A Novel Method to Classify Videos Based VBR Trace

Xiaojie Yu, Min Qiu, and Lizhi Peng(✉)

Shandong Provincial Key Laboratory of Network Based Intelligent Computing,
University of Jinan, Jinan 250022, People's Republic of China
yuxiaojie814@gmail.com, plz@ujn.edu.cn

Abstract. Video classification research has been studied for many years. Traditional video classification methods are based on text, sound, and visual content. However, all these approaches require that the video content can be inspected. If the video content can not be investigated. For example, the video frame is encrypted or transmitted on the network device, then we can only measure the size of the video frame bitrate. In this paper, we propose two novel feature extraction methods based on variable bit rate (VBR) trace. The first one is extracting features in sliding windows. The second one is based on change points techniques to obtain more reasonable windows. We carry out empirical studied on our data sets to discriminate the action videos from the other videos. The experiment shows that we can identify the action video with 87% g-mean.

Keywords: Video classification · VBR trace · Change points.

1 Introduction

In the past decade, Internet witnessed the burst of video, especially for the mobile Internet. All kinds of videos extremely enhanced user experiences. However, different video contents have posed a new challenge, that is how to identify the Internet video types. It is necessary for large video sites such as Netflix, YouTube, and Amazon to classify their videos to provide high quality video services. Schools need to ensure that students are exposed to health videos. From the view of Internet management, it is necessary to pick out illegal videos from other videos.

Traditional video classification is generally divided into four ways: text-based approaches, audio-based approaches, visual-based approaches, and those that used some combination of text, audio, and visual features [3,7,11,15,20,21,23]. Many of the standard classifiers such as Bayesian, support vector machines

This research was partially supported by the National Natural Science Foundation of China under grant No. 61972176, No. 61472164, No. 61672262, No. 61572230, and No. 61573166, the Shandong Provincial Key R&D Program under Grants No. 2018CXGC0706 and No. 2017CXZC1206.

X. Chen et al. (Eds.): ML4CS 2020, LNCS 12487, pp. 270–282, 2020.
https://doi.org/10.1007/978-3-030-62460-6_24

(SVM) can be used for video classification. Gaussian mixture models (GMMs) and hidden Markov models (HMMs) are particularly popular on video classification in the past few years [3]. Much progress has been made in video classification in recent years. Convolutional Neural Networks (CNNs) has been proven to perform very well on video classification tasks [4,12,13]. However, all these techniques are based on condition that video contents can be inspected. If the video frame is encrypted or transmitted on a network device, then we can only measure the size of the video frame. In such cases, all these traditional techniques are invalid as the video contents cannot be inspected. Hence, in this work, we explore the method of time series classification based on video frame size.

Variable Bit Rate (VBR) Trace. Most popular streaming services use variable bitrate encoding. Therefore, the bitrate of an encoded video varies with its content. Variable bit rate encoding is also used on H.264 video. H.264 is the most widely used encoding standard for Internet video. In this standard, a video is encoded into a series of consecutive GOP (Group of pictures) groups. For each GOP, there is one I frame, several B frames and P frames. The I frame (intra coded picture) reference image is equivalent to a fixed image and is independent of other image types. Each image group starts with an I frame. A P frame (predictive coded picture) contains the difference information from the previous I or P frame. B frames (bidirectionally predictive coded pictures) contain difference information from previous and/or subsequent I or P-frames. VBR allows a higher bitrate to build more complex segment of media files while less space is allocated to less complex segments. Hence, the frame size changes with the bit rate. In this work, we choose B frame size traces as VBR traces.

The main contributions of this paper are summarized as follows:

- To the best of our knowledge, this paper is the first to use the change point method on the video frame size traces to explore the scene classification. We propose a new fusion function to obtain change points more accurately.
- We extract statistical features from variable bit rate (VBR) trace, resulting a larger and more effective feature set.
- We verified the effectiveness of the extracted features on our own data set, and further explored the impact of different parameters on classification.

