Efficient Thumbnail Identification through Object Recognition

Salvatore Carta[1], Eugenio Gaeta[2], Alessandro Giuliani[1], Leonardo Piano[1]
and Diego Reforgiato Recupero[1]
[1]*Department of Mathematics and Computer Science, University of Cagliari, Cagliari, Italy*
[2]*BuzzMyVideos, London, U.K.*

Keywords: Thumbnail Generation, Object Recognition, Machine Learning, YouTube.

Abstract: Given the overwhelming growth of online videos, providing suitable video thumbnails is important not only to influence user's browsing and searching experience, but also for companies involved in exploiting video sharing portals (YouTube, in our work) for their business activities (e.g., advertising). A main requirement for automated thumbnail generation frameworks is to be highly reliable and time-efficient, and, at the same time, economic in terms of computational efforts. As conventional methods often fail to produce satisfying results, video thumbnail generation is a challenging research topic. In this paper, we propose two novel approaches able to provide relevant thumbnails with the minimum effort in terms of time execution and computational resources. The proposals rely on an object recognition framework which captures the most topic-related frames of a video, and selects the thumbnails from its resulting frames set. Our approach is a trade-off between content-coverage and time-efficiency. We perform preliminary experiments aimed at assessing and validating our models, and we compare them with a baseline compliant to the state-of-the-art. The assessments confirm our expectations, and encourage the future improvement of the proposed algorithms, as our proposals are significantly faster and more accurate than the baseline.

1 INTRODUCTION

Recently, the flourishing popularity of social networks and video sharing websites produced a massive growth of videos uploading. The continuous demand of video accessing and uploading entailed the requirement of providing reliable and fast sharing services able to propose appealing videos to users. The most known video sharing website is YouTube[1], which is is the second most-visited site in the world, only behind Google. It represents the epicenter of content creation, search engine optimization, and video content marketing. More than *500 hours of video* are uploaded to YouTube *every minute*, and more than *1 billion hours of YouTube videos* are watched *every day*[2]. In this context, let us consider video sharing from a further perspective: the popularity of YouTube, recently, has captured the interest of businesses, small and large, which currently adopt the platform to promote their projects, expose their brands, and naturally monetize. Indeed, a successful marketing activity is based on exploiting social popularity of videos, as higher popularity usually means stronger influence, which translates into higher revenues. In turn, improving video popularity is a key point for video uploaders and channel creators to increase the probability that their videos would be selected by advertisers. To improve users' video searching experiences, a video hosting website typically allows and suggests users to attach metadata (e.g., description or keywords) to the video. However, this task may prove to be challenging for users, notably in the common case of using mobile devices (Ames and Naaman, 2007). Therefore, a hot research topic is devising algorithms able to automatically perform video optimization, which refers to all strategies aimed at increasing its probability of being highly indexed by search engines and, consequently, the probability of being viewed by users. Given a video, among its video optimization activities, generating an appropriate "sketch", either in the form of text or image, plays an essential role. In particular, a common way to provide viewers a straightforward and concise representation of video contents is to generate appropri-

[1]https://www.youtube.com
[2]https://www.omnicoreagency.com/youtube-statistics/

Carta, S., Gaeta, E., Giuliani, A., Piano, L. and Recupero, D.
Efficient Thumbnail Identification through Object Recognition.
DOI: 10.5220/0010108802090216
In *Proceedings of the 16th International Conference on Web Information Systems and Technologies (WEBIST 2020)*, pages 209-216
ISBN: 978-989-758-478-7

ate thumbnails (Liu et al., 2015; Song et al., 2016), a thumbnail being a frame aimed at providing a visual snapshot of the video.

In this paper, we define two novel approaches of video thumbnail generation aimed at suggesting relevant thumbnails in fast execution time using a less amount of computational resources. Our proposal is based on the adoption of a pre-trained object recognition framework aimed at capturing the most topic-related frames in a video, and on the selection of thumbnails from its resulting frames set. Our approach is a trade-off between content-coverage and time-efficiency. The novelty of the proposal is that we suggest thumbnails in a "dynamic" way (see Section 3), which permits to significantly decrease the use of resources. Results highlight that the performances are higher than the classic methods.

