Multimed Tools Appl
https://doi.org/10.1007/s11042-018-5943-3

Embedded learning for computerized production of movie trailers

Jiachuan Sheng[1] · **Yaqi Chen**[1] · **Yuzhi Li**[1] · **Liang Li**[2]

Received: 15 December 2017 / Revised: 19 March 2018 / Accepted: 26 March 2018

Abstract Movie trailers are usually extracted from the most exciting, interesting, or other noteworthy parts of the movies in order to attract the audience and persuade them to see the film. At present, hand-crafted movie trailers currently occupy almost all the filming market, which is costly and time-consuming. In this paper, we propose an embedded learning algorithm to generate movie trailers automatically without human interventions. Firstly, we use CNN to extract features of candidate frames from the film by a rank-tracing technique. Secondly, SURF algorithm is utilized to match the frames of the movie with the corresponding trailer, thus the labeled and unlabeled dataset are prepared. Thirdly, the mutual information theory is introduced into the embedded machine learning to formulate a new embedded classification algorithm and hence characterize similar key elements of the trailers. Finally, semi-supervised support vector machine is applied as the classifier to obtain the satisfactory key frames to produce the predicted trailers. By treating several famous movies and their manual handling trailers as the ground-truth, series of experiments are carried out, which indicate that our method is feasible and competitive, providing a good potential for promoting the rapid development of the film industry in terms of publicity, as well as providing users with possible solutions for filtering large amounts of Internet videos.

✉ Liang Li
liangli@tju.edu.cn

Jiachuan Sheng
jiachuansheng@tjufe.edu.cn

Yaqi Chen
yaqichen@stu.tjufe.edu.cn

Yuzhi Li
liyuzhi@tjufe.edu.cn

[1] Department of Computer Science, Tianjin University of Finance & Economics, Tianjin 300222, China

[2] School of Computer Software, Tianjin University, Tianjin 300350, China

Published online: 24 April 2018

Keywords Production of movie trailers · Embedded learning · Video summarization · CNN-based feature extraction · Movie key frames

1 Introduction

Along with the rapid development of Internet and multimedia technology, movies occupy an increasingly dominant position among multimedia consumers. Movies production is no longer the privilege of professionals in the meantime, anybody can make a film by using software or even the App on the phone and then share their own movies to others. Massive movies make it hard to find the one that's right for them. Facing the huge resources of multimedia big data, how to quickly and accurately find a movie that is satisfying to the interests has become a significant research challenge. To solve this problem, producers often create movie trailers, which normally contains some of the most exciting, funny, or otherwise noteworthy parts of the movie to attract attentions from the public. As viewers can make better decisions whether the movie is worth watching or not based on the content of the trailer, trailers play a significant role in providing information and guidelines in this selective process. However, manual production of trailers requires a lot of human and financial resources, which has to be limited to those professionally produced movies, and little efforts are ever made to generate trailers for those massively produced movies by amateurs. To this end, there is an urgent demand for automatically generating trailers for the filming industry.

Generating a trailer is one aspect of video summarization, the basic idea of which is to analyze the contents of the video, filter important video clips, and finally combine the selected video clips into a summary video. Video summarization approaches can be categorized into two major groups: static and dynamic video summary [8, 14, 18]. The former summarizes a video by filtering frames with important video information and optimizing its diversity or representativeness, while the latter is to process the contents of the original video stream (including audio and motion content), and compile a set of video shots which present the most important and interesting contents by reducing spatial and temporal redundancy in the video [31].

Movie trailers is a type of video summary, but still significantly different from other video summarization techniques. Generating a movie trailer is a process that collects the factors of commercial and artistic, and its purpose is to advertise to attract audience. Therefore, the trailer is more necessary to express the film in the special highlights and create an artistic atmosphere. A high-quality trailer usually contains the following five attributes [23]: (1) protagonists and key objects; (2) prominent storyline; (3) the overall artistic atmosphere; (4) dialogue scenes; (5) creating suspense by avoiding the story ending. According to the above analysis we can see that the key frames in trailers generally have the same characteristics.

In summary, our contributions can be highlighted as follows: (i) unlike manually-made trailers, our proposed can produce movie trailers automatically without any human intervention; (ii) unlike the hand-crafted feature extraction, we introduce CNN-based deep learning unit to describe the common feature of trailers; (iii) we propose an improved embedded classification algorithm to make the classification, by which selection of features and their level of importance are directly integrated with the embedding process to implement the principle of friends being closer and enemies being apart.

The rest of the paper is organized into four sections, where section 2 introduces the related work of generating movie trailers in terms of extracting key frames, section 3 reports our algorithm, section 4 reports the experimental results and evaluations, and section 5 provides concluding remarks.

2 Related work

Most previous works have been well studied on video summarization. Li et al. [14] proposed a new shot boundary detection algorithm, which combined both global and local feature representations and utilized sparse coding for shot boundary detection. Then, a series of frames are selected to represent the content of a wide variety of shots, which are key frames of the video summary. The resulting video summary can represent the video content in its entirety, but may not highlight the features of the video. Zhang et al. [35] proposed a supervised learning technique for video summarization, which was transferred summary structures from training videos to test ones by learning nonparametrically. The method worked well only for videos that are very similar in structure and content. Otani et al. [22] segmented the video based on its relevance to the input text, and its priority in each digest was specified in each video segment. Sun et al. [28] used barrage comments to select the most-reviewed candidate highlight, then scored the clips based on the content and the number of the bullet screen comments. It requires user generated data, namely the bullet screen notes in the calculation. Movie trailers have their own common nature, to this end, the algorithms of video summarization do not fit well with the generation of the movie trailers.

Existing studies have limitations in characterizing the distinctive attributes of trailers. We can preprocess the dataset by embedding the frames into the lower dimensional embedded space that captures the intrinsic features of the movies. Roweis et al. [24] and Tenenbaum et al. [29] proposed an embedded learning algorithm, which can be divided into two categories: unsupervised machine learning and embedding supervised techniques. The former provides just a compact and informative representation of the data, and the latter is to separate the sets of intra-class and inter-class points that are close to each other in the embedded space, ultimately increasing the classification accuracy between classes [7, 9, 15, 19, 33]. Unfortunately, the algorithm ignores the relationship between samples, and the effectiveness is strongly dependent on the inherent characteristics of a particular dataset in general.