The remainder of the paper is structured as follows. First, we introduce the background about VBR trace in Sect. 2. Then, we outline the releted work in Sect. 3. Next, we present the framework in this work in Sect. 4. In Sect. 5, we describe in detail the method of extracting features. Implementation details and experimental results are described in Sect. 6. Discussion and future work are provided in Sect. 7.

2 Related Work

In general, the classification and matching of videos are mainly studied in two fields, one is the field of computer vision, and the other is the field of network security.

In the field of computer vision, there are many studies on video classification [4,12,13], mostly based on deep learning methods. They basically pay attention to the recognition of various actions. On the other hand, there are also research and explorations on video classification using Zero-shot learning [1,2,10,24].

In the field of cybersecurity, R Schuster et al. [22] showed that due to the segmentation prescribed by the MPEG-DASH standard, many video streams are uniquely characterized by their burst patterns. R Dubin et al. [6] and J Gu et al. [9] also explored the burst patterns to identify encrypted video streams. For the same reason, H Li et al. [16] and X Liu et al. [18] explored the action recognition on surveillance traffic.

Last but not least, FHP Fitzek et al. [8] present a publicly available library of frame szie traces of long MPEG-4 and H.263 encoded videos. They also present a thorough statistical analysis of the traces. Q Liang et al. [17] used fuzzy techniques to model and classify MPEG VBR videos.

Inspired by all these efforts, we explore a novel method to classify H.264 encoded videos. We extract features of VBR traces in following sections. And further present the effect of different parameters and feature combinations on the results.

3 The Framework

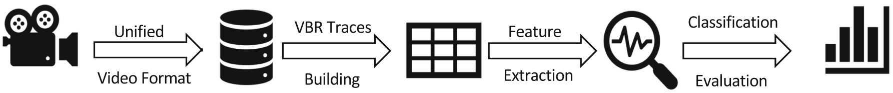


Fig. 1. Framework with four steps

3.1 A. Video Pre-processing

We first convert the videos with different formats to a single format using Axiom [19]. The target parameters are shown in Table 1.

Table 1. Format parameter

Video codec	Encode speed	HW accel	Quality	Pass	Pixel format	Frame rate
x264	Medium	off	High	2Pass	yuv420p	24

Then each movie is split into multi segments with fixed length of 120 s. And then, we pick out all the segments with actions to build the action movie set, and the left ones for the other movie set.

3.2 B. VBR Traces Building

The VBR data are extracted from the movie sets using FFmpeg, as VBR is the most effective method to get video information without inspecting the video contents. As we know, B frame is the dominant frame type in video data. Most differences between frames are contained in B frame. Therefore we only use the VBR trace of B frame. Each VBR trace is an array $D_i = (t_1, t_2, \dots, t_n)$, thus all arrays with label form a vector sets $D = \{D_1, D_2, \dots, D_m\}$. Obviously, the length of each row in vector sets is not same because the number of each segment is not equal.

3.3 C. Feature Extraction

Due to the high dimensional raw data and the varying length of the VBR traces, it is necessary to extract high-level semantic features to reduce the computing complexity and to achieve high classification performance. Inspired by [18], a basic idea is to calculate trace rate change C for each row data $D_i = (t_1, t_2, t_3, \dots, t_n)$, where C is defined in Eq. 1.

As each VBR trace is essentially a time serial. We use a window sliding on each VBR trace to extract windowed-features. Then, some statistics are got from each window as the additional features. Detailed techniques will be introduced in the next section.

3.4 D. Classification

At the final step we utilize machine learning algorithms to discriminate the action movies from the other movies. To validate the effectiveness of the VBR trace and its features, we carry out our empirical studies using six well-known classic machine learning algorithms: Random Forests (RF), Logistic Regression (LR), Gaussian Naive Bayes (GNB), Decision Tree (DT), linear Suppor Vector Classification (LinearSVC), SVM with rbf kernel (SVM-rbf).

4 Extract Features

In this section, two novel feature extraction methods will be illustrated in detail. The first one is inspired by the method which is proposed in [18]. The second one is based on the change point detection techniques in statistics [14,16]. Features we extracted are shown in Table 2.

