Video Processing and Analysis Through Optimum-Path Forest

Guilherme B. Martins
Department of Computing
São Paulo State University – UNESP
Bauru - SP, Brazil
gui.bmartins3@gmail.com

Jurandy Almeida
Institute of Science and Technology
Federal University of São Paulo – UNIFESP
São José dos Campos - SP, Brazil
jurandy.almeida@unifesp.br

João P. Papa
Department of Computing
São Paulo State University – UNESP
Bauru - SP, Brazil
papa@fc.unesp.br

Abstract—**Currently, a number of improvements related to computational networks and data storage technologies have allowed a considerable amount of digital content to be provided on the Internet, mainly through social networks. In order to exploit this context, video processing and pattern recognition approaches have received a considerable attention in the last years. The main goal of this work is to employ the Optimum-Path Forest classifier in both video summarization and video genre classification processes as well as to conduct a viability study of such classifier in the aforementioned contexts. The results have shown this classifier can achieve promising performances, being very close in terms of summary quality and consistent recognition rates to some state-of-the-art video summarization and video genre classification approaches, respectively.**

I. Introduction

Techniques for video summarization are commonly classified in static or dynamic ones. The main goal of the former methodologies is to obtain keyframes of the original video in order to compose the compressed representation, whereas the dynamic techniques aim at finding out a collection of segments (set of frames nearby the keyframes) to provide more reasonable summaries, which can also include sound effects.

A considerable number of works that deal with video summarization can be referred in the literature, being most of them machine learning-oriented. The reason is that video summarization aims at extracting features from frames, for further clustering them in order to group frames with similar content. After that, the most representative sample from each cluster is then elected as the *keyframe*, i.e., the one that shall compose the final video summary. Almeida et al. [1], for instance, proposed the VISON, which works on compressed videos to allow a fast and effective design of video summaries. Avila et al. [2] presented VSUMM, a video summarization approach based on color information and k-means, which works well in several public datasets, and Papadopoulos et al. [3] applied a Self-Organized Neural Gas network to produce video summaries, which is able to compute dynamically the number of clusters, each one containing a possible keyframe candidate.

Another common situation involving digital videos is the need for classification methods capable of allowing a faster retrieval response when searching for specific content in video databases. One way to perform this task is through video genre classification, which aims at finding or predicting a corresponding genre for a given video sequence. Huang and Wang [4], for instance, employed the well-known Support Vector Machines (SVMs) together with a Self-Adaptive Harmony Search optimization algorithm to classify movie genres, and Karpathy et al. [5] employed a Convolutional Neural Network for the same purpose.

Some years ago, Papa et al. [6], [7] proposed the Optimum-Path Forest (OPF) classifier, which models the pattern recognition task as a graph partition problem. Basically, the dataset samples (feature vectors) are represented by graph nodes, which are connected to each other through an adjacency relation. After that, some key nodes (*prototypes*) rule a competition process among themselves in order to conquer the remaining samples offering to them optimum-path costs. When a sample is conquered, it receives the very same label of its conqueror, as well as the cost it has been offered.

An interesting unsupervised OPF variation was proposed by Rocha et al. [8] to resolve problems that demand a clustering resolution. Basically, it is a graph-based approach with the same basic rules defined in the supervised OPF version, except it considers a k-nearest neighbors (k-nn) graph as main adjacency relation. It also includes a pdf computation to find density values for each sample and to determine their corresponding prototypes, and minimization cost function to set the optimum-path trees, consequently partitioning the dataset into clusters rooted at prototype nodes.

Therefore, the main goal of this work[1] is to introduce OPF for video classification and static summarization tasks using image and video properties obtained from different descriptors, as well as to compare OPF against with some state-of-the-art pattern recognition techniques for each aforementioned task.

II. Optimum-Path Forest

In this section, we present the theoretical background about Optimum-Path Forest. First, we describe the OPF classifier proposed by Papa et al. [6], [7] for video genre classification (Section II-A), and further its unsupervised variant [8] to resolve clustering problems during static video summarization (Section II-B).

[1] This work relates to a M.Sc. dissertation.