To improve the embedded learning, Martinez et al. [20] and Mu et al. [21] proposed an improved framework DEFC (data embedding framework for classification). Instead of fixing the embedding generation model and then train it with a given dataset, it attempts to generate a model that is optimally suitable to the given dataset, without any requirement from the users for manual settings. In principle, frames inside the movie trailers show a stronger visual impact than other frames, with some of the features making a more significant contribution. In the process of computation of the similarity between frames, however, it does not consider the influence of the feature importance on the similarity feature calculation, which leads to the restriction of the classification performances. To solve this problem, we use VGG-F model to extract the characteristics of movie frames, and introduce a new mutual information measurement into the embedded learning. To this end, the embedded principle that friends being closer and enemies being apart can be achieved by feature selection according to their influence.

3 Movie trailer generation via embedded learning

Figure 1 shows the overview of our proposed system, where a number of well-known movies with their official trailers are used as the training dataset and another part of movies without trailers as the test dataset. As shown in Fig. 1(a) and (d), by using rank tracing algorithm [1], the candidate frames can be extracted from training and test dataset respectively. Based on the

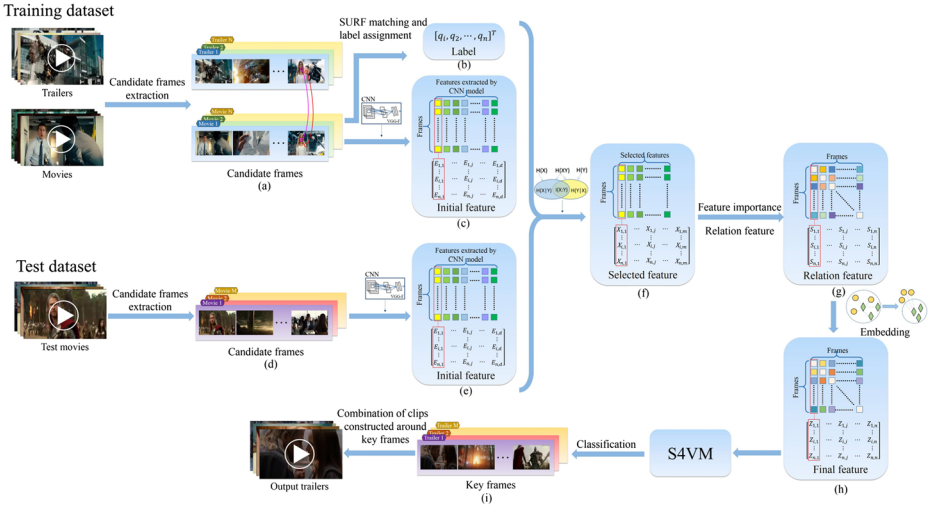


Fig. 1 Overview of the proposed embedded semi-supervised learning approach

SURF (Speeded Up Robust Features) algorithm [3], as seen in Fig. 1(b), the matching process is carried out to seek the key frames which resemble those inside the manually-made trailers, and the key frames are labeled as positive, while the rest of frames are labeled as negative. In this regard, if the *ith* frame of the movie can match a frame inside the trailer, its label is assigned as: $q_i = +1$, and otherwise we have: $q_i = -1$. Figure 1(c) and (e) are feature matrices which are obtained by extracting the features of candidate frames by the CNN model, where E_{ij} represents the *jth* feature of the *ith* frame and different colors represent different features. Figure 1(f) shows the selected features by calculating the mutual information between features and labels, where X_{ij} represents the *jth* feature of the *ith* frame. Figure 1(g) illustrates the relational feature between frames under the influence of feature importance, where S_{ij} represents similarity between the *jth* and the *ith* frames, different colors representing different similarity between the frames. Figure 1(h) shows similar features between frames by using embedded mapping, where Z_{ij} is the embedded mapping matrix of S_{ij}. Comparing Fig. 1(g) with Fig. 1(h), the representative block color is changed, where a darker color indicates a decrease in similarity. In the embedded mapping process, the circle and the diamond represent two different classes of frames, respectively. The original space (left) is transformed to the embedded one (right), where the friends are pulled closer, while its enemies are pushed apart. As shown in Fig. 1(i) after S4VM algorithm classification, we obtain the label matrix q'_j of the candidate frames from the test movies, whose $q'_j = +1$ will be utilized as the key frame. An output trailer can be automatically produced by using a combination of clips constructed around the key frames.

In summary, our system can be divided into three steps: data preprocessing, semi-supervised embedded learning, and output Trailers. Each step of the method is explained as follows.

3.1 Data preprocessing

In order to prepare the dataset for the classification of frames, we need to get the appropriate features of the candidate frames. Firstly, we extract the candidate frames from the movies for

the official trailers. Secondly, we extract CNN features of the frames. Thirdly, we prepare the labeled and unlabeled dataset.

3.1.1 Candidate frame extraction

All videos have a common property that dynamic effects can be displayed by fast playback, whilst the content of adjacent frames is usually similar. Therefore, it is feasible to use a frame to represent a shot with similar content. In order to process the video more efficiently, we detect shot boundaries and extracts candidate frames from the video sequences of the official movie trailers and films [1, 25, 26]. It is robust to a wide range of digital effects when the camera shot is changed.

Let x^t represents the feature vector of the frame at time t, it is defined as follows:

$$\mathrm{x}^t = [h_H\ h_S\ h_V], \tag{1}$$

where h_H, h_S, h_V are the histograms of Hue-Saturation-Value color space respectively. Define m_H, m_S, m_V, respectively, as their length, the dimension of x^t can be derived as: $M = m_H + m_S + m_V$.

Let N represents the time of a window, the feature matrix can be produced as follows:

$$X^t = \begin{bmatrix} \mathrm{x}^t \\ \mathrm{x}^{t-1} \\ \vdots \\ \mathrm{x}^{t-N+1} \end{bmatrix}, t = N, \cdots, T. \tag{2}$$

Obviously, X^t consists of feature vectors in a window. T represents the number of frames. Next, do the Singular Value Decomposition calculation for X^t:

$$X^t = U\Sigma V^T, \tag{3}$$

where U is a $N \times N$ unitary matrix, V^T is a $M \times M$ unitary matrix. Σ is a positive diagonal matrix with $N \times M$ dimensions, whose elements Σ_i on the diagonal is the singular value of X^t. When Σ_i are sorted in a descending order, Σ_1 represents the maximum. The threshold is denoted by η, then r^t represents the number of Σ_i that satisfy $\frac{\Sigma_i}{\Sigma_1} \geq \eta$. From the previous analysis we can see that if $r^t > r^{t\text{-}1}$ is satisfied, the picture changes greatly from time t to time t-1. On the other hand, if $r^t < r^{t\text{-}1}$, the picture will not change substantially from time t to time t-1. So we extract the frame with the largest r^t as the candidate frame.