In order to capture the information of VBR rate change, we first calculate the VBR trace rate change $C_i = (a_1, a_2, a_3, \dots, a_{n-1})$ of the ith VBR trace $D_i = (t_1, t_2, \dots, t_n)$. a_j represents the difference in frame size between the $(j+1)$-th and the j-th frame.

$$a_j = t_{j+1} - t_j, \qquad j \in [1, n-1] \tag{1}$$

Table 2. Features

Data type	Features
VBR trace	Mean, variance, skewness, kurtosis
Rate change	Mean, variance, skewness, kurtosis
DFT	Amplitude, phase

The mean values of the VBR trace and the frame size rate change: These features can show the intensity of scene changes in the videos. Given a VBR trace $D_i = (t_1, t_2, t_3, \ldots, t_n)$ and $C_i = (a_1, a_2, a_3, \ldots, a_{n-1})$, the mean values $\overline{t}$ and $\overline{a}$ are defined as:

$$\overline{t} = \frac{1}{n}\sum_{i=1}^{n} t_i, \qquad \overline{a} = \frac{1}{n-1}\sum_{i=1}^{n-1} a_i \tag{2}$$

The variances of the VBR trace and the frame size rate change: These features can show the complexity of scene change. Given a frame size traces $D_i = (t_1, t_2, t_3, \ldots, t_n)$ and $C_i = (a_1, a_2, a_3, \ldots, a_{n-1})$, the variance t^{var} and a^{var} are defined as:

$$\begin{aligned} t^{var} &= \frac{1}{n}\sum_{i=1}^{n}(t_i - \overline{t})^2 \\ a^{var} &= \frac{1}{n-1}\sum_{i=1}^{n-1}(a_i - \overline{a})^2 \end{aligned} \tag{3}$$

The skewness of the VBR trace and the frame size rate change: These features describe the symmetry of data distribution. Given a frame size traces $D_i = (t_1, t_2, t_3, \ldots, t_n)$ and $C_i = (a_1, a_2, a_3, \ldots, a_{n-1})$, the skewness t^{sk} and a^{sk} are defined as:

$$\begin{aligned} t^{sk} &= \frac{\frac{1}{n}\sum_{i=1}^{n}(t_i - \overline{t})^3}{(\frac{1}{n-1}\sum_{i=1}^{n}(t_i - \overline{t})^2)^{\frac{3}{2}}} \\ a^{sk} &= \frac{\frac{1}{n-1}\sum_{i=1}^{n-1}(a_i - \overline{a})^3}{(\frac{1}{n-2}\sum_{i=1}^{n-1}(a_i - \overline{a})^2)^{\frac{3}{2}}} \end{aligned} \tag{4}$$

The kurtosis of the VBR trace and the frame size rate change: These features describe the shapes of the distribution of the original data. Kurtosis is a measure of whether the distribution is peaked or flat relative to a normal distribution. Given a frame size traces $D_i = (t_1, t_2, t_3, \ldots, t_n)$ and $C_i = (a_1, a_2, a_3, \ldots, a_{n-1})$, the kurtosis t^{ku} and a^{ku} are defined as:

$$\begin{aligned} t^{ku} &= \frac{\frac{1}{n}\sum_{i=1}^{n}(t_i-\overline{t})^4}{(\frac{1}{n}\sum_{i=1}^{n}(t_i-\overline{t})^2)^2}-3 \\ a^{ku} &= \frac{\frac{1}{n-1}\sum_{i=1}^{n-1}(a_i-\overline{a})^4}{(\frac{1}{n-1}\sum_{i=1}^{n-1}(a_i-\overline{a})^2)^2}-3 \end{aligned} \tag{5}$$