A. *Supervised learning*

Let $\mathcal{D} = \mathcal{D}_1 \cup \mathcal{D}_2$ be a labeled dataset, such that $\mathcal{D}_1$ and $\mathcal{D}_2$ stands for the training and test sets, respectively. Let $\mathcal{S} \subset \mathcal{D}_1$ be a set of prototypes of all classes (i.e., key samples that best represent the classes). Let $(\mathcal{D}_1, A)$ be a complete graph whose nodes are the samples in $\mathcal{D}_1$, and any pair of samples defines an arc in $A = \mathcal{D}_1 \times \mathcal{D}_1$[2]. Additionally, let π_s be a path in $(\mathcal{D}_1, A)$ with terminus at sample $s \in D_1$.

The OPF algorithm employs the path-cost function f_{max} due to its theoretical properties for estimating prototypes (Section II-A1 gives further details about this procedure):

$$\begin{aligned} f_{max}(\langle s \rangle) &= \begin{cases} 0 & \text{if } s \in S \\ +\infty & \text{otherwise,} \end{cases} \\ f_{max}(\pi_s \cdot \langle s,t \rangle) &= \max\{f_{max}(\pi_s), d(s,t)\}, \end{aligned} \quad (1)$$

where $d(s,t)$ stands for a distance between nodes s and t, such that $s,t \in \mathcal{D}_1$. Therefore, $f_{max}(\pi_s)$ computes the maximum distance between adjacent samples in π_s, when π_s is not a trivial path. In short, the OPF algorithm tries to minimize $f_{max}(\pi_t)$, $\forall t \in \mathcal{D}_1$.

1) Training: We say that $\mathcal{S}^*$ is an optimum set of prototypes when OPF algorithm minimizes the classification errors for every $s \in \mathcal{D}_1$. We have that S^* can be found by exploiting the theoretical relation between the minimum-spanning tree and the optimum-path tree for f_{max}. The training essentially consists of finding S^* and an OPF classifier rooted at S^*. By computing a Minimum Spanning Tree (MST) in the complete graph $(\mathcal{D}_1, A)$, one obtain a connected acyclic graph whose nodes are all samples of $\mathcal{D}_1$ and the arcs are undirected and weighted by the distances d between adjacent samples. In the MST, every pair of samples is connected by a single path, which is optimum according to f_{max}. Hence, the minimum-spanning tree contains one optimum-path tree for any selected root node.

The optimum prototypes are the closest elements of the MST with different labels in $\mathcal{D}_1$ (i.e., elements that fall in the frontier of the classes). By removing the arcs between different classes, their adjacent samples become prototypes in S^*, and the OPF algorithm can define an optimum-path forest with minimum classification errors in $\mathcal{D}_1$.

2) Classification: For any sample $t \in \mathcal{D}_2$, we consider all arcs connecting t with samples $s \in \mathcal{D}_1$, as though t were part of the training graph. Considering all possible paths from S^* to t, we find the optimum path $P^*(t)$ from S^* and label t with the class $\lambda(R(t))$ of its most strongly connected prototype $R(t) \in S^*$. This path can be identified incrementally, by evaluating the optimum cost $C(t)$ as follows:

$$C(t) = \min\{\max\{C(s), d(s,t)\}\}, \ \forall s \in \mathcal{D}_1. \quad (2)$$

Let the node $s^* \in \mathcal{D}_1$ be the one that satisfies Equation 2 (i.e., the predecessor $P(t)$ in the optimum path $P^*(t)$). Given that $L(s^*) = \lambda(R(t))$, the classification simply assigns $L(s^*)$ as the class of t, where $L(\cdot)$ is a function that assigns the true label to a given sample. An error occurs when $L(s^*) \neq \lambda(t)$.