The rest of the paper is organized as follows: Section 2 describes the background and the related work in thumbnail generation; Section 3 regards all the details on the proposed approach, together with the chosen baseline. In Section 4 we report all the experiments we have carried out, which are discussed in Section 5. Section 6 ends the paper with conclusions and future directions where we are headed.

2 BACKGROUND

Traditionally, thumbnail images were generated either manually or automatically. The former approaches involve intensive manual effort, as a video contains hundreds or thousands of frames. The latter were often based on the selection of random fixed frames (e.g., the first or the middle frame), with the goal of suggesting thumbnails in a fast and immediate way. Such approaches convey to often suggest meaningless images being unable to provide a clear topic of the video. To this end, researches focused on automated thumbnail generation by adopting machine learning methods. First attempts relied on assigning a thumbnail by identifying a single key-frame in the video. However, a unique thumbnail is quite limited in its ability of representing the entirety of a video. Nevertheless, in real video applications only a fixed small number of key-frames should be proposed as thumbnails, as proposed by Wang et al. (Wang et al., 2018). This choice is motivated by two main issues: (i) the space limitation of UIs (specially for mobile devices) and (ii) the behavior of users in deciding whether to watch the video, i.e., the decision is taken relying on a few number of thumbnails. Consequently, selecting a limited set of images able to clearly represent the entire content of a video is a popular strategy in thumbnail generation, as it is a suitable trade-off between content-coverage and time-efficiency. Our algorithm, compliant with this choice, suggests a fixed number of thumbnails. To this end, machine learning-based methods able to identify a limited set of meaningful frames have massively been employed by more researchers (Liu et al., 2011; Li et al., 2014). Our proposal focuses on exploiting learned models, in particular relying on an object recognition framework. Moreover, several studies focused on learning models that are specific to different video domains (Zhang et al., 2016; Potapov et al., 2014). These approaches produce higher quality than unsupervised approaches that are blind to the video domain. To this end, we also focused on domain-dependent models, as reported in the following Sections.

Although many studies gave rise to perform also a semantic analysis of the video content, e.g. by using natural language processing (NLP) techniques Liu et al. (2015), a widespread methodology is still based on learning visual representativeness purely from visual content (Kang and Hua, 2005; Gao et al., 2009). In particular, high-level features like important object, people and subjects are often utilized (Rav-Acha et al., 2006; Lee et al., 2012). In accordance with such works, we focused, in this preliminary stage, on visual features, i.e., we adopt an object recognition framework to select the frames basing only on the visual contents of images.

The main motivation is that dealing with undesired time complexity and computational consumption is a current challenge for real-world applications. As a matter of fact, most companies (e.g., a small software house) cannot invest in hardware resources like big companies (e.g., Google or Microsoft). Hence, a frequent requirement is to perform thumbnail generation in a fast, economic and reliable way. For example, a considerable improvement on this line was made in (Song et al., 2016), where the system produces a thumbnail in only under 10% of video length, on a conventional laptop computer using a single CPU. The main novelty of our system is the capability of dealing with the problem of computational resources consumption, addressing it by relying on time-efficient tools and algorithms. We claim that our proposal is significantly faster than classic approaches, and provides better performances, as highlighted in the following Sections.

Furthermore, let us point out that another limitation of many proposed thumbnail generation methods is represented by don't consider if two or more selected frames belong to the same scene in the video. We also rely on scene identification. The novelty of our approach is also due to performing a dynamic

analysis of frame sequence for identifying scenes.

Let us finally remark that the most known and accessed video sharing portal is YouTube, which has recently become the leading focus also of research activities on video optimization. In so doing, we implemented a system able to access and analyze YouTube videos directly by a proper API (see Section 3).

3 METHODOLOGY

In the following sections, after giving a brief overview of thumbnail generation, we first introduce the baseline, and then we give the details of the proposed methodology.

3.1 Overview

A generic approach of video thumbnail extraction can be briefly described as follows. First, the video sequence is segmented into multiple "shots". The basic idea is to split the video by scene change detection algorithms. Typically, for each scene, only a single frame is taken as a keyframe. Among them, the most relevant keyframes are selected as thumbnails. A widespread technique to perform this task is to identify relevant objects contained in a given frame, usually exploiting object-detection frameworks.

