2014 22nd International Conference on Pattern Recognition

Automatic Video Genre Classification Using Multiple SVM Votes

Won-Dong Jang†, Chulwoo Lee†, Jae-Young Sim‡, and Chang-Su Kim†
†School of Electrical Engineering, Korea University, Seoul, Korea
‡School of Electrical and Computer Engineering,
Ulsan National Institute of Science and Technology, Ulsan, Korea
Emails: {wdjang, chulwoo}@mcl.korea.ac.kr, jysim@unist.ac.kr, cskim@mcl.korea.ac.kr

***Abstract*—A video genre classification algorithm based on the voting from multiple SVMs is proposed in this work. While conventional genre classifiers use generic baseline features, we employ more specialized features to describe five video genres: animation, commercial, entertainment, drama, and sports. We also present a robust classification algorithm using multiple SVMs, which consider all possible binary grouping of the five genres. Given a query video, each SVM casts a probabilistic vote for each genre. Then, the optimal genre with the maximum votes is selected. Experimental results show that the proposed algorithm provides more accurate classification performance than conventional algorithms.**

I. INTRODUCTION

With the popularization of video capturing devices, it has become easier to produce and process video materials, even using mobile phones. However, finding video sequences of interest is a difficult task, since there are a huge amount of videos on televisions and the internet. To facilitate the search and narrow down choices, we can use video attributes, including durations, creators, creation times, and genres. Especially, the genre of a video is an important cue for video categorization, which can be used in many applications, such as genre-specific video retrieval [1] and semantic video indexing [2]. However, the genre is usually unknown, until it is tagged manually. Attempts have been made to develop automatic video genre classifiers in the last decade [3]–[6].

In general, a genre classification algorithm consists of a feature extractor and a classifier. Yuan *et al*. [3] proposed spatiotemporal features and classified videos into multiple genres by employing several support vector machines (SVMs) in a tree structure. This hierarchical SVM classifier often yields a wrong decision, since all binary classifications along the tree should be correct for the right decision. Wang *et al*. [4] developed a genre classifier based on a hidden Markov model, but their classifier works for sports genres only and requires acoustic information in addition to visual information. Kowdle *et al*. [5] extracted SIFT features and adopted the bag-of-words approach for the video classification. Their object-based classifier is sensitive to the training set. Xu *et al*. [6] represented videos using self-similarity matrices, which are distinctive across video genres. Their scheme was presented theoretically, but not proved experimentally.

In this work, we propose an efficient video genre classifier to categorize videos into five genres: animation, commercial, entertainment, drama, and sports. These five genres are commonly broadcast over television networks. In addition to baseline features exploited in conventional algorithms, we propose novel genre-specific features to describe the target genres effectively. We also propose a voting-based classifier, which consists of multiple SVMs. Each binary SVM decides a different combination of positive and negative genres. A soft decision is possible by counting the votes from the multiple SVMs. We design a weighted voting scheme to improve the classification accuracy. Moreover, we develop an optional divide-and-conquer voting scheme. Experimental results demonstrate that the proposed algorithm outperforms the conventional algorithms [3], [7].

Note that a video classifier can use audio features as well as visual features [8]. A genre classifier also can employ both audio and visual information to improve the classification performance. However, in this work, we focus on the genre classification using visual features only.

The rest of the paper is organized as follows. Sections II and III describe the proposed feature extractor and classifier, respectively. Section IV presents experimental results. Finally, Section V concludes this work.

II. FEATURE EXTRACTION

Feature selection is important for robust genre classification. However, it is hard to find representative characteristics of each genre, since there are various types of videos in a single genre. In other words, even though videos have the same genre, they may yield different characteristics. Therefore, to describe each genre effectively, a combination of diverse features should be considered instead of a single feature.