B. *Unsupervised learning*

Let $\mathcal{D}$ be an unlabeled dataset such that for every sample $s \in \mathcal{D}$ there is a feature vector $\vec{v}(s)$. The fundamental problem in data clustering is to identify natural groups in $\mathcal{D}$. A graph $(\mathcal{D}, \mathcal{A})$ is defined such that the arcs $(s,t) \in \mathcal{A}$ connect k-nearest neighbors in the feature space. The arcs are weighted by $d(s,t)$ and the nodes $s \in \mathcal{D}$ are weighted by a density value $\rho(s)$, given by:

$$\rho(s) = \frac{1}{\sqrt{2\pi\sigma^2}|\mathcal{A}(s)|} \sum_{\forall t \in \mathcal{A}(s)} \exp\left(\frac{-d^2(s,t)}{2\sigma^2}\right), \quad (3)$$

where $|\mathcal{A}(s)| = k$, $\sigma = \frac{d_f}{3}$, and d_f is the maximum arc weight in $(\mathcal{D}, \mathcal{A})$. This parameter choice considers all nodes for density computation, since a Gaussian function covers most samples within $d(s,t) \in [0, 3\sigma]$. By taking into account the k-nearest neighbors, unsupervised OPF handles different concentrations and reduces the scale problem to the one of finding the best value of k within $[1, k_{\max}]$, for $1 \leq k_{\max} \leq |\mathcal{D}|$. The solution provided by Rocha et al. [8] considers the minimum graph cut provided by the clustering results for $k \in [1, k_{\max}]$, according to a measure suggested by Shi and Malik based on graph cuts [9].

Among all possible paths π_t with roots on the maxima of the pdf, unsupervised OPF finds a path whose the lowest density value along it is maximum. Each maximum should then define an influence zone (cluster) by selecting the samples that are more strongly connected to it, according to this definition, than to any other maximum. More formally, we wish to maximize $f(\pi_t)$ for all $t \in \mathcal{N}$ where

$$\begin{aligned} f(\langle t \rangle) &= \begin{cases} \rho(t) & \text{if } t \in \mathcal{R} \\ \rho(t) - \delta & \text{otherwise} \end{cases} \\ f(\langle \pi_s \cdot \langle s,t \rangle \rangle) &= \min\{f(\pi_s), \rho(t)\} \end{aligned} \quad (4)$$

for $\delta = \min_{\forall (s,t) \in \mathcal{A} | \rho(t) \neq \rho(s)} |\rho(t) - \rho(s)|$ and $\mathcal{R}$ being a root set with one element for each maximum of the pdf. Higher values of delta reduce the number of maxima. We are setting $\delta = 1.0$ and scaling real numbers $\rho(t) \in [1, 1000]$ in this work. The OPF algorithm maximizes $f(\pi_t)$ such that the optimum paths form an optimum-path forest — a predecessor map P with no cycles that assigns to each sample $t \notin \mathcal{R}$ its predecessor $P(t)$ in the optimum path from $\mathcal{R}$ or a marker nil when $t \in \mathcal{R}$. In essence, each maximum of the pdf, i.e., prototype, will be the root of an optimum-path tree - OPT (cluster), and the collection of all OPTs originates the optimum-path forest.

III. Proposed approaches

A. *OPF clustering for static video summarization*

The proposed approach based on OPF to obtain static video summaries, was structured into six steps: (*i*) video sampling, (*ii*) feature extraction, (*iii*) removal of meaningless frames, (*iv*) clustering, (*v*) removal of redundant keyframes, and (*vi*) video summary generation.

The first step uses a pre-sampling approach for extracting frames from the videos to be summarized. The *video sampling*

[2] The arcs are weighted by the distance between their corresponding nodes.

was performed by the well-known *ffmpeg* tool[3] in a sampling rate of one frame per second in two public datasets described in Section IV-A1.

The second step performs the *feature extraction* from each frame extracted in the previous step. To do this task, we considered the following descriptors: Auto Color Correlogram (ACC), Color Coherent Vector (CCV), Border/Interior pixel Classification (BIC), and Global Color Histogram (GCH) for encoding color information; Generic Fourier Descriptor (GFD) and Haar-Wavelet Descriptor (HWD) for analyzing spectral properties. In addition, we built a Bag-of-Features (BoF) representation using SIFT (Scale-Invariant Feature Transform) features. For that, we constructed a visual dictionary using k-Means with $k = 4000$ visual words, where k value was empirically chosen. They were selected based on the comparative study upon algorithms to better describe digital images conducted by Penatti et al. [10].