Amplitude and Phase transformed from DFT: According to reference [18], we use DFT to obtain coefficients containing frequency information. We apply DFT in sliding window directly. Given a frame size traces $D_i = (t_1, t_2, t_3, \dots, t_n)$, the coeffcients we get are complex numbers with the form of $z = a + bi$. Amplitude and phase were proven to be effective in practice [18]. Thus, we use these features. Amplitude and phase are defined as:

$$Amplitude = \sqrt{a^2+b^2}, \qquad Phase = \arctan\frac{b}{a} \tag{6}$$

Sliding window: For each frame size traces $D_i = (t_1, t_2, t_3, \dots, t_n)$, sliding window may get more detailed information. We do not fix the size of the windows, but the number of windows: m is fixed. A consequent issue is the impact of the parameter m, which will be explored in the empirical studies. Furthermore, we explored the effect of different m. For example, given a frame size traces $D_i = (t_1, t_2, t_3, \dots, t_n)$, result is $D_i = (d_1, d_2, \dots, d_j, \dots, d_m)$, d_j is sub-sequence. The sub-sequece d_j is defined as:

$$d_j = \begin{cases} \{t_{[(j-1)\frac{n}{m}]}, \dots, t_{j\frac{n}{m}}\}, & 1 \le j \le (m-1) \\ \{t_{[(j-1)\frac{n}{m}]}, \dots, t_n\}, & j = m \end{cases} \tag{7}$$

Since the length of a scene in each video is not fixed, fixed length of sliding window is not reasonable. The perfect case is that a single window corresponds with a single scene. Therefore, we apply PELT algorithm [14] first to detect change points. Then, we get more reasonable length of window.

Change point detection for a time series data $y_{1:n}$, assume we get m change points with their positions $\tau = \{\tau_1, \tau_2, \dots, \tau_m\}$ in $y_{1:n}$. Let $\tau_0 = 0$ and $\tau_{m+1} = n$. One commonly used method to identify multiple change points is to minimize

$$\sum_{i=1}^{m+1}[\mathcal{C}(y_{(\tau_{i-1}+1):\tau_i})] + \beta f(m) \tag{8}$$

Here $\mathcal{C}$ is a cost function for a segment and $\beta f(m)$ is a penalty guard against overfitting.

We present a weighted fusion cost function which combines cost function $\mathcal{C}_1$ for exponential distribution with changing mean and cost function $\mathcal{C}_2$ for normal distribution with variable variance. More formally, for a segmented subsequence

$y_{1:n}$ between $\tau_{i-1}+1$ and τ_i, $n = \tau_i - (\tau_{i-1}+1)$, we have

$$\mathcal{C}_1 = -n(\log(\sum_{j=1}^{n} y_j)) \tag{9}$$

$$\mathcal{C}_2 = n\log\sigma^2 + \frac{1}{\sigma^2}\sum_{j=1}^{n}(y_j-\mu)^2 \tag{10}$$

Here,

$$\sigma^2 = \frac{1}{n}\sum_{j=1}^{n}(y_j-\bar{y}) \tag{11}$$

$$\mu = \frac{1}{n}\sum_{j=1}^{n} y_j \tag{12}$$

Figure 2 shows a case study of comparing the methods using $\mathcal{C}_1$,$\mathcal{C}_2$ and the combined cost function. As shown in Fig. 2, the distribution of the change points obtained by the cost function $\mathcal{C}_2$ is relatively dense. The distribution of the change points obtained by the cost function $\mathcal{C}_1$ is relatively sparse. Hence, we use cost function $\mathcal{C} = \theta_1\mathcal{C}_1 + \theta_2\mathcal{C}_2$. In our study, we set the parameters as $\theta_1 = 0.7, \theta_2 = 0.3$. We also carry out empirical studies to explore the impacts on these parameters.

Another problem is that some of the neighbour change points are too close to support reasonable segments. For the time of two change points T_{τ_i} and $T_{\tau_{i-1}}$, if $T_{\tau_i} - T_{\tau_{i-1}} > 0.5$ s, then we ignore τ_i.