3.1.2 CNN-based feature extraction

Convolutional Neural Networks (CNNs) [2, 6, 13, 16, 32] have showed excellent performance when applied to object detection benchmark datasets and standard image classification, including handwriting recognition, object recognition, human action tracking and many more. Such networks have a considerably more sophisticated structure than standard representations, comprising several layers of non-linear feature extractors.

Content-rich movie frames require a higher level of feature representation. Compared with the hand-crafted features, CNN can build low-level to high-level maps to achieve the purpose of learning a hierarchy of features. To this end, CNN based features are proposed. In this paper, we use the VGG-F model pre-trained on the ImageNet dataset to extract the features of frames.

The VGG-F model can establish an intrinsic representation of data, because its deep structure is derived by extracting complex structure from a large amount of information. The features of VGG model which has been pre-trained on the ImageNet are effective to represent the movie frames. Firstly, ImageNet is a very rich dataset containing more than 10 million natural images, hence the extracted features can directly or indirectly contain similar features of those movie frames. Secondly, the contents of movies are basically real scenes, even special effects in science fiction movies are also made very realistic. Finally, to accommodate the model to the updated movie frames feature extraction issue, we carry out dataset specific fine-tuning, which make the improvement of performance. The input frames and the architecture of the feature extraction model is shown in Fig. 2, which includes seven learnable layers, five of which are convolutional, and the last two layers are fully-connected. The input image size is converted to 224×224 pixels [4]. The processed images are extracted by the filtering operation of the convolutional layers and the down sampling of the pooling layers. Each frame output a 4096-D features vector, thus the corresponding feature matrixes can be constructed from these frames.

3.1.3 Preparation of labeled and unlabeled dataset

As semi-supervised classification requires some labeled data, the preparation of the training dataset can be implemented as such that the candidate frames of the movie trailers are classified as positive. In this paper, the approach of image matching by speeded up robust features (SURF) [3] is adopted to calculate the similarity between the frames of movies and the trailers, which not only improved the speed of operation, but also invariant to image rotations.

First of all, due to the better accuracy of the Hessian matrix, the algorithm uses the Hessian matrix to extract the interesting points. The formula for Hessian matrix is presented as follows:

$$HM(X,\sigma) = \begin{bmatrix} G_{xx}(X,\sigma) & G_{xy}(X,\sigma) \\ G_{xy}(X,\sigma) & G_{yy}(X,\sigma) \end{bmatrix}, \tag{4}$$

where

$$G_{xx}(X,\sigma) = \frac{\partial^2 g(X,\sigma)}{\partial x^2} \otimes I(X). \tag{5}$$

In other words, $G_{xx}(X,\sigma)$ is the convolution of the Gaussian second order derivative in point $X=(x,y)$ at scale σ, $I(X)$ represents the pixel of point X. $G_{xy}(X,\sigma)$ and $G_{yy}(X,\sigma)$ are similar to

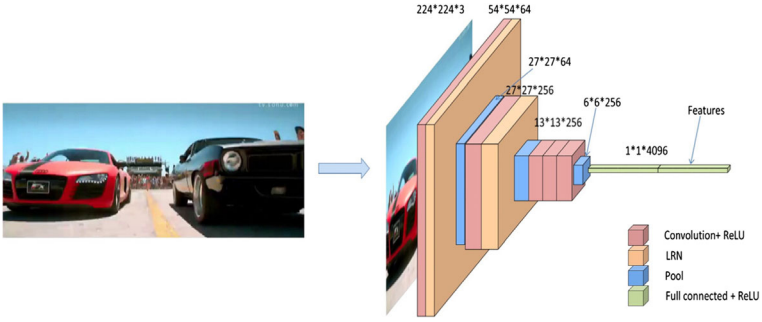


Fig. 2 CNN-based features extraction

the calculation of $G_{xx}(X, \sigma)$. It mainly uses the integral principle, which greatly reduces the computation.

The points of interests are assigned directions in order to achieve feature matching when the image is rotated. The horizontal and vertical Haar wavelet responses of all pixels are calculated and multiplied by the Gaussian weights of the corresponding locations, respectively, in a circular domain centered at the points of interests with a radius of 6 s (s is the scale value of the point). The direction of the largest sum of responses is the main direction of this feature point.

Figure 3 shows the SURF matching results for frames of a movie and its trailer. The small circles with different colors indicate the feature points (taking 10 points as an example), the matching points are joined by a straight line. As seen in the figure, the SURF algorithm has a strong robustness in the presence of different frame sharpness, different picture aspect ratios, and the slight change in position, including the angle and size of the objects.

Let the label of movie frame x_i be q_i, o_j indicates the frame of the official trailer, the label $Q = \{q_1, q_2, \cdots, q_i, \cdots, q_n\}$ is defined as follows:

$$q_i = \begin{cases} +1 & \mathrm{x}_i \text{ matches with } o_j \\ -1 & \mathrm{x}_i \text{ does not match with } o_j \end{cases}, \tag{6}$$

where the frames of the movie matched with the corresponding trailer are treated as the positive instances, and the rest of frames are treated as the negative instances. Frames of the movies without the trailers are served as unlabeled data samples.

As the official movie trailer is very short, usually only a few minutes or even shorter, there must be a large number of frames that are regard as negative instances, whilst only a few dozen samples can be chosen as positive samples. In order to prevent the final classification result from being affected by the extreme imbalance of the data, we uniformly sample the large number of negative instances, so the number of positive and negative instances will not be far apart.