Further, we performed the removal of meaningless frames from the feature-based dataset aiming at avoiding unnecessary frames during the clustering process. Note that a meaningless frame is the one whose image is composed of a single color (i.e., full black or white frames) due to a fade-in or fade-out effects. Therefore, such frame is then removed from the feature-based dataset only if the color variance of its quantized image is equal to zero [1].

In the third step, OPF computes the clusters from the feature-based dataset aiming at finding the most representative frames on each cluster (keyframes). Since OPF finds the prototypes in the regions with highest density, they tend to be located at the clusters center, thus being good candidates to become keyframes. In order to improve the clusters computation, we considered the contributions presented in [11], i.e., the partitioning of the feature-based dataset into small subsets, and the adhibition of a modified Euclidean distance function able to consider more temporal information during the computation of the “distance” among frames.

Even after the clustering be performed on each subset, one can also have small clusters, which means they may not contribute with relevant information to the final video summary. In order to remove such non-relevant clusters, we compute the average cluster size for each subset, and then we keep the clusters whose size (number of samples that belong to it) is greater than the half of the average cluster size [2]. Soon after, we then extract one keyframe from each remaining cluster (*keycluster*), being such keyframe the prototype of that cluster. The collection of all keyframes composes the final frame set.

The fourth step is responsible for removing redundant keyframes from the frame set obtained in the previous phase. This process is described as follows: each keyframe is compared against all other keyframes using the Euclidean distance. If the resulting distance is smaller than 0.15, this keyframe is considered irrelevant, thus being removed from the summary. The threshold used for comparison purposes was selected empirically.

In the final step, the keyframes are chronologically ordered to generate the video summary. Therefore, the final static summary can now be used for comparison purposes against others.

B. Supervised OPF for video genre classification

The strategy adopted for supervised video classification was built in two main steps: (i) video features extraction and (ii) OPF supervised classification.

We employed three main approaches to extract video visual properties: “Bag-of-Visual-Words” [12], “Bag-of-Scenes” [13] and “Histogram of Motion Patterns” [14]. The former two approaches are based on video frames and disregard transitions between them, whereas the latter one is based on motion information, and it considers the transitions between video frames aiming to better preserve it. Basically, they were chosen due to be new description image approaches which aim to enhance generic video representation.

Concerning about the classification step, we considered the supervised version of OPF using complete graph adjacent relation to classify videos into their respective genres (classes).

IV. Experiments

For both proposed approaches, the experiments were conducted following the same sequence structure. First, we performed OPF on videos belonging to public databases using different configurations. Then, we compared OPF results against other video summarization and genre classification techniques through some proper evaluation methodology.

A. Static video summarization

1) Datasets: The video summarization experiments were performed using two public video datasets[4]: Open Video and Youtube. The former contains 50 videos randomly selected from the Open Video Project [5], which are distributed among three different genres (i.e., documentary, educational, and lecture) and their duration varies from 1 to 4 minutes. The latter is composed of 40 videos collected from the Youtube [6], which are distributed among five genres (i.e., sports, news, tv-shows, commercials, and home videos) and their duration varies from 1 to 10 minutes.

2) Evaluation and experimental results: In this work, we adopted a subjective evaluation method to assess the quality of video summaries, known as *Comparison of User Summaries* (CUS) [2]: initially, the subjects are asked to watch the whole video, and further they are oriented to freely select a subset of frames able to summarize the video content. Finally, their summaries are compared to the automatic summaries provided by the algorithms through pixel-wise matching method proposed by Almeida et al. [15], which led us to the number of frames gathered. The standard measures *precision* and *recall*

[3] http://www.ffmpeg.org/

[4] http://sites.google.com/site/vsummsite/

[5] http://www.open-video.org/

[6] http://www.youtube.com/

can then be used to evaluate the automatic summary, being precision the ratio of the number of matching frames to the total number of frames in the automatic summary. Recall is the ratio of the number of matching frames to the total number of frames in the user summary.

In this paper, we chose F-measure as the metric to evaluate performance since it presents an interesting trade-off between precision and recall. The increase of one value decreases the second, and vice-versa, which makes F-measure a suitable choice to evaluate the approaches considered in this work.