5 Evaluation

5.1 A. Date Collection

Table 3. Collection of videos

Movies	Episodes
MIT 18.065	All
Avengers	2, 3
Transformers	1, 2, 3, 4, 5
Iron Man	1, 3
Pirates of the Caribbean	1, 2, 3, 4, 5

Collection of video is shown in Table 3. We first convert the videos with the parameters in Table 1 using [19]. Then each movie is split into multi segments

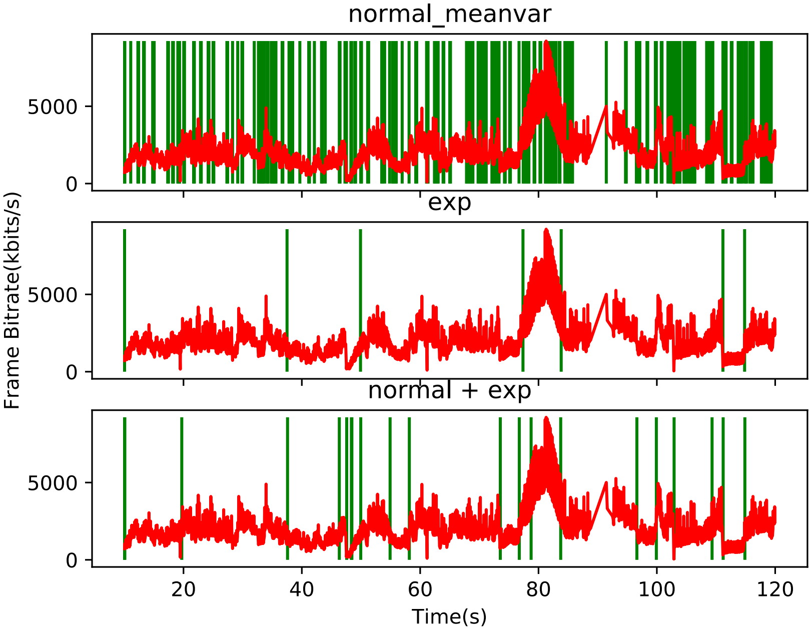


Fig. 2. The change points obtained by different functions. The first line and the second line are the results of normal distribution and exponential distribution, the last line is the result of the fusion of the two functions.

with fixed length of 120 s. And then we extract B frame traces on each segment. Finally, we got 2144 samples including 379 action scene samples and 1765 other scene samples. Obviously, the data sets is imbalanced, we over-sample minority classes by the Synthetic Minority Oversampling Technique (SMOTE) [5]. In this work, we use cross validation method to evaluate the performance of extracted features, so we split samples to 5 folds randomly.

5.2 B. Experimental Setup

We use six model to evaluate the effective of features. Firstly, we test on features which is extracted by fixed windows. As the Table 4 shown.

We can evaluate the effect of different number of windows. Secondly, for the features extracted by change points method, we choose a best model in last step to evaluate the performance of those features. We arrange the features between the change points τ_i and τ_{i+1} as $(time\ span, mean, variance, skewness, kurtosis)$, and define them as features in an interval, called span features. Then different combinations of features are tested according to Table 5. We also use Wilcoxon's Sign Rank Test to test the difference between different features.

Table 4. Combinations of features

Features	1window	1window-DFT	9windows-DFT
(VBR trace) mean, variance, skewness, kurtosis	✓	✓	✓
(rate change) mean, variance, skewness, kurtosis	✓	✓	✓
(DFT) amplitude, phase	_	✓	✓

Table 5. Different combinations of features

Features	V1	V2	V3
span features	✓	✓	✓
(1 window) mean, variance, skewness, kurtosis	_	✓	✓
(rate change features of 1 window) mean, variance, skewness, kurtosis	_	_	✓

5.3 C. Performance Metrics

We use G-mean, Accuracy, F1-score to evaluate the performance of our features.

G-mean. We define g-mean as

$$G - mean = \sqrt{\frac{TP}{TP + FN} \times \frac{TN}{TN + FP}} \tag{13}$$

where TP and FP represent the true postives and the false positives of samples, respectively. Besides, TN and FN represent the true negative and false negative of samples, respectively.

