name	Precision	Recall	F-score	Accuracy
0.245	TTBT_S01E01.mp4	1.0	0.8889	0.9412	0.9947
0.245	TTBT_S01E02.mp4	0.6667	0.75	0.7059	0.9723
0.245	TTBT_S01E03.mp4	0.7273	0.9412	0.8205	0.9821
0.245	TTBT_S01E04.mp4	0.9	0.75	0.8182	0.989
0.25	TTBT_S01E01.mp4	1.0	0.8889	0.9412	0.9947
0.25	TTBT_S01E02.mp4	0.6	0.75	0.6667	0.9821
0.25	TTBT_S01E03.mp4	0.7273	0.9412	0.8205	0.9821
0.25	TTBT_S01E04.mp4	0.9091	0.8333	0.8696	0.9917
0.2525	TTBT_S01E01.mp4	1.0	0.9444	0.9714	0.9973
0.2525	TTBT_S01E02.mp4	0.5714	0.75	0.6486	0.9643
0.2525	TTBT_S01E03.mp4	0.7273	0.9412	0.8205	0.9821
0.2525	TTBT_S01E04.mp4	0.9091	0.8333	0.8696	0.9917

Table 4.2: Precision, Recall, F-score and Accuracy on each video of the dataset

Table 4.3 contains F-scores of all four videos and average of them for all threshold values from -8 to -17. We present this table to state that static threshold generation is not helpful. Best result for each video is coming for different threshold values. This result emphasizes the need for dynamic thresholding.

Threshold Value	TTBT_S01E01	TTBT_S01E02	TTBT_S01E03	TTBT_S01E04	Average
-8	0.2	0.48	**0.8889**	0.6316	0.5501
-9	0.2857	0.5	0.8293	**0.8333**	0.6121
-10	0.56	**0.7272**	0.7556	0.6897	0.6831
-11	0.8	0.7059	0.7391	0.5641	**0.7023**
-12	0.8387	0.7059	0.7391	0.5238	0.7019
-13	0.875	0.7059	0.7083	0.5106	0.7
-14	0.9091	0.6316	0.68	0.4615	0.6706
-15	**0.9714**	0.6316	0.6667	0.3871	0.6642
-16	**0.9714**	0.5652	0.5484	0.3243	0.6023
-17	0.9444	0.5385	0.4857	0.2609	0.5574

Table 4.3: F-scores on the test videos in different threshold values from -8 to -17

Table 4.4 contains a comparison of our approach with[2] on the same dataset. Our method performed pretty well against the existing method.

Video	Precision		Recall		F-score		Accuracy	
	Our Method	[2]	Our Method	[2]	Our Method	[2]	Our Method	[2]
TTBT S01E01	**1.0**	0.0596	0.9444	**1.0**	**0.9714**	0.1125	**0.9973**	0.5697
TTBT S01E02	**0.5714**	0.0424	0.75	**0.875**	**0.6466**	0.0809	**0.9643**	0.5247
TTBT S01E03	**0.7273**	0.0464	0.9412	**1.0**	**0.8205**	0.0888	**0.9821**	0.5245
TTBT S01E04	**0.9091**	0.0351	0.8333	**1.0**	**0.8696**	0.0678	**0.9917**	0.5203
Overall	**0.802**	0.0459	0.8672	**0.9688**	**0.8275**	0.0875	**0.9839**	0.5348

Table 4.4: Comparison of our approach with existing method

Mel Frequency Cepstral Coefficients (MFCCs) are a widely used feature in automatic speech and speaker recognition. They were introduced by Davis and Mermelstein in the 1980's.
The MFCC coefficients mostly depend on the amplitude of each frequency bin in the spectrum. MFCCs describe the distribution of the amplitudes (not the phases) of the frequency bins, after transforming the frequency axis to a logarithmic-like scale (actually, the Mel scale). The lowest MFCC coefficient finds broader amplitude peaks in the spectrum, a high value of higher coefficients indicates a fluctuating distribution.
If a cepstral coefficient has a positive value, the majority of the spectral energy is concentrated in the low-frequency regions.
If a cepstral coefficient has a negative value, it represents that most of the spectral energy is concentrated at high frequencies.
So, High mfcc value suggests existence of more low-frequency components and low mfcc value suggests existence of more high-frequency components.

V. Conclusion

We designed an audio-only method for detection of video shots containing advertisements. We used dynamic thresholding instead of static thresholding because static thresholding led us to less accuracy and thus we thought that thresholding should depend upon the content of the test video. For some non-ad shot, our approach is giving a false positive. It is found out that those shots are too short in size (their length is around 1 second). Our approach needs some modification to overcome this phenomenon.

References

[1] Sadlier, D. A., Marlow, S., O'Connor, N., Murphy, N. (2002).Automatic TV advertisement detection from MPEG bitstream. PatternRecognition, 35(12), 2719-2726
[2] Ramires, A., Cocharro, D., Davies, M. E. (2018). An audio-only method for advertisement detection in broadcast television content. arXivpreprint arXiv:1811.02411
[3] Hossari, M., Dev, S., Nicholson, M., McCabe, K., Nautiyal, A., Conran, C., ... & Pitié, F. (2018). ADNet: A deep network for detecting adverts. *arXiv preprint arXiv:1811.04115*.
[4] Hussain, Z., Zhang, M., Zhang, X., Ye, K., Thomas, C., Agha, Z., ... & Kovashka, A. (2017). Automatic understanding of image and video advertisements. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 1705-1715).
[5] Logan, B. (2000, October). Mel frequency cepstral coefficients for music modeling. In *Ismir* (Vol. 270, pp. 1-11).