We extract features from an input video V, as shown in Fig. 1. The video V is decomposed into multiple shots. To reduce the complexity, we sample frames selectively with sampling rate r_s and detect the shot transition at each sample frame. Specifically, we measure the dissimilarity between consecutive sample frames, based on the frame difference and the RGB histogram distance. If the dissimilarity is higher than

This work was supported partly by the Global Frontier R&D Program on <Human-centered Interaction for Coexistence> funded by the National Research Foundation of Korea grant funded by the Korean Government (MSIP) (2011-0031648), and partly by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT And Future Planning (2009-0083495).

1051-4651/14 $31.00 © 2014 IEEE
DOI 10.1109/ICPR.2014.459

TABLE I: We use 23 features to describe a video sequence, which are composed of baseline and proposed features.

	Feature type	Feature index	Feature	Feature extraction rate	Feature computation
Baseline	Shot	1	Shot frequency	Each sample frame	Number of shots / Video duration
	HSV	2	Frequent hue	1st frame in each shot	Median of values
		3-4	Mean of saturation	1st frame in each shot	Mean and standard deviation of values
		5-6	Mean of brightness	1st frame in each shot	Mean and standard deviation of values
		7-8	Standard deviation of saturation	1st frame in each shot	Mean and standard deviation of values
		9-10	Standard deviation of brightness	1st frame in each shot	Mean and standard deviation of values
	Gradient	11-12	Gradient magnitude	1st frame in each shot	Mean and standard deviation of values
	Motion	13-14	Motion intensity	r_m	Mean and standard deviation of values
		15-16	Motion variance	r_m	Mean and standard deviation of values
	Text	17	Text frame frequency	r_m	Number of frames with texts / Video duration
		18	Average text area	r_m	Total text area / Number of text regions
Proposed	Shot	19	Repeated shot frequency	1st frame in each shot	Number of repeated shots / Number of shots
		20	Shot type variety	1st frame in each shot	Number of shot types / Video duration
		21	Flat shot detection	1st frame in each shot	Binary existence
	Gradient	22-23	Extreme gradient	1st frame in each shot	Mean and standard deviation of values

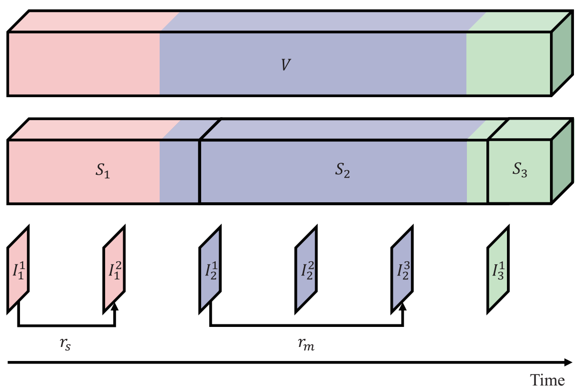


Fig. 1: We decompose an input video V into multiple shots S_i, and sample frames I_i^j from shot S_i with sampling rate r_s. For the motion and text features, we further decimate the sampled frames with sampling rate r_m, where $r_m > r_s$.

a threshold, we determine that there is a shot transition. In Fig. 1, S_i denotes an estimated shot, and I_i^j denotes a sample frame in S_i. Note that the estimated shots may be different from the true shots, which are depicted by different colors in Fig. 1, due to the sampling. Most features are extracted from the first frame in each shot. However, for some features, we carry out the feature extraction periodically, with sampling rate r_m, where $r_m > r_s$. In this work, $r_s = \frac{1}{5}$s and $r_m = 5r_s$.

Let us describe the baseline and proposed features, which are summarized in Table I, subsequently. Note that we extract a 23-dimensional feature vector $\mathbf{f} = [f_1, f_2, ..., f_{23}]$.

A. Baseline Features

We adopt 18 baseline features, which are often used in video classifiers [8]. The selected baseline features have five types: shot detection, HSV color, gradient, motion, and text.

We detect shots and use the shot frequency as the first feature f_1. Note that the shot detection is also important for efficient feature extraction. With the shot information, we can select a representative frame within each shot and extract features at that frame only. We use the first frame I_i^1 in shot S_i as the representative frame. As listed in Table I, most features are extracted only at the first frame in each shot.