Before comparing OPF against other video summarization techniques using F-measure, it was necessary to understand OPF parameters behavior in order to select the best set-up to improve OPF performance. Concerning about that, we established two OPF versions:

1) OPF: original form of OPF clustering, which considers Euclidean distance on the clusters computation using no partitioned dataset;
2) OPF*: On clusters computation it considers the Temporal distance described in Section III-A and uses dataset partition.

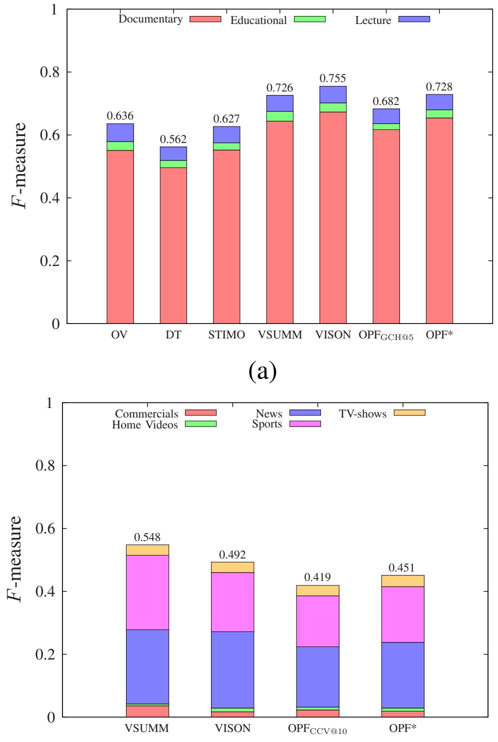
For both versions, OPF needs to be correctly setup. As aforementioned, OPF computes clusters on-the-fly based on optimum paths and a variable k_{max} (Section II-B), which defines the maximum number of nearest neighbors to be considered during the cluster computation. Although the reader may argue that the algorithm does not compute clusters fully automatically, it is important to highlight changing the k_{max} value causes less impact on the final result than varying the value of k for k-means, for instance.

In a first moment, for each feature-based dataset on both OPF versions, we evaluated k_{max} value within the range $[5, 50]$ with steps of 5. Specially to OPF*, we incremented this evaluation considering different subset size percentages (15%, 20%, 25%, 30%, 35%, 40%, 50% and 60%). Finally, we select the set of parameters that maximizes F-measure on Open Video and Youtube video datasets. The set-ups chosen for OPF and OPF* are shown in Table I.

TABLE I
BEST SETUP CHOSEN FOR OPF AND OPF*.

	Dataset	k_{max}	Descriptor	Subset size
OPF	Open Video	5	GCH	-
	Youtube	10	CCV	-
OPF*	Open Video	5	GCH	25%
	Youtube	5	CCV	25%

The second moment was responsible for comparing OPF against the results reported by five known static video summarization techniques. In the Open Video dataset, we compared both OPF variations against DT [16], STIMO [17], VSUMM [2], VISON [1] and Open Video (OV)[7]. On the other hand, in the YouTube dataset, OPF and OPF* were compared only against VSUMM and VISON due to the lack of techniques performed on the aforementioned video dataset.



(a)

(b)

Fig. 1. Mean F-measure achieved by different approaches on each video category for (a) Open Video and (b) Youtube datasets. Extracted from [11].

Figure 1 shows the F-measure values for all techniques and datasets considered in this paper. Clearly, OPF* obtained more accurate results than OPF for both datasets, as well as it has been the second best technique considering Open Video dataset (Figure 1a). Additionally, it has been placed as the third more accurate technique in the YouTube dataset (Figure 1b). However, the best technique in YouTube dataset uses k-means for clustering purposes, thus requiring the number of clusters beforehand. Note that information is not a main concern regarding OPF-based techniques. It is worth noting to stress OPF requires less user interaction than VISON technique as well, since it has some user parameters.