Object Detection. Both baseline and our proposal rely on the adoption of object-detection frameworks aimed at identifying and assessing only frames containing relevant objects. To this end, we used YOLO (You Only Look Once)[3], a well known framework capable of identifying frames containing relevant objects. Unlike classifier-based approaches, YOLO is trained on a loss function that directly corresponds to detection performance, and the entire model is trained jointly (Redmon et al., 2015). The framework can be trained directly on full images.

A suitable functionality of Yolo is to analyze an image, and predict which objects it contains, associating a probability score σ_Y, which is taken into account as a measure of usefulness estimation of the frame.

3.2 Baseline

We compared our algorithms against a fast video thumbnail generation method which relies on a frame *pruning* process. Pruning consists of removing, from a sequence of video frames, images having a low level of quality. The method is proposed in two variants, differing on how the "quality" is computed. In particular, the first method relies on blurring each frame, while the variant relies on the colorfulness of the image. We named the former as **Blur-based Frame Pruning** approach (BFP) and the latter as **Colorfulness-based Frame Pruning** (CFP). Both approaches are compliant with the schema depicted in Figure 1.

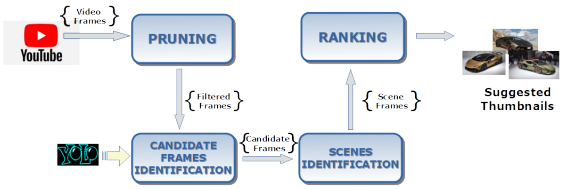


Figure 1: Schematic representation of baseline (for both BFP and CFP).

Both approaches are also summarized step-by-step in Algorithm 1. The input is the sequence (X_N) of the N video frames, and the domain D of the underlying video. The first step is to estimate the frame quality. The function **estimateQuality** returns its quality score, which is computed with two different methods, as described in the following sections. After that, the pruning step consists of computing first the gradient of the quality scores series Ψ_Q. The peaks in Ψ_Q correspond to local maximums of "quality"; in so doing, we discard the frames not associated to peaks. The remaining images (filtered frames set X_F, hereinafter) are subsequently analyzed by YOLO, in order to identify relevant objects. Let us remark that the function *YOLO* takes the image and also the video domain D as inputs. The framework returns which relevant objects, i.e., objects related to D, an image contains, together with their probability score. If an image does not contain relevant objects, it will be discarded. The remaining set (X_{YOLO}) contains the candidate frames from which selecting the final thumbnails. The subsequent step is the scene identification. The function **getSceneFrames** computes the correlation between the histograms associated to a given frame and the subsequent frame in X_{YOLO}; if the correlation is lower than a given threshold, two consecutive frames are considered belonging to different scenes. Each scene change splits X_{YOLO} in subsets, each one corresponding to a scene. For each scene, the frame having the best σ_Y is selected. The set of selected "scene" frames X_S is finally ranked by their scores σ_Y; the ranked frames set is represented by X_T. The channel creator then may select the best k ranked images as final thumbnails. Let us point out that in our experiments we selected a fixed number of suggested

[3]https://pjreddie.com/darknet/yolo/

thumbnails (3, in our case), rather than suggesting a variable number. This choice, already proposed in literature (Wang et al., 2018), is a trade-off between the need of proposing more than one image to capture the entire video content and the need of being proposed in a stable, fixed and compact UI.

Algorithm 1: Baseline approach.

Input: Video frames set (X_N), Domain (D)
Data:
Ψ_Q: quality measures
X_F: filtered frames (after pruning)
X_S: scene frames (one frame per scene)
X_{YOLO}: frames containing YOLO objects
Output: Selected thumbnails (X_T)
begin
  $\Psi_Q, X_{YOLO} = \{\}, \{\}$
  `/* Pruning */`
  for $x \in X_N$ **do**
    $\Psi_Q \longleftarrow \Psi_Q \cup \{\textbf{estimateQuality}(x)\}$
  $X_F \longleftarrow \textbf{applyPruning}(\Psi_Q, X_N)$
  `/* Candidate frames selection */`
  for $x \in X_F$ **do**
    $obj \longleftarrow YOLO(x, D)$
    if $obj \neq NULL$ **then**
      $X_{YOLO} \longleftarrow X_{YOLO} \cup \{x\}$
  `/* Scenes identification */`
  $X_S \longleftarrow \textbf{getSceneFrames}(X_{YOLO})$
  `/* Ranking */`
  $X_T \longleftarrow \textbf{rank}(X_S)$
  return $X_T.top(K)$

As already remarked, the difference between the two variants of the baselines is in the **estimateQuality** function. Let us describe in the following subsections how such s quality is estimated.