(a) Frame # 719 (b) Frame # 735 (c) Frame # 791 (d) Frame # 975

Fig. 2: Alternatively repeated shots in a drama.

Next, we obtain HSV color features. The saturation and brightness channels have linear values, whereas the hue channel has cyclic values. Thus, in each representative frame, we measure the mean and standard deviation of saturation and brightness, while we determine the most frequent hue. For each feature, the number of feature values is equal to the number of shots in the video. We should summarize these feature values temporally for the overall video: we find the median of the most frequent hues (f_2), and calculate the mean and standard deviation of the saturation and brightness values ($f_3, ..., f_{10}$). Next, we extract luminance gradients in each representative frame using the Sobel operator. We measure the average magnitude of the gradients in the representative frame and compute the mean and standard deviation of the average magnitudes (f_{11}, f_{12}).

Motion and text features tend to vary within a shot, whereas the color and gradient features tend to be consistent. Thus, we sample frames with sampling rate r_m, and then estimate motion vectors and detect text regions at each sample frame. Then, we compute the mean and standard deviation of the motion intensities (f_{13}, f_{14}), and those of the directional variations (f_{15}, f_{16}). Also, we find edges in the luminance channel, analyze key strokes, and detect text regions. We compute the frequency of frames with texts (f_{17}) and the average area of the text regions (f_{18}).

B. Proposed Features

Although the baseline features are effective for various visual classifications tasks, they are too generic and are not

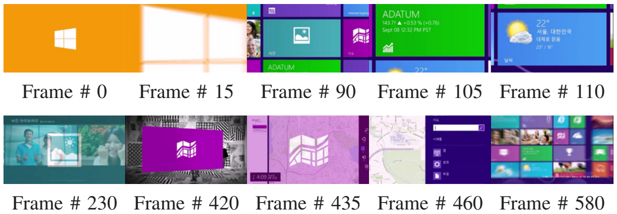
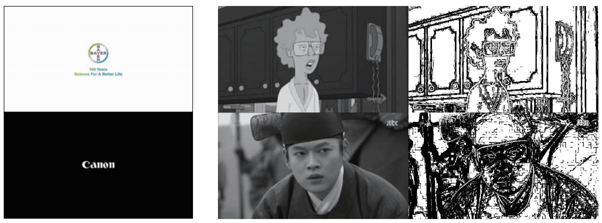


Fig. 3: Diverse shot types in a commercial.



(a) (b)

Fig. 4: Examples of (a) the flat shot detection and (b) the extreme gradient detection. The left and right subfigures in (b) are the luminance frames and the extreme gradient maps.

tailored for the genre classification. For more accurate classification, we propose five additional features, which are genre-specific and intuitive. The proposed features use shot detection results and spatial gradients.

First, we find a similar shot to the current shot S_i among the previous shots S_{i-2}, S_{i-3}. To this end, we construct the RGB histogram of the first frame I_i^1 in each shot S_i. Then, we compute the histogram differences between I_i^1 and I_{i-2}^1 and between I_i^1 and I_{i-3}^1, respectively. If either difference is smaller than a threshold, we declare S_i as a repeated shot. We use the frequency of repeated shots as feature f_{19}. As shown Fig. 2, two actors/actresses are often taken alternatively. The existence of repeated shots indicates that the input video is likely to be a drama.

Next, we employ the number of shot types as feature f_{20}. As exemplified in Fig. 3, a typical commercial is composed of diverse shots to give impacts within a short duration. Given each frame I_i^1, we compare its RGB histogram with those of all previous frames $I_1^1, \ldots, I_{i-2}^1$. If the histogram of I_i^1 is similar to that of a previous frame I_j^1, S_i is assigned the same shot type as S_j. Otherwise, S_i is assigned a new shot type. This procedure is carried out sequentially, and the number of shot types are counted at the end of the video.

Also, we detect a flat shot, which has flat background with small distinctive objects. Flat shots are often used in commercial videos to emphasize logos or products, as shown in Fig. 4(a). If the RGB histogram has a large maximum value, the corresponding shot is declared as flat. Then, the binary feature f_{21} is assigned bit 1 if the input video contains a flat shot, and bit 0 otherwise.