3.2 Semi-supervised embedded learning

By literature search, we find that although supervised learning is usually more effective than unsupervised learning, it is arduous to obtain supervised datasets for large numbers of movie frames [27]. On the other hand, the classification results of supervised learning rely heavily on training set. Since the variety of movie content, the supervised classifier cannot solve our issue. Therefore, in order to improve the feasibility of classification, we choose to use the semi-supervised learning algorithm [10, 34]. There are multiple large-margin low-density separators

Fig. 3 The SURF matching results for frames of the movie “Les Misérables” and its trailer

in the semi-supervised classification, which coincide well with the labeled data. It may be risky to select any one of the separators without distinguishing them by further prior information [12]. To achieve the best possible balance, we use the safe semi-supervised learning (S4VM) to classify the data, which has the advantage of considering all the candidate separators [11, 17]. For the sake of better classification accuracy, we calculate the new training dataset and test dataset generated by embedded mappings to drive the S4VM.

In order to adjust the distance of the intra-class and inter-class data in the classification space to further improve the classification efficiency, we introduce the embedded algorithm into the calculation process. The disadvantage of the existing embedded algorithms is that the level of feature importance often neglects the relationship between samples, thus limiting the classification effect. To settle this problem, mutual information is introduced to weigh the importance of features during the process of embedded learning [36]. Unlike other studies, the classification of unlabeled data is guided by the semi-supervised learning algorithm, which requires the labeled data to be priori. Consequently, the effect of the labeled data can be significantly enhanced after the importance of the features is adjusted according to the characteristics of the trailers, and the experiments support that our arrangement improves the classification accuracy by calculating the effect of feature selection and feature importance. Finally, the selected features are used to form a new dataset to be applied to the classification algorithm.

Data embedding framework for classification (DEFC) proposed by Martinez et al. and Mu et al. [20, 21] is formulated under the principle of "Friends being close and enemies being apart", in which friend closeness (C_F) and enemy dispersion (D_E) are defined as follows, respectively:

$$C_F = \sum_{i,j=1}^{n} wdist_F\left(X_i, X_j | W_{ij}\right), \tag{7}$$

$$D_E = \sum_{i,j=1}^{n} wdist_E\left(X_i, X_j | W_{ij}\right). \tag{8}$$

To influence the embedding process, we introduce a feature selection factor ($T_{Feature}$), and hence the new formula can be defined as follows:

$$C_F = \sum_{i,j=1}^{n} wdist_F\left(X_i, X_j | W_{ij}, T_{Featrure}\right), \tag{9}$$

$$D_E = \sum_{i,j=1}^{n} wdist_E\left(X_i, X_j | W_{ij}, T_{Featrure}\right), \tag{10}$$

Where C_F and D_E represent the distances between intra-class and inter-class objects of the dataset X under the influence of weights and feature importance. The weights of the samples in C_F are mostly larger, which makes them closer in the embedded space. In contrast, the weights of the samples in D_E make them more distant in the embedded space. The frames belong to different classes are forced to distance themselves resulting in an improvement in the final classification.

3.2.1 Calculation of feature weights

The mutual information MI_i between feature vectors E_i from the dataset E, which is extracted by CNN model, and the label vector Q is calculated as:

$$MI(\mathrm{E_i}, Q) = H(\mathrm{E_i}) + H(Q) - H(\mathrm{E_i}, Q), \tag{11}$$

where $H(E_i)$ and $H(Y)$ represent the marginal entropy of E_i and Y, respectively. $H(E_i, Y)$ calculates the joint entropy of E_i and Y.

δ represents the set of feature weights of E, δ is calculated as:

$$\delta_i = \frac{MI_i}{\sum_{i=1}^{d} MI_i}, \tag{12}$$

where d is the dimension of the optimum feature.

Finally, we propose to reconstruct the new feature set X for training dataset, and X' for test dataset, as follows via consideration of the mutual information entropy, where those feature vectors whose mutual information entropy is zero are excluded. Specially, the selection of X' deeply depends on X. Furthermore, θ represents the set of feature weights.

$$X = \{E_i | MI_i > 0\}, \tag{13}$$

$$X' = \{E'_i | MI_i > 0\}, \tag{14}$$

$$\theta = \{\delta_i | MI_i > 0\}. \tag{15}$$

3.2.2 Computation of the embedding process

We incorporate the principle of "friends being close and enemies being apart" into the design of our embedded algorithms. It improves classification by reducing the distance between samples of the same class and increasing the distance between samples of different classes. Detailed description of the embedding computation are below.

1) *Compute the indicator matrix*

The labeled data samples are represented as pairwise constraints (x_i, x_j, l_{ij}), when the labels q_i and q_j of the frames x_i and x_j are same, otherwise, $l_{ij} = -1$. The indicator matrix L in this paper is a two-valued indicator, which is calculated as follows:

$$l_{ij} = \begin{cases} 1 & q_i \oplus q_j = 1 \\ -1 & q_i \oplus q_j = 0 \end{cases}. \tag{16}$$

2) *Compute the similarity between samples*

A wide variety of algorithms are used to calculate distances, including Cosine similarity, Euclidean norm, Polynomial kernel, et al. Experimentally, we find that the Cosine similarity classification works best in our case. We use the Cosine similarity algorithm to acquire the similarity between samples accordingly. The similarity matrix W is calculated as below:

$$w_{ij} = \left| \frac{x_i^T x_j}{\|x_i\| \|x_j\|} \right|, \tag{17}$$

where w_{ij} represents the similarity between two feature vectors x_i and x_j of the frames in dataset X, which is formulated in Eq. (13). $\|x_i\|$ and $\|x_j\|$ are the norm of x_i and x_j, respectively.

3) *Calculate the relation feature set*

As samples can be indicated by their relative positions expressed in terms of their similarity to other objects, relational values provide a new method to represent the information of the dataset rather than entering the original features. While the existing research has proved the validity of relational values instead of the original samples applied as the input to classifiers, the similarity feature is not calculated in connection with the influence of the features, which restricts the classification result.

We propose a $n \times n$ feature matrix $S_{ij}=\phi(x_i, x_j)$, where x_i and x_j are the *ith* and *jth* objects of the training set X in Eq. (13), an $m \times n$ feature matrix $S'_{ij}=\phi(x'_i, x_j)$ where x'_i is the *ith* element of test set X' in Eq. (14) and x_j is the *jth* element of the training set X. We can obtain the nonlinear structures in the dataset by capturing the interaction between the objects via the relation features calculated. As the calculation complexity of the embedding algorithms depends on d, the relational features simplify calculations when the initial data dimensionality is much greater than the amount of data ($d>>n$).