Blur-based Frame Pruning (BFP)

Although blur (or "smoothing") is often seen as a nuisance, as it is basically an image degradation that makes visual content less interpretable by humans, blurring an image removes "outlier" pixels that may be noisy in the image. Blur also combines information about both texture and motion of the objects in a single image. Hence, recovering texture and motion from blurred images is also used to understand dynamics of scenes. Moreover, a video sequence usually contains many frames "naturally" blurred (e.g., frames of scenes change). Therefore, this method is adopted with the twofold aim of noise reduction and normalization. The **estimateQuality** function of BFP approach assigns a score to each frame by blurring the image (applying a Laplacian filter), and computing its variance, which will be our quality score.

Colorfulness-based Frame Pruning (CFP)

Instead of computing the blur scores, with this baseline the **estimateQuality** function assigns a colorfulness score to each image. The score is computed as defined in the work of Hasler and Süsstrunk (2003). This preprocess is usually faster than blurring, with no loss of information.

3.3 The Proposed Approaches

As already pointed out, one of the main requirements of companies is the videos optimization, and in particular for thumbnail generation, is to rely on frameworks being equally not computationally expensive and time-efficient. The baseline systems described above rely on a preprocessing step for the pruning step, which requires a significant consumption of resources. On the one hand, we deem to devise an approach that does not perform pruning, not only because it would require more computational effort, but also because discarding too many frames may lead to miss meaningful and potentially relevant thumbnails. On the other hand, let us remark that pruning is aimed at reducing the amount of images being sent to the object detection framework. To this end, we propose two different algorithms able also to not "overload" the object detection framework. In the following subsections the details of both algorithms are described.

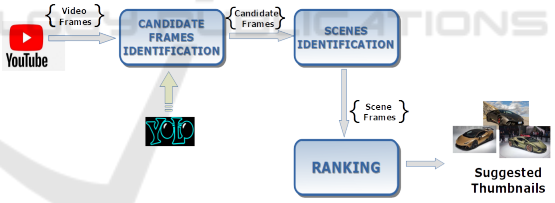


Figure 2: Schematic representation of the DOD approach.

Dynamic Object Detection-based (DOD)

The first proposal performs the same steps of BFP and CFP, except for pruning, as depicted in Figure 2.

The main difference, apart for the pruning, is how the candidate frames are selected. Instead of sending to YOLO a fixed set of frames (obtained by pruning, in the baseline), the frames are dynamically selected in the following way (summarized by Algorithm 2): (i) the first frame x_1 is selected as a candidate; (ii) each frame x_i in the sequence is compared with the previous frame x_{i-1} by measuring their correlation, computed as the difference between histograms of x_i and x_{i-1}; (iii) if the correlation is lower or equal than a given threshold, x_i is selected as a candidate, otherwise x_i and x_{i-1} are considered similar frames, and

Algorithm 2: DOD approach.

Input: Video frames set (X_N), Domain (D)
Data:
X_S: scene frames (one frame per scene)
X_{YOLO}: frames containing YOLO objects
th: correlation threshold
Output: Selected thumbnails (X_T)
begin
  $X_{YOLO} = \{\}$
  `/* Candidate frames selection */`
  $X_{YOLO} \longleftarrow X_{YOLO} \cup \{x_1\}$
  for $x_i \in (X_N - \{x_1\})$ **do**
    if ***corr***$(x_i, x_{i-1}) \leq th$ **then**
      $obj \longleftarrow YOLO(\mathrm{x}, D)$
      if $obj \neq NULL$ **then**
        $X_{YOLO} \longleftarrow X_{YOLO} \cup \{x_i\}$
  `/* Scenes identification */`
  $X_S \longleftarrow$ **getSceneFrames**(X_{YOLO})
  `/* Ranking */`
  $X_T \longleftarrow$ **rank**(X_S)
  return $X_T.top(K)$

x_i is discarded. Scene identification and ranking are performed in the same way of the baseline approaches.

Fast Scene Identification (FSI)

On the one hand, the DOD approach is expected to