As the final features f_{22} and f_{23}, we count the pixels with extremely large or extremely small gradients in the first frame in each shot and compute the mean and standard deviation of those numbers. As illustrated in Fig. 4(b), animation videos include a lot of extreme gradients, as compared with videos

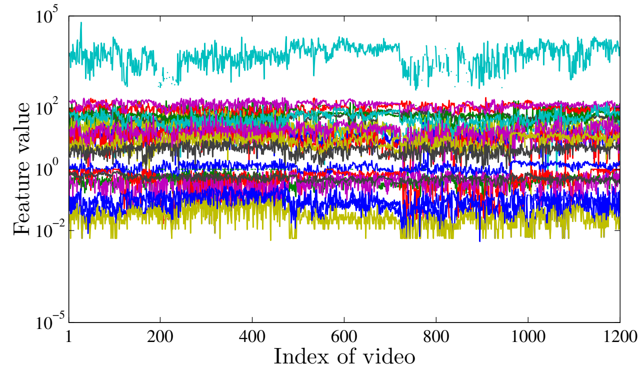


Fig. 5: Feature value variations of the training videos in the logarithmic scale: each curve depicts the variations of a single feature.

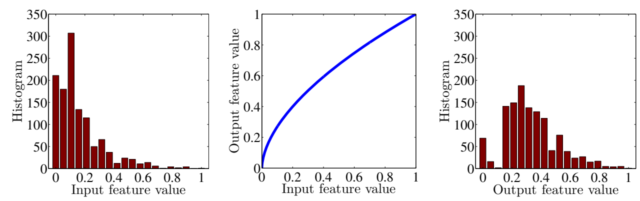


Fig. 6: A feature transform example: (a) input feature distribution, (b) power law transform, and (c) output feature distribution.

in the other genres. Therefore, features f_{22} and f_{23} help to distinguish the animation genre from the other genres.

C. Feature Transform

Each feature f_i has a different range in Fig. 5. For the classification, it is desirable to equalize the ranges of the multiple features. Thus, we first normalize each feature linearly to

$$\tilde{f}_i = \frac{f_i - f_i^{\min}}{f_i^{\max} - f_i^{\min}} \tag{1}$$

where $f_i^{\max}$ and $f_i^{\min}$ are the maximum and minimum of f_i.

The linear normalization, however, is not optimal if a feature has a biased distribution. For example, suppose that feature values are concentrated near 0, as in Fig. 6(a). Even after the normalization, most values are smaller than 0.4. To alleviate the bias, we use the power low to make the feature values more uniformly distributed. The power law tunes a value by

$$\hat{f}_i = \tilde{f}_i^{\gamma} \tag{2}$$

where $\hat{f}_i$ denotes a transformed feature. If the parameter γ is smaller than 1, the power law elevates feature values, as shown in Fig. 6(b). On the contrary, a larger γ reduces feature values. We determine γ using the mean of the linearly normalized feature values. A low mean indicates that the values are biased to 0. Therefore, if the mean is low, we set a small γ. On the contrary, if the mean is high, we set a large γ. Specifically, we have

$$\gamma = \begin{cases} (0.5 + \bar{f})^2 & \text{if } \bar{f} > 0.5, \\ 1/\left(1.5 - \bar{f}\right)^2 & \text{otherwise}, \end{cases} \tag{3}$$

where $\bar{f}_i$ denotes the mean of $\tilde{f}_i$. Fig. 6(c) shows the feature distribution after applying the power law.

III. Classification

We use multiple SVMs to classify genres. Note that SVM is widely employed in many computer vision problems [9]–[12]. For genre classification, Yuan *et al*. [3] used multiple SVMs in a tree structure hierarchically. For the right categorization, all binary classifications, leading to the leaf node corresponding to the true genre, should be correct. Once an SVM at an intermediate node outputs a false decision, there is no way to determine the true genre. To overcome this disadvantage, we propose a novel classifier using multiple SVMs that adopts a weighted voting scheme.