In order to solve the problem that each feature of a movie frame tends to exhibit different intensities when expressing different styles, we use the Euclidean distance to describe the relationship between the features based on weighting theory [5]. The weighted Euclidean distance process is described below.

$$d(X,Y)=\sqrt{\sum_{i=1}^{l}\xi_i^{\alpha}(x_i-y_i)^2}, \tag{18}$$

where $X=(x_1,\cdots,x_l)^T$, $Y=(y_1,\cdots,y_l)^T$, α is the weight coefficient, ξ_1, $\cdots$, $\xi_l \in R^+$, and $\sum_{i=1}^{l}\xi_i=1$.

In summary, S_{ij} and S'_{ij} can be calculated through the feature weighting θ, and the corresponding formula is formulated as follows:

$$S_{ij}=\sqrt{\sum_{k=1}^{m}\theta_k^{\alpha}\left(x_{ik}-x_{jk}\right)^2}; \tag{19}$$

$$S'_{ij}=\sqrt{\sum_{k=1}^{m}\theta_k^{\alpha}\left(x'_{ik}-x_{jk}\right)^2}, \tag{20}$$

where m is the feature dimension of dataset X, θ is calculated by Eq. (15), θ_k is the feature weight of the *kth* feature, x_{ik}, x'_{ik} and x_{jk} are the eigenvalues of the *kth* feature of elements x_i, x'_i and x_j, respectively.

4) *Calculate the optimal projection matrix*

In this session, the training set uses the relational features set S which is provided by Eq. (19). Let P be the projection matrix, then $Z=SP$ implements the mapping transformation from S to Z, where $PP'=I$ and I is the unit matrix.

Let C_F and D_E represent the "friend closeness" and "enemy dispersion", respectively, they can be formulated as:

$$C_F = \sum_{i,j=1}^{m} \frac{l_{ij}\left(l_{ij}+1\right)}{2} w_{ij} \left\| z_i - z_j \right\|^2, \tag{21}$$

$$D_E = \sum_{i,j=1}^{m} \frac{l_{ij}\left(l_{ij}-1\right)}{2} w_{ij} \left\| z_i - z_j \right\|^2, \tag{22}$$

where the indicator L and weight function W are obtained from Eq. (16) to Eq. (17), respectively, m is the number of samples in set Z.

The C_F and D_E metrics can be optimized in a number of ways, such as minimizing C_F individually or maximizing D_E individually. We implement these options using various trace optimization templates to improve the compactness and separability of those classes, and hence to generate the optimum embedding for dimensionality reductions.

Let

$$A' = \left[a_{ij}{}' \right]; \tag{23}$$

$$B' = \left[b_{ij}{}' \right], \tag{24}$$

where

$$a_{ij}{}' = l_{ij}\left(l_{ij}-1\right) w_{ij}; \tag{25}$$

$$b_{ij}{}' = l_{ij}\left(l_{ij}+1\right) w_{ij}. \tag{26}$$

By combining (23) and (25) with (24) and (26), A' and B' can be reconstructed as:

$$A' = L.*(L - I_{m\times m}).*W, \tag{27}$$

$$B' = L.*(L + I_{m\times m}).*W, \tag{28}$$

where $I_{m\times m}$ is the all-ones matrix, in which m is the feature dimension of dataset Z.

In combination with Eq. (21) and Eq. (28), the minimization of C_F is calculated as:

$$\begin{aligned} \min_{pp^T=1} C_F &= \min_{pp^T=1} \sum_{i,j=1}^{d} \frac{b_{ij}{}'}{2} \left\| z_i - z_j \right\|^2 \\ &= \min_{pp^T=1} trace\left(\sum_{i,j=1}^{d} z_i b_{ij}{}' z_i{}^T - \sum_{i,j=1}^{d} z_i b_{ij}{}' z_j{}^T \right). \\ &= \min_{pp^T=1} trace\left(ZD\left(B'\right)Z^T - ZB'Z^T \right) \\ &= \min_{pp^T=1} trace\left(Z\left(D\left(B'\right) - B'\right)Z^T \right) \end{aligned} \tag{29}$$

In combination with Eq. (22) and Eq. (27), the maximization of C_E can be calculated as:

$$\begin{aligned}\max_{pp^T=1} D_E &= \max_{pp^T=1} \sum_{i,j=1}^{d} \frac{a_{ij}^{'}}{2} \left\| z_i - z_j \right\|^2 \\ &= \max_{pp^T=1} trace\left(\sum_{i,j=1}^{d} z_i a_{ij}^{'} z_i^T - \sum_{i,j=1}^{d} z_i a_{ij}^{'} z_j^T \right), \\ &= \max_{pp^T=1} trace\left(ZD\left(A^{'}\right) Z^T - ZA^{'} Z^T \right) \\ &= \max_{pp^T=1} trace\left(Z\left(D\left(A^{'}\right) - A^{'} \right) Z^T \right)\end{aligned} \tag{30}$$

where $D(B^{'})$ and $D(A^{'})$ in the Eq. (29) and Eq. (30) are the diagonal matrices of $B^{'}$ and $A^{'}$, respectively. The formula punishes the dissimilarities between intraclass points or rewards their similarities by minimizing $trace(Z(D(B^{'}) - B^{'})Z^T)$, whilst it rewards the dissimilarities between interclass points or punishes their similarities by maximizing $trace(Z(D(A^{'}) - A^{'})Z^T)$. Next, the Laplacian matrices of $A^{'}$ and $B^{'}$ can be calculated as follow:

$$A = D\left(A^{'}\right) - A^{'}; \tag{31}$$

$$B = D\left(B^{'}\right) - B^{'}. \tag{32}$$

Since the matrixes A, B, ZAZ^T, ZBZ^T, indicator matrix L and similarity matrix W in the above formula are symmetric, we can get the following conversion relationship:

$$trace\left(ZAZ^T\right) = trace\left(\left(ZAZ^T\right)^T\right), \tag{33}$$

$$trace\left(ZBZ^T\right) = trace\left(\left(ZBZ^T\right)^T\right). \tag{34}$$

In summary, we obtain the Eq. (35) and Eq. (36) as follow:

$$\begin{aligned}\min_{pp^T=1} C_F &= \min_{pp^T=1} trace\left(ZBZ^T\right) \\ &= \min_{pp^T=1} trace\left(Z^T BZ\right) \\ &= \min_{pp^T=1} trace\left(P^T S^T BSP\right),\end{aligned} \tag{35}$$

$$\begin{aligned}\max_{pp^T=1} D_E &= \max_{pp^T=1} trace\left(ZAZ^T\right) \\ &= \max_{pp^T=1} trace\left(Z^T AZ\right) \\ &= \max_{pp^T=1} trace\left(P^T S^T ASP\right).\end{aligned} \tag{36}$$

The projection matrices procured by resolving Eq. (35) and Eq. (36) are found inconsistency. However, the classification results based on transformation dataset are consistent. Therefore, projection matrix P can be calculated by resolving either Eq. (35) or Eq. (36).