We observed OPF* seems to work better with smaller subsets, since larger ones do not favor the temporal information. In our experiments, we observed that small values for α [8] did not contribute a lot for the final results. In regard to Open Video dataset, OPF* achieved better results than OPF concerning the “Documentary” and “Educational” videos. The rationale behind that concerns with the fact that “Educational” videos contain similar frames but at different temporal positions in the video. Imagine some lecturer teaching a specific subject, and further we may have some pictorial explanation about that, and once again the teacher gets focused again in the video.

[7] Storyboards generated using the algorithm of DeMenthon et al. [18] and refined using some manual intervention to obtains better results.

[8] Relaxation term that weights the amount of temporal information.

Although we have quite spatial-similar frames, they are placed at different temporal positions within the video.

With respect to Youtube dataset, the best improvement regarding OPF* concerns with “Sports” videos, which are also expected to cover similar situations that have near-spatial frames, such as the best moments from a soccer game, for instance. Since we used color descriptors, it is very likely from this point of view that different soccer games seem similar to each other. Once again, the temporal information played an important role in this situation.

B. Supervised video genre classification

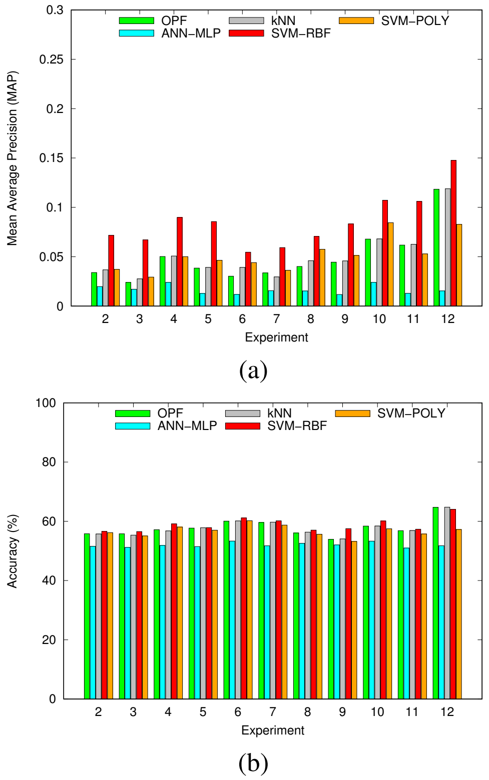
1) Dataset: In this work, we employed a benchmarking dataset provided by the MediaEval 2012 organizers for the Genre Tagging Task [19]. The dataset is composed of 14,838 videos divided into a development set (5,288 videos) and a test set (9,550 videos), comprising a total of 3,288 hours of video data. All the video sequences were collected from the blip.tv[9], and they are distributed among 26 video genre categories assigned by the media platform of the referred web site.

2) Evaluation and experimental results: To assess the robustness of OPF classifier, it was compared against four well-known classifiers: Artificial Neural Network with Multilayer Perceptron (ANN-MLP), k-Nearest Neighbors (k-NN) and two variations of Support Vector Machines (SVM): the first using a polynomial (SVM-POLY) kernel and the second using a Radial Basis Function (SVM-RBF) kernel. It is noteworthy SVM parameters have been optimized through cross-validation.

For each classifier, we performed 12 different experiments[10] considering the visual features encoded with BoVW, BoS, and HMP. In order to evaluate the results, we considered two performance measures: (i) the Mean Average Precision (MAP) and (ii) a recognition rate proposed by Papa et al. [7], which considers unbalanced data, as well as we compute the computational load for both training and classification steps. Figures 2(a) and 2(b) depict the results considering MAP and accuracy measures, respectively.

In regard to both measures, the experiment number #12, i.e., HMP video descriptor using 6075 motion patterns, has showed the best results for all classifiers (except for ANN), which might be due to the robustness of HMP to several transformations, besides being suitable for very large collections of video data [14], which is in accordance with the MediaEval 2012 dataset used in our experiments. In terms of MAP, OPF has been placed in second or third in most cases, while for the accuracy measure OPF obtained the first or second position in most part of the experiments.