A. Voting Scheme

Given a feature vector $\mathbf{f}$, a single SVM classifier determines the class using a decision function

$$d(\mathbf{f}) = \sum_{l=1}^{L} \left(y_l \alpha_l k(\mathbf{f}, \mathbf{x}_l) + b \right) \tag{4}$$

where $\mathbf{x}_l$ is a support vector, and y_l and α_l are the sign and the Lagrange multiplier of $\mathbf{x}_l$, respectively. Also, k is a kernel, L is the number of support vectors, and b is a bias parameter. The binary SVM classifier categorizes the feature vector $\mathbf{f}$ into the positive or negative class, when the sign of $d(\mathbf{f})$ is positive or negative, respectively.

We perform the genre classification by counting votes from multiple SVMs. Let us recall that our objective is to classify videos into the five genres: (1) animation, (2) commercial, (3) entertainment, (4) drama, and (5) sports. There are

$$15 = \begin{pmatrix} 5 \\ 1 \end{pmatrix} + \begin{pmatrix} 5 \\ 2 \end{pmatrix}$$

combinations to divide these five genres into two classes. One class is called positive, and the other negative. We use 15 SVMs, one for each combination.

For example, an SVM distinguishes positive genres 1, 2 from negative genres 3, 4, 5. If the SVM classifies a query input as positive, genres 1 and 2 have the equal probability of $\frac{1}{2}$ to be selected as the estimated genre. Similarly, if it declares an input as negative, genres 3, 4, 5 have the probability of $\frac{1}{3}$.

In general, let $\psi_c(g)$ be a function, indicating whether genre g belongs to the positive class or the negative class in an SVM classifier c: $\psi_c(g) = 1$ if genre g is positive, and $\psi_c(g) = -1$ otherwise. Then, we assign a probability to genre g for the SVM classifier c as

$$p_c(g) = \frac{1}{\sum_{h=1}^{5} \delta\left(\psi_c(h) - \psi_c(g)\right)} \tag{5}$$

where δ is the Dirac delta function. The denominator is the number of the genres that have the same class as genre g.

TABLE II: Properties of training and test videos. FPS: frames per second.

Genre	Number of training videos	Number of test videos	Duration (sec)	FPS
Animation	240	144	120	23-47
Commercial	240	156	9-120	15-30
Entertainment	240	132	120	24-30
Drama	240	120	120	23-29
Sports	240	104	27-120	25-59
Total	1200	656	-	-

We perform the voting from all 15 SVM classifiers. For genre g, we count the votes by

$$v(\mathbf{f}, g) = \sum_{c=1}^{15} p_c(g) \cdot \max\{d_c(\mathbf{f})\psi_c(g), 0\}. \tag{6}$$

When the decision function returns a positive value, only the positive genres get votes. The negative genres get votes in a similar way. Note that each vote is the product of the decision margin $|d_c(\mathbf{f})|$ and the probability $p_c(g)$.

B. Classifier Weights

In (6), all 15 classifiers cast votes without discrimination. However, it is reasonable to give a low weight to an inferior classifier. We determine the weight of each classifier by measuring the error rate. The error rate of a trained classifier can be estimated by classifying the training set itself. Specifically, we compute the error rate ϵ_c of classifier c by

$$\epsilon_c = \sum_{t \in \mathcal{T}} \max\left\{-d_c(\mathbf{f}_t)\psi_c(g_t), 0\right\} \tag{7}$$

where t is a sample in the training set $\mathcal{T}$, $\mathbf{f}_t$ is the feature vector of t, and g_t are the ground-truth genre of t. In the case of false classification, $d_c(\mathbf{f}_t)$ and $\psi_c(g_t)$ yield opposite signs, and their absolute product is added to the error rate.