5) *Calculate the new dataset*

We calculate the new training dataset Z and the new test dataset $Z^{'}$ by using optimal projection matrix P, relation feature set S and $S^{'}$ calculated by Eq. (19) and Eq. (20), respectively. The mapping process is as shown in the following formula:

$$Z = SP; \tag{37}$$

$$Z^{'} = S^{'}P. \tag{38}$$

3.2.3 Safe semi - supervised learning (S4VM)

As calculated above, a labeled dataset Z, Q and an unlabeled dataset Z' are prepared. And then, we utilize S4VM [17] to complete the final classification of the frames. Let n and m are the number of the labeled and unlabeled samples, respectively. We need a function to calculate the labels q' for the test dataset. The target function of the single large-margin low-density is defined as:

$$\varphi\left(f, q^{'}\right) = \min_{f\in\Gamma, q'\in\chi} \frac{\|f\|_{\Gamma}}{2} + \gamma_1 \sum_{i=1}^{n} \ell(q_i, f(z_i)) + \gamma_2 \sum_{j=1}^{m} \ell\left(q'_j, f\left(z'_j\right)\right), \tag{39}$$

where $\ell(q, f(z)) = \max\{0, qf(z) - 1\}$ represents the hinge loss, Γ represents the Reducing Kernel Hilbert Space (RKHS), and $\chi = \left\{q^{'} \in \{\pm 1\}^m \middle| -\beta \le \frac{\sum_{j=1}^{m} q'_j}{m} - \frac{\sum_{i=1}^{n} q_i}{n} \le \beta\right\}$ represents a set of labels procured from field knowledge. γ_1 and γ_2 are regularized parameters, we set $\gamma_1 = 100$, $\gamma_2 = 0.01$ in this case.

Apply the following formula to calculate multiple separators $\{f_t\}_{t=1}^{T}$ and the corresponding label assignments:

$$\min_{\{f_t, q'_t \in \chi\}_{t=1}^{T}} \sum_{t=1}^{T} \varphi(f_t, q'_t) + M\Omega\left(\{q'_t\}_{t=1}^{T}\right), \tag{40}$$

where T and Ω are the amount of separators and punishment about the multiplicity of separators, respectively. M is a greater constant which enforces large diversity.

To make the improved performances, we learn a classifier over an inductive SVM [17]. Let $earn(q, q^{'}, q^{svm})$ represents the increased accuracies. $lose(q, q^{'}, q^{svm})$ represents the decreased accuracies by contrast. The goal of S4VM is to learn a classifier q to maximize the performance gains on SVM. In the calculations below, we convert it to an optimization problem to maximize its roles:

$$q = \arg\max_{q\in\{\pm 1\}^m} \min_{q'\in\{q'_t\}_{t=1}^{T}} earn\left(q, q^{'}, q^{svm}\right) - \lambda lose\left(q, q^{'}, q^{svm}\right), \tag{41}$$

where parameter λ plays a role of trading-off how much risk to undertake, and assume q' can realize the ground truth boundary, which is the worst-case separator in $\{q'_t\}_{t=1}^{T}$.

3.3 Output Trailers

Our empirical study finds that a shot in the trailer usually does not exceed 3 s. Further, the National Association of Theatre Owners (NATO) recommends that film trailers run no more

Table 1 Movie information

No.	Movie	Time (hh: mm: ss)	Frames per second (FPS)
1	Les Misérables	02:38:06	18
2	The Dark Knight Rises	02:26:14	18
3	Transformers 3	02:34:19	24
4	Inception	02:28:06	24
5	Iron Man 3	02:10:32	24
6	Thor 2	01:52:04	25
7	Edge of Tomorrow	01:53:29	25
8	Harry Potter and the Goblet of Fire	02:37:05	24
9	Prometheus	02:03:46	24
10	Resident Evil: Retribution	01:35:37	24

than 2 min [30]. While the guideline is not mandatory, it does provide a reference for the general duration of a movie trailer. Therefore, the movie trailer is composed of clips, which are extracted from neighboring key-frames, and each clip lasts about 3 s.

4 Experimental evaluations

Our method is evaluated using a dataset with 10 movies, the specific information is listed in Table 1. The top 5 movies in the table and their corresponding trailers are utilized as the labeled dataset, and the rest is made as the unlabeled dataset. Details of the official trailers as training data are contained in Table 2. We manually removed the special effect clips in the trailers before fetching the frames because these shots did not appear in the original movies.

Some sample frames from the official trailer and the trailer we produced for the movie "Thor 2" are shown in Fig. 4. As illustrated in Fig. 4, frames (1) ~ (5) are taken from the official trailer, while frames (6) ~ (10) are from the automatically predicted trailer. As seen, our automatically generated trailer is quite similar to the official trailer. Moreover, the predicted trailer includes the above mentioned protagonists, prominent scenes such as explosions and actions, dialogue scenes, the iconic items and so on. It indicates that our approach has the ability to produce high-quality trailers.

Figure 5 compares the automatically generated trailer with the official trailer in the timeline of "Harry Potter and the Goblet of Fire". The top of the timeline shows some frame samples inside our trailer and the bottom is the official trailer. As shown in Fig. 5, our trailer coincides with the official trailer in many scenes, which manifests that our method is effective in generating movie trailers. Some scenes are unique in our generated trailer (as shown in the green sections), which are especially attractive by sentiment, tableau sense, color and scenario.

Table 2 Official trailer information of the training movies

No.	Movie	Time (hh: mm: ss)	FPS
1	Les Misérables	00:01:02	25
2	The Dark Knight Rises	00:00:26	25
3	Transformers 3	00:00:24	25
4	Inception	00:01:48	25
5	Iron Man 3	00:00:24	25

Fig. 4 Some sample frames in official trailer and our generated trailer of the movie "Thor 2"

This is a powerful illustration of our approach being accurate and flexible enough to fully promote the movie. It can also be noticed that a few scenes from the official trailer, as shown in the red sections, are not acquired by our algorithm, suggesting that these samples not very remarkable in our algorithm, and our algorithm considers these as negative instances.