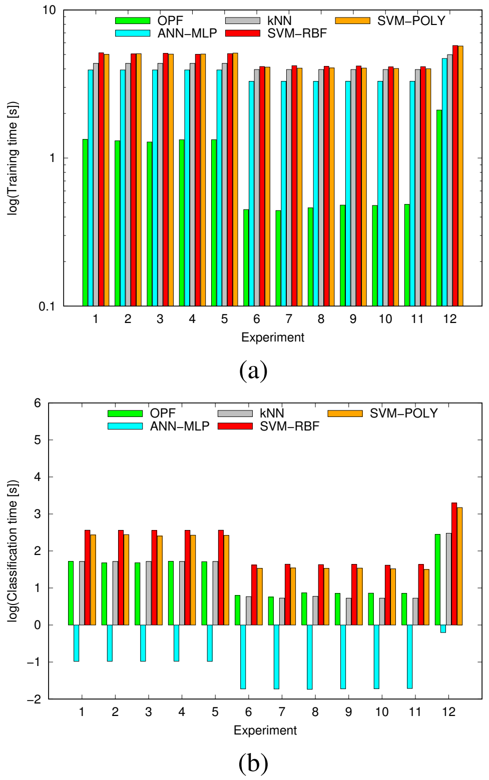
If we consider the computational load displayed in Figures 3(a) and 3(b) for the training and test steps, respectively, we shall observe OPF has been the fastest classifier for training in almost all experiments, as well as the second fastest classifier considering the classification time.



(a)

(b)

Fig. 2. Recognition results in terms of (a) MAP and (b) accuracy measures.



(a)

(b)

Fig. 3. Computational load for (a) training and (b) classification steps.

In light of those results, we may conclude OPF is a suitable

[9] http://blip.tv (as of May, 2015).

[10] Consult [20] for details about all experimental setup.

technique for the automatic classification of videos based on visual information, since it has obtained good recognition rates in a smaller amount of time when compared to the other techniques (except for ANN-MLP). Such skill might be very interesting in online classification and recommendation systems, in which a high trade-off between effectiveness and efficiency is extremely desired.

V. Conclusions

In this work, we introduced the Optimum-path Forest classifier in the context of video processing. While OPF clustering was used for static video summarization, the supervised version of the aforementioned classifier was employed to classify videos based on their genres.

The proposed approach for video summarization achieved promising results, mainly due to the changes we did partitioning the dataset into smaller subsets and using a different distance function aiming to consider both spatial and temporal information from videos. Consequently, we obtained results very competitive to some state-of-the-art techniques for static video summarization in two public datasets.

With respect to video genre classification, we considered supervised OPF classifier against four classification techniques using three approaches for video description setup with different test configurations. In our analysis, OPF obtained good recognition rates (considering both MAP and accuracy) in all problems, as well as it required a low computational load for both training and classification steps when compared to the other classification techniques.

VI. Publications

This work obtained the following publications:

- *Static Video Summarization through Optimum-Path Forest*. Published in Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications: 19th Iberoamerican Congress, CIARP 2014 [21]. **(Qualis B1)**
- *Supervised Video Genre Classification using Optimum-Path Forest*. Published in Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications: 20th Iberoamerican Congress, CIARP 2015 [20]. **(Qualis B1)**
- *Temporal- and Spatial-Driven Video Summarization using Optimum-Path Forest*. Published in Graphics, Patterns and Images (SIBGRAPI), 2016 29th SIBGRAPI Conference [11]. **(Qualis B1)**
- *OPFSumm: On the Video Summarization Using Optimum-Path Forest*. Under final revision process for publication in Elsevier's Pattern Recognition Letters. **(Qualis A1)**

Acknowledgment

The authors are grateful to Capes, CNPq grants #470571/2013-6 and #306166/2014-3, and FAPESP grant #2014/16250-9. It is noteworthy this work contains the synthesis of former publications.

References

[1] J. Almeida, N. J. Leite, and R. S. Torres, "VISON: VIdeo Summarization for ONline applications," *Pattern Recognition Letters*, vol. 33, no. 4, pp. 397–409, 2012.

[2] S. E. F. Avila, A. P. B. Lopes, A. Luz Jr., and A. A. Araújo, "VSUMM: A mechanism designed to produce static video summaries and a novel evaluation method," *Pattern Recognition Letters*, vol. 32, no. 1, pp. 56–68, 2011.