We determine the weight w_c of classifier c by

$$w_c = \frac{\exp\left(-\epsilon_c/\sigma\right)}{\sum_{n=1}^{15} \exp\left(-\epsilon_n/\sigma\right)}, \tag{8}$$

where σ controls the relative importance of classifiers. In this work, we fix $\sigma = 1.5$. Then, the voting rule in (6) is modified to the weighted voting rule, which is given by

$$v_{\text{weight}}(\mathbf{f}, g) = \sum_{c=1}^{15} w_c \cdot p_c(g) \cdot \max\{d_c(\mathbf{f})\psi_c(g), 0\}. \tag{9}$$

Finally, we classify the query feature vector $\mathbf{f}$ into the optimal genre g^* with the maximum votes, which is given by

$$g^* = \arg\max_{g} v_{\text{weight}}(\mathbf{f}, g). \tag{10}$$

C. Divide-and-Conquer Voting

A part of a video may contain noisy features, which degrade the classification performance. This problem can be alleviated by dividing a video into sub-videos and extracting features from the sub-videos independently. The independent feature extraction makes the voting results from the sub-videos

TABLE III: Comparison of genre classification accuracies. By employing the proposed features in addition to the baseline features, the accuracy increases for all genres. A bold-faced number denotes the higher accuracy value in each genre.

Features	Genre classification accuracy (%)					
	Animation	Commercial	Entertainment	Drama	Sports	Total
Baseline features	75.0	88.5	71.2	85.8	82.7	80.6
Baseline features + Proposed features	**88.2**	**91.0**	**84.8**	**86.7**	**86.5**	**87.7**

TABLE IV: Performance comparison of the proposed classifier with the multi-class SVM [7] and the hierarchical SVM [3]. A bold-faced number denotes the highest accuracy in each genre.

Classifier	Genre classification accuracy (%)					
	Animation	Commercial	Entertainment	Drama	Sports	Total
Multi-class SVM + Feature transform	81.3	87.2	68.2	83.3	76.9	79.7
Hierarchical SVM + Feature transform	80.6	89.7	70.5	79.2	76.9	79.9
Voting SVM	87.5	**91.0**	78.8	85.0	81.7	85.2
Voting SVM + Feature transform	86.8	**91.0**	**84.8**	85.0	83.7	86.6
Voting SVM + Classifier weighting	87.5	89.1	82.6	**86.7**	85.6	86.4
Voting SVM + Feature transform + Classifier weighting	**88.2**	**91.0**	**84.8**	**86.7**	**86.5**	**87.7**

also independent. By accumulating the votes from the sub-videos, we can filter out the effects of outlier features.

In this work, we divide a video into four sub-videos, accumulate votes from each sub-video, and find the genre with the maximum votes. This divide-and-conquer voting not only achieves robust genre classification, but also is suitable for real-time applications in which the genres of video segments (*e.g.* from a television channel) are updated continuously. Since a short sub-video may contain insufficient information, we apply the divide-and-conquer voting only to videos with durations longer than 110 seconds.

IV. Experimental Results

We train the proposed genre classification algorithm using 1200 video clips, and then evaluate its performance on 656 test video clips. The training videos are not used for the test videos. Both training and test videos are collected from United States, Republic of Korea, and Japan, and they are resized to the resolution of 640×360. Table II lists the properties of these videos. Fig. 7 shows several example videos in each genre. To implement the multiple SVMs, we use the LIBSVM package [13], and set the kernel $k(\mathbf{f}, \mathbf{x}_l)$ in (4) to be the radial basis function $\exp(-\|\mathbf{f}-\mathbf{x}_l\|^2/\sigma_k)$, where σ_k is the number of features as recommended in [13].

A. Efficacy of Proposed Features

We first evaluate the efficacy of the proposed features. Table III compares the accuracy of the classifier, which employs both baseline and proposed features, with that using the baseline features only. In both cases, we apply the same classifier, proposed in Section III. The accuracy of a classifier is defined as

$$\text{Accuracy} = \frac{\text{Number of correctly classified test videos}}{\text{Number of test videos}} \times 100. \tag{11}$$

We see that, by including the proposed features, the classification accuracy increases for all genres. Especially, for the entertainment genre, the addition of the proposed features improves the accuracy by 13.6%.