Accuracy. we define accuracy as

$$accuracy = \frac{TP + TN}{TP + TN + FN + FP} \tag{14}$$

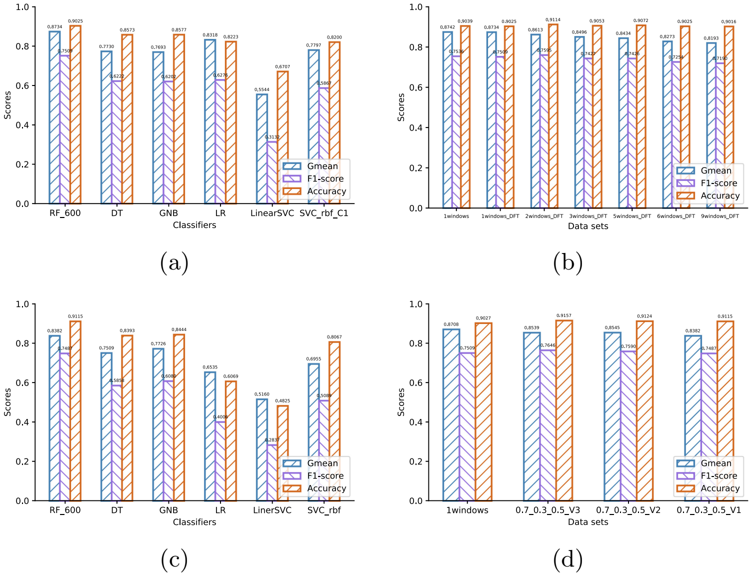
F1-score. we define f1-score as

$$f1 - score = \frac{2TP}{2TP + FP + FN} \tag{15}$$

5.4 D. Experimental Results

In this part, we carry out experiments to evaluate the effectiveness of features. As the Fig. 3(a) shown, when the number of windows m is 1, the performance of random forest is best. The performance of svm algorithm with rbf kernel is better than that with linear kernel. The results show that our data requires a nonlinear method to fit. And random forest with 600 trees have more powerful fitness.

As the Fig. 3(b) shown, we experiment on features obtained with different window numbers by using random forest model. As the number of windows m increases, the g-mean score and f1-score score decreases. The best g-mean score, g-mean score of feature $1windows\_DFT$, is higher 5.41% than feature



(a)

(b)

(c)

(d)

Fig. 3. The impact of different features on different models. (a) is the performance of features on six models when windows $m = 1$; (b) is the performance of the features obtained by the different numbers of windows on the random forest; (c) is the performance of Feature V1 on six models; (d) is the performance of the combination of several features of the two methods. Here, $0.7\_0.3\_0.5\_V3$ means $\theta_1 = 0.7$ and $\theta_2 = 0.3$, the span is 0.5 s, so we can ignore some change points as mentioned above.

$9windows\_DFT$'s. We guess that the fewer windows, the larger the window size, which can capture more macro fluctuation characteristics. Another phenomenon is that there is no significant difference between those with Fourier transform features and those without Fourier transform features. In theory, we believe that selecting the appropriate frequency domain characteristics can better reflect the fluctuation characteristics of the data. We guess the reason for this result may be that our method of selecting frequency domain features is not suitable.

As the Fig. 3(c) and (d) shown, for the features extracted by changepoints methods, The performance of random forest also is best. Decision trees and Gaussian Naive Bayes perform well. In the random forest experiment, there are no obvious differences between the three combinations of feature points based on change points method. And compared with the features extracted with only one window, the method of change points does not remind of obvious advantages. In theory, we think that the change points method is more reasonable, and the problem may lie in our treatment of features. Since the change point of each sample is different, the number of extracted features is different. In order to obtain training samples of equal length, we fill in the zeros behind the features, which causes a lot of information redundancy, so that the classifier does not get good results.

We conduct experiments on the change points data obtained by different combinations of θ_1 and θ_2. As the Fig. 4 and Fig. 5 shown, when $\theta_1 = 0.4$ and $\theta_2 = 0.6$, the result is the best one. And f1-score and accuracy, the result of feature 0.4_0.6_$V2$ is a little higher than the result of feature $1window$.