[3] D. P. Papadopoulos, A. A. Chatzichristofis, and N. Papamarkos, "5th international conference on computer vision/computer graphics collaboration techniques," ser. Lecture Notes in Computer Science, A. Gagalowicz and W. Philips, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, vol. 6930, ch. Video Summarization Using a Self-Growing and Self-Organized Neural Gas Network, pp. 216–226.

[4] Y.-F. Huang and S.-H. Wang, "Movie genre classification using svm with audio and video features," in *Active Media Technology*, ser. Lecture Notes in Computer Science, R. Huang, A. A. Ghorbani, G. Pasi, T. Yamaguchi, N. Y. Yen, and B. Jin, Eds. Springer Berlin Heidelberg, 2012, vol. 7669, pp. 1–10.

[5] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. F.-F., "Large-scale video classification with convolutional neural networks," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1725–1732.

[6] J. P. Papa, A. X. Falcão, V. H. C. Albuquerque, and J. M. R. S. Tavares, "Efficient supervised optimum-path forest classification for large datasets," *Pattern Recognition*, vol. 45, no. 1, pp. 512–520, 2012.

[7] J. P. Papa, A. X. Falcão, and C. T. N. Suzuki, "Supervised pattern classification based on optimum-path forest," *International Journal of Imaging Systems and Technology*, vol. 19, no. 2, pp. 120–131, 2009.

[8] L. M. Rocha, F. A. M. Cappabianco, and A. X. Falcão, "Data clustering as an optimum-path forest problem with applications in image analysis," *International Journal of Imaging Systems and Technology*, vol. 19, no. 2, pp. 50–68, 2009.

[9] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, Aug 2000.

[10] O. A. B. Penatti, E. Valle, and R. S. Torres, "Comparative study of global color and texture descriptors for web image retrieval," *Journal of Visual Communication and Image Representation*, vol. 23, no. 2, pp. 359–380, 2012.

[11] G. B. Martins, J. P. Papa, and J. Almeida, "Temporal-and spatial-driven video summarization using optimum-path forest," in *29th SIBGRAPI Conference on Graphics, Patterns and Images*, 2016, pp. 335–339.

[12] Y.-L. Boureau, F. Bach, Y. LeCun, and J. Ponce, "Learning mid-level features for recognition," in *CVPR*, 2010, pp. 2559–2566.

[13] O. A. B. Penatti, L. T. Li, J. Almeida, and R. da S. Torres, "A visual approach for video geocoding using bag-of-scenes," in *ICMR*, 2012, pp. 1–8.

[14] J. Almeida, N. J. Leite, and R. S. Torres, "Comparison of video sequences with histograms of motion patterns," in *ICIP*, 2011, pp. 3673–3676.

[15] J. Almeida, R. S. Torres, and N. J. Leite, "Rapid video summarization on compressed video," in *IEEE Int. Symp. Multimedia (ISM'10)*, 2010, pp. 113–120.

[16] P. Mundur, Y. Rao, and Y. Yesha, "Keyframe-based video summarization using Delaunay clustering," *Int. J. on Digital Libraries*, vol. 6, no. 2, pp. 219–232, 2006.

[17] M. Furini, F. Geraci, M. Montangero, and M. Pellegrini, "STIMO: STIll and MOving video storyboard for the web scenario," *Multimedia Tools Appl.*, vol. 46, no. 1, pp. 47–69, 2010.

[18] D. DeMenthon, V. Kobla, and D. S. Doermann, "Video summarization by curve simplification," in *ACM Int. Conf. Multimedia (MM'08)*, 1998, pp. 211–218.

[19] S. Schmiedeke, C. Kofler, and I. Ferrané, "Overview of mediaeval 2012 genre tagging task," in *MediaEval*, 2012.

[20] G. B. Martins, J. Almeida, and J. P. Papa, "Supervised Video Genre Classification Using Optimum-Path Forest," in *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*. Springer, 2015, pp. 735–742.

[21] G. B. Martins, L. C. S., D. Osaku, J. G. Almeida, and J. P. Papa, in *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*. Springer, 2014, ch. Static Video Summarization through Optimum-Path Forest Clustering, pp. 893–900.