As there exists no universal standard for estimating a movie trailer to be satisfied or not, people always evaluate trailers by subjective consciousness. To assess our achievement properly, we put forward a suitable method to calculate the similarities between the official trailer and the trailer produced by our algorithm. By considering the official trailers as the ground-truth, we calculate the degree of similarity by matching each frame in the official trailer with all frames of our predicted trailers to assess the effectiveness of our proposed trailer production algorithm. If the similarity degree between these two frames is greater than 0.9, we regard them as a successful match. Giving the number of matching frames as k, the accuracy μ can be calculated as:

$$\mu = \frac{\mathrm{k}}{\mathrm{n}}, \tag{42}$$

where n stands for the total number of frames inside the generated trailers.

We compare the results from four methods, including hand-crafted features combined with supervised SVM [27], hand-crafted features combined with S4VM [11], Deep Learning algorithm [2], and our proposed approach. The data in Table 3 shows that our approach outperforms the other three methods on all test movies by average values. As seen from

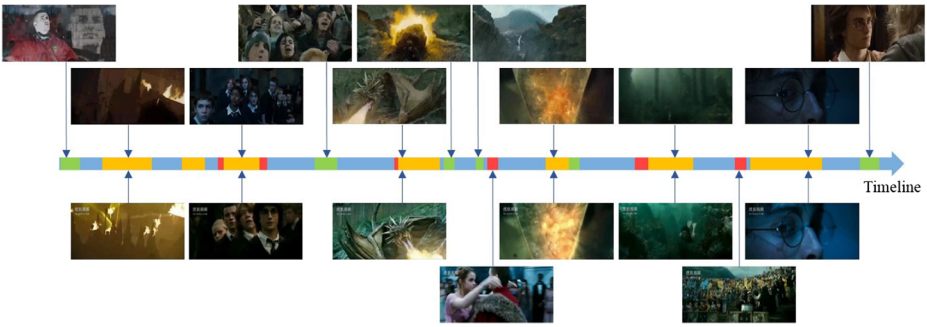


Fig. 5 Compare the scenes included in our automatically generated trailer with those in the official trailer of the movie called "Harry Potter and the Goblet of Fire". This is the timeline of the movie. Scenes in the orange sections of the timeline appear in both our trailer and the official trailer, which indicate that our trailer overlap the official trailer in the original film. The green sections indicate the scenes are unique to our trailer, while the red sections represent the particular components in the official trailer

Table 3 Accuracy of different methods on official movie trailers (%)

Movie	Hand-crafted features and SVM	Hand-crafted features and S4VM	Deep learning	Ours
Thor 2	22.73	52.68	73.78	80.16
Edge of tomorrow	25.92	36.13	84.29	91.86
Harry Potter and the Goblet of Fire	21.68	40.38	68.06	75.95
Prometheus	11.50	24.25	66.39	67.50
Resident evil: retribution	55.43	31.12	86.45	88.05
Average	27.45	36.91	75.79	80.70

Table 3, movies "Edge of Tomorrow" and "Resident Evil: Retribution" have high accuracy. These two films contain a lot of attractive and exciting contents such as actions and explosions, as well as the scenes with perfect composition such as protagonists' close-up and dialogues. These shots are often the important parts of the trailer and can be recognized by our algorithm, leading to high accuracy. It can also be noticed that "Prometheus" has the lowest accuracy, due to the fact that the trailer of "Prometheus" creates a mysterious atmosphere mainly with soundtracks, horror scenes and close-up of aliens. Although the content is very different from those official trailer, our trailer still contains high-impact factors, such as characters, dialogues, objects and environment. Further research can be identified that the importance of the role and audio information can be introduced to improve the performances of our approach.

5 Conclusion

In this study, we propose an embedded learning algorithm to drive the semi-supervised classification and achieve automated production of movie trailers. While reducing redundancy of the movies by extracting candidate frames, we exploit the powerful CNN model to extract the features of candidate frames and prepare the labels for the training dataset, in which the frames of the movie matched with the corresponding trailer are regarded as the positive instances, and the rest of frames are regarded as the negative instances. Following that, we calculate the new dataset by the introduced embedded learning algorithm, in which the feature selection and the influence of feature importance play important roles. Finally, S4VM is adopted to make full use of the guiding role of labeled data to implement a semi-supervised classification. As a result, the trailer is generated by stitching the clips around the key frames of the classifications, and empirical evaluation show that our proposed performs more effective than the other representative benchmarks. This illustrates that our proposed method can be applied to alleviating the work of filmmakers and providing viewers with high-quality viewing guides via such automatically generated movie trailers. There also exist enormous potential and space for further improvements, that more factors can be included to enhance the presentation of the movie content, such as the importance of the casts and the sounds.

Acknowledgements The authors wish to acknowledge the financial support for the research work under National Natural Science Foundation in China (Grant No.61502331, No.61602338, No.11701410), Natural Science Foundation of Tianjin (Grant No.15JCQNJC00800).