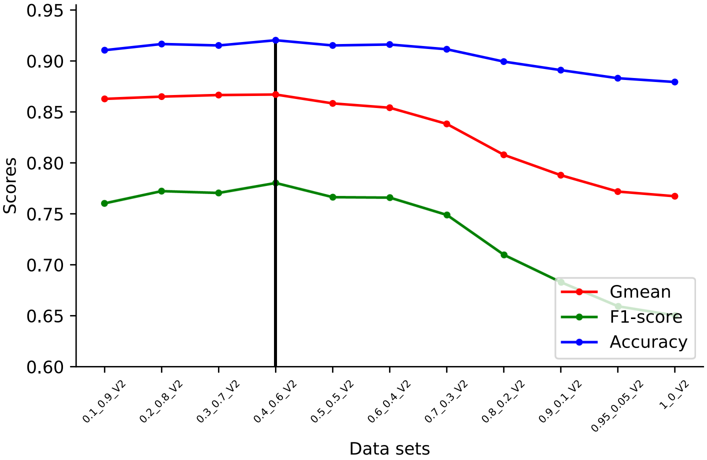


Fig. 4. Results on various selection of θ_1 and θ_2. For example, 0.1_0.9_$V2$ means $\theta_1 = 0.1$ and $\theta_2 = 0.9$. We obtain the Feature V2 to test.

In addition, through Wilcoxon's Sign Rank Test, we compared the differences between different feature combinations. Firstly, we do hypothesis testing on the best features of the two ideas. Secondly, we do hypothesis testing for different combinations of features in each idea. The result is shown in Table 6.

Table 6. The results of Wilcoxon's Sign Rank Test

Test Group	P-value	Result
(1window-DFT, v2)	0.0938	0
(1window-DFT, 1windows)	0.8438	0
(1window-DFT, 9windows-DFT)	0.4375	0
(V1, V3)	0.0313	1
(V1, V2)	0.0625	0
(V2, V3)	0.0625	0

The result of hypothesis testing show that there is no significant difference in performance between most of the features of our experiments. But for the best model in our experiments, such as random forest, the difference in different features is still obvious.

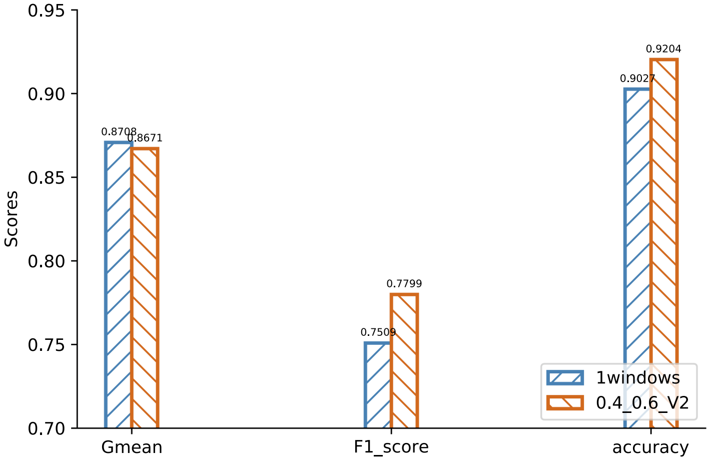


Fig. 5. Comparison between Feature 1window and Feature 0.4_0.6_$V2$

6 Conclusion and Future Research

In this paper, we explored a novel method to classifier videos. The VBR trace can show some semantic features which is useful to classification. Further, we explored the more effective features obtained by segmenting data using change points method. Basically, our features are effective in the binary classification of videos. But the result of the hypothesis test is not what we thought. We realize that there are still many unsolved things. We need to solve the problem of zero-filling of changing points. Maybe a faster and more effective change point algorithm can be used. The fusion of the objective function of the change point and the corresponding weight have a large adjustment space, etc. Finally, in the future, we can try the multi-classification task.

References

1. Bishay, M., Zoumpourlis, G., Patras, I.: Tarn: temporal attentive relation network for few-shot and zero-shot action recognition. arXiv preprint arXiv:1907.09021 (2019)
2. Brattoli, B., Tighe, J., Zhdanov, F., Perona, P., Chalupka, K.: Rethinking zero-shot video classification: end-to-end training for realistic applications. arXiv preprint arXiv:2003.01455 (2020)
3. Brezeale, D., Cook, D.J.: Automatic video classification: a survey of the literature. IEEE Trans. Syst. Man Cyber. Part C (Appl. Rev.) **38**(3), 416–430 (2008)
4. Caba Heilbron, F., Escorcia, V., Ghanem, B., Carlos Niebles, J.: ActivityNet: a large-scale video benchmark for human activity understanding. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 961–970 (2015)
5. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: SMOTE: synthetic minority over-sampling technique. J. Artif. Intell. Res. **16**, 321–357 (2002)