B. Genre Classification Performance

Next, we compare the performance of the proposed genre classification algorithm based on the voting SVM with those of the conventional algorithms: the multi-class SVM [7] and the hierarchical SVM [3]. For all classifiers, we employ both baseline and proposed features. We combine the feature transform and/or the classifier weighting scheme with the voting SVM. Note that the feature transform is applied for the multi-class SVM and hierarchical SVM as well.

Table IV compares the classification accuracies. We observe that the proposed voting SVM outperforms the multi-class SVM and hierarchical SVM for all 5 genres. In addition, the voting SVM further improves the classification accuracy when it is combined with the feature transform and the classifier weighting scheme. Notice that the proposed algorithm achieves the accuracy of about 88%, while the conventional algorithms yield the accuracies of about 80% only. Moreover, the proposed classifier yields the smallest performance variations according to the video genres, which means that it is not biased for specific genres.

Fig. 8 shows the confusion matrices of the multi-class SVM, the hierarchical SVM, and the proposed voting SVM with the feature transform and the classifier weighting. In Fig. 8(c), the diagonal entries are clearly red, whereas the other entries are evenly blue, as compared with Figs. 8 (a) and (b). This indicates that the proposed classifier provides significantly better performance than the conventional classifiers. More specifically, entertainment videos tend to have diverse characteristics, and thus they are usually confused with commercials and sports in the conventional classifiers. In contrast, the proposed classifier alleviates the confusion for the entertainment videos by employing the robust voting scheme.

V. Conclusions

In this paper, we proposed a robust classifier for the five video genres: animation, commercial, entertainment, drama, and sports, based on the voting from the multiple SVMs. We first investigated human cognitive characteristics and proposed efficient features for the video genre description. The proposed algorithm employs both baseline and proposed features and

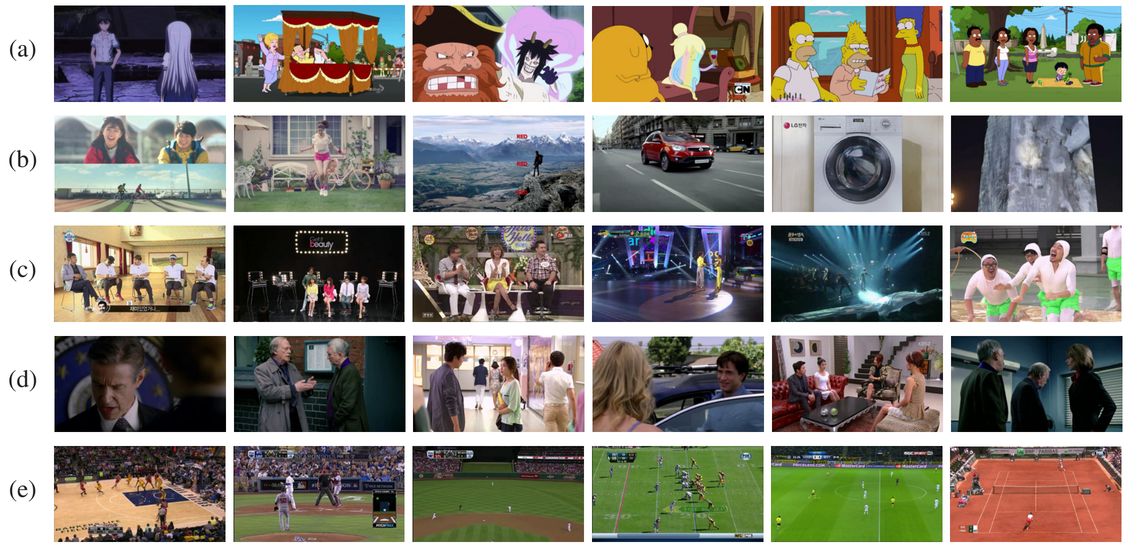


Fig. 7: Example videos in the five genres: (a) animation, (b) commercial, (c) entertainment, (d) drama, and (e) sports.