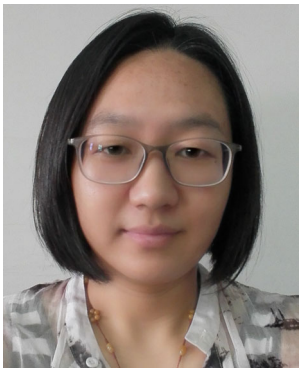
References

1. Abd-Almageed W (2008) Online, simultaneous shot boundary detection and key frame extraction for sports videos using rank tracing. In: Proceedings of the IEEE International Conference on Image Processing pp 3200–3203
2. Almuashi M, Hashim SZM, Mohamad D et al (2017) Automated kinship verification and identification through human facial images: a survey[J]. Multimed Tools Appl 76(1):265–307
3. Bay H, Tuytelaars T, Gool LV (2006) Speeded-up robust features (SURF). In: Proceedings of the European Conference on Computer Vision pp 404–417
4. Chatfield K, Simonyan K, Vedaldi A et al (2014) Return of the devil in the details: delving deep into convolutional nets. In: Proceedings of the British Machine Vision Conference, BMVA Press
5. Cheng D, Nie F, Sun J et al (2017) A weight-adaptive laplacian embedding for graph-based clustering[J]. Neural Comput 29(7):1902–1918
6. Cheng G, Yang C, Yao X et al (2018) When deep learning meets metric learning: remote sensing image scene classification via learning discriminative CNNs. IEEE Transactions on Geoscience & Remote Sensing pp 1–11
7. Ding C, Zhang L (2015) Double adjacency graphs-based discriminant neighborhood embedding[J]. Pattern Recogn 48(5):1734–1742
8. Ejaz N, Tariq TB, Baik SW (2012) Adaptive key frame extraction for video summarization using an aggregation mechanism[J]. J Vis Commun Image Represent 23(7):1031–1040
9. Han Y, Yang Y, Wu F et al (2015) Compact and discriminative descriptor inference using multi-cues. IEEE Trans Image Process 24(12):5114–5126
10. Han Y, Yang Y, Yan Y et al (2015) Semisupervised feature selection via spline regression for video semantic recognition. IEEE Trans Neural Netw Learn Syst 26(2):252–264
11. Huang F, Wen C, Luo H et al (2016) Local quality assessment of point clouds for indoor mobile mapping[J]. Neurocomputing 196(C):59–69
12. Joachims T (1999) Transductive inference for text classification using support vector machines. Sixteenth Int Conf Mach Learn 117(827):200–209
13. Kim J, Lee JK, Lee KM (2016) Accurate image super-resolution using very deep convolutional networks. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp 1646–1654
14. Li J, Yao T, Ling Q et al (2017) Detecting shot boundary with sparse coding for video summarization[J]. Neurocomputing 266(C):66–78
15. Liu J, Pengren A, Ge Q et al (2017) Gabor tensor based face recognition using the boosted nonparametric maximum margin criterion[J]. Multimed Tools Appl 1–15
16. Liu P, Guo JM, Wu CY et al (2017) Fusion of deep learning and compressed domain features for content based image retrieval. IEEE Trans Image Process 26(12):5706–5717
17. Li YF, Zhou ZH (2015) Towards making unlabeled data never hurt. IEEE Trans Pattern Anal Mach Intell 37(1):175–188
18. Lu S (2004) Content analysis and summarization for video documents. PhD thesis, Research Associate, VIEW lab, the Chinese University of Hong Kong, Department of Computer Science & Engineering
19. Maronidis A, Tefas A, Pitas I (2015) Subclass graph embedding and a marginal fisher analysis paradigm[J]. Pattern Recogn 48(12):4024–4035
20. Martinez E, Mu T, Jiang J et al (2013) Automated induction of heterogeneous proximity measures for supervised spectral embedding. IEEE Trans Neural Netw Learn Syst 24(10):1575–1587
21. Mu T, Jiang J, Wang Y et al (2012) Adaptive data embedding framework for multiclass classification. IEEE Trans Neural Netw Learn Syst 23(8):1291–1303
22. Otani M, Nakashima Y, Sato T et al (2017) Video summarization using textual descriptions for authoring video blogs[J]. Multimed Tools Appl 76(9):12097–12115
23. Pfeiffer S, Lienhart R, Fischer S et al (1996) Abstracting digital movies automatically[J]. Vis Commun Image Represent 7(4):345–353
24. Roweis ST, Saul LK (2000) Nonlinear dimensionality reduction by locally linear embedding[J]. Science 290(5500):2323–2326
25. Sheng J, Jiang J (2014) Recognition of chinese artists via windowed and entropy balanced fusion in Classification of their authored ink and wash paintings (IWPs). Pattern Recogn 47(2):612–622
26. Sheng J, Jiang J (2013) Style-based classification of ink and wash Chinese paintings[J]. Opt Eng 52(9): 093101-1-093101-8
27. Smeaton AF, Lehane B, O'Connor NE et al (2006) Automatically selecting shots for action movie trailers. ACM Sigmm International Workshop on Multimedia Information Retrieval, Mir 2006, October 26-27, Santa Barbara, California, USA. DBLP 231–238
28. Sun S, Wang F, He L (2017) Movie summarization using bullet screen comments[J]. Multimed Tools Appl 1–18
29. Tenenbaum JB, De SV, Langford JC (2000) A global geometric framework for nonlinear dimensionality reduction[J]. Science 290(5500):2319–2323

30. Theaters advocate shorter trailers, marketing (2014) MarketingMovies.net. http://www.marketingmovies.net/news/theaters-advocate-shorter-trailers-marketing (accessed 2014.01.28)
31. Yao T, Mei T, Rui Y (2016) Highlight detection with pairwise deep ranking for first-person video summarization. In: IEEE International Conference on Computer Vision and Pattern Recognition pp 982–990
32. Yao X, Han J, Zhang D et al (2017) Revisiting co-saliency detection: A novel approach based on two-stage multi-view spectral rotation co-clustering. IEEE Trans Image Process 26(7):3196–3209
33. Zhang D, Meng D, Han J (2017) Co-saliency detection via a self-paced multiple-instance learning framework. IEEE Trans Pattern Anal Mach Intell 39(5):865–878
34. Zhang J, Han Y, Jiang J (2017) Semi-supervised tensor learning for image classification[J]. Multimedia Systems 23(1):63–73
35. Zhang K, Chao WL, Sha F et al (2016) Summary transfer: exemplar-based subset selection for video summarization. IEEE Conference on Computer Vision and Pattern Recognition, pp 1059–1067
36. Zhu J, Pu Y, Xu D et al (2016) The effect of image quality for visual art analysis[J]. J Comput Aided Des Comput Graph 28(8):1269–1278

Jiachuan Sheng received her PhD degree in computer science at Tianjin University in 2013. Currently, she is an associate professor at Tianjin University of Finance & Economics. She is also a master tutor. Her research interests include digital media processing via image processing and video processing approaches and pattern recognition.

Yaqi Chen received her Bachelor of Engineering degree in computer science in 2017. Now she is a post-graduate student of the Tianjin University of Finance & Economics. Her research interests include multimedia processing and machine learning.



Yuzhi Li is a lecturer at Tianjin University of Finance & Economics. Her research interests include machine learning and image processing. She is also a PhD. student at Tianjin University of Finance & Economics.

Liang Li received his PhD degree in the School of Computer Science and Technology at Tianjin University in 2014. Currently, he is a lecture at Tianjin University. His main research interests include video segmentation, object detection and multimedia related applications.