6. Dubin, R., Dvir, A., Pele, O., Hadar, O.: I know what you saw last minute-encrypted HTTP adaptive video streaming title classification. IEEE Trans. Inf. Forensics Secur. **12**(12), 3039–3049 (2017)
7. Fan, J., Luo, H., Xiao, J., Wu, L.: Semantic video classification and feature subset selection under context and concept uncertainty. In: Proceedings of the 2004 Joint ACM/IEEE Conference on Digital Libraries, pp. 192–201. IEEE (2004)
8. Fitzek, F.H., Reisslein, M.: MPEG-4 and H. 263 video traces for network performance evaluation. IEEE Netw. **15**(6), 40–54 (2001)
9. Gu, J., Wang, J., Yu, Z., Shen, K.: Walls have ears: traffic-based side-channel attack in video streaming. In: IEEE INFOCOM 2018-IEEE Conference on Computer Communications, pp. 1538–1546. IEEE (2018)
10. Hahn, M., Silva, A., Rehg, J.M.: Action2Vec: a crossmodal embedding approach to action learning. arXiv preprint arXiv:1901.00484 (2019)
11. Hauptmann, A., et al.: Video classification and retrieval with the informedia digital video library system (2002)
12. Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., Fei-Fei, L.: Large-scale video classification with convolutional neural networks. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2014
13. Kay, W., et al.: The kinetics human action video dataset. arXiv preprint arXiv:1705.06950 (2017)
14. Killick, R., Fearnhead, P., Eckley, I.A.: Optimal detection of changepoints with a linear computational cost. J. Am. Stat. Assoc. **107**(500), 1590–1598 (2012)
15. Kobla, V., DeMenthon, D., Doermann, D.S.: Identifying sports videos using replay, text, and camera motion features. In: Storage and Retrieval for Media Databases 2000, vol. 3972, pp. 332–343. International Society for Optics and Photonics (1999)
16. Li, H., He, Y., Sun, L., Cheng, X., Yu, J.: Side-channel information leakage of encrypted video stream in video surveillance systems. In: IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications, pp. 1–9. IEEE (2016)
17. Liang, Q., Mendel, J.M.: Mpeg VBR video traffic modeling and classification using fuzzy technique. IEEE Trans. Fuzzy Syst. **9**(1), 183–193 (2001)
18. Liu, X., Wang, J., Yang, Y., Cao, Z., Xiong, G., Xia, W.: Inferring behaviors via encrypted video surveillance traffic by machine learning. In: 2019 IEEE 21st International Conference on High Performance Computing and Communications; IEEE 17th International Conference on Smart City; IEEE 5th International Conference on Data Science and Systems (HPCC/SmartCity/DSS), pp. 273–280. IEEE (2019)
19. McManis, M.: Axiom: an FFmpeg interface for windows. https://axiomui.github.io
20. Roach, M., Mason, J., Xu, L.Q.: Video genre verification using both acoustic and visual modes. In: 2002 IEEE Workshop on Multimedia Signal Processing, pp. 157–160. IEEE (2002)
21. Roach, M., Mason, J.S.: Classification of video genre using audio. In: Seventh European Conference on Speech Communication and Technology (2001)
22. Schuster, R., Shmatikov, V., Tromer, E.: Beauty and the burst: remote identification of encrypted video streams. In: 26th {USENIX} Security Symposium ({USENIX} Security 17), pp. 1357–1374 (2017)
23. Xu, L.Q., Li, Y.: Video classification using spatial-temporal features and PCA. In: 2003 International Conference on Multimedia and Expo. ICME 2003, Proceedings (Cat. No. 03TH8698), vol. 3, pp. III-485. IEEE (2003)
24. Zhu, Y., Long, Y., Guan, Y., Newsam, S., Shao, L.: Towards universal representation for unseen action recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 9436–9445 (2018)