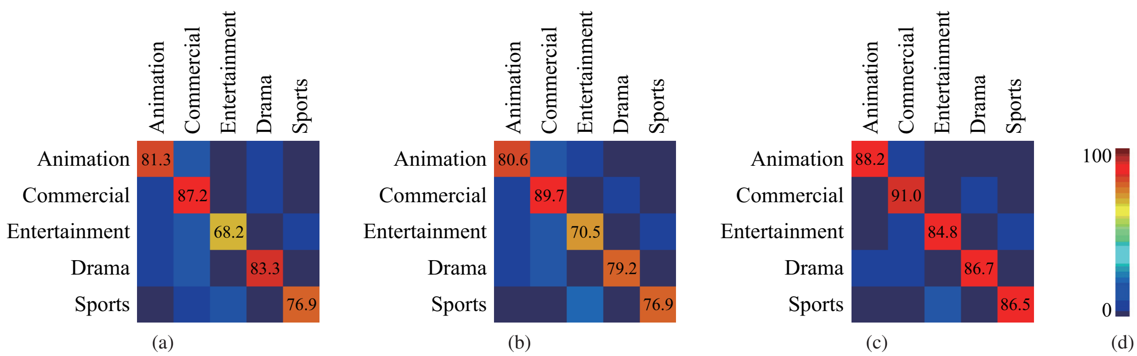


Fig. 8: Confusion matrices of (a) the multi-class SVM [7], (b) the hierarchical SVM [3], and (c) the proposed voting SVM: The rows and columns represent the real genres and the classified genres, respectively. The numbers in diagonal entries are the classification accuracies for the five genres, whereas the other entries are color-shaded according to the classification rates using the color bar in (d).

transforms them using the power law. Moreover, we developed the classification algorithm, which votes for the most probable genre for an input video using the multiple SVMs. Experimental results demonstrated that the proposed genre classifier achieves a significantly higher accuracy than the conventional classifiers.

References

[1] D. Borth, J. Hees, M. Koch, A. Ulges, C. Schulze, T. Breuel, and R. Paredes, "TubeFiler: An automatic web video categorizer," in *Proc. ACM Multimedia*, Oct. 2009.

[2] J. Wu and M. Worring, "Efficient genre-specific semantic video indexing," *IEEE Trans. Multimedia*, vol. 14, no. 2, pp. 291–302, Apr. 2012.

[3] X. Yuan, W. Lai, T. Mei, X.-S. Hua, X.-Q. Wu, and S. Li, "Automatic video genre categorization using hierarchical SVM," in *Proc. IEEE ICIP*, Oct. 2006.

[4] J. Wang, C. Xu, and E. Chng, "Automatic sports video genre classification using Pseudo-2D-HMM," in *Proc. ICPR*, Aug. 2006.

[5] A. Kowdle, K.-W. Chang, and T. Chen, "Video categorization using object of interest detection," in *Proc. IEEE ICIP*, Sep. 2010.

[6] P. Xu, D. M. J. Tax, and A. Hanjalic, "A structure-based video representation for web video categorization," in *Proc. ICPR*, Nov. 2012.

[7] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, Sep. 1995.

[8] D. Brezeale and D. J. Cook, "Automatic video classification: A survey of the literature," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 38, no. 3, pp. 416–430, May 2008.

[9] S. Avidan, "Support vector tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 8, pp. 1064–1072, Aug. 2004.

[10] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE CVPR*, Jun. 2005.

[11] C. Yang, M. Dong, and J. Hua, "Region-based image annotation using asymmetrical support vector machine-based multiple-instance learning," in *Proc. IEEE CVPR*, Jun. 2006.

[12] N. Kumar, A. C. Berg, P. N. Belhumeur, and S. K. Nayar, "Attribute and simile classifiers for face verification," in *Proc. IEEE CVPR*, Sep. 2009.

[13] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. Intell. Syst. and Technol.*, vol. 2, no. 3, pp. 27:1–27:27, Apr. 2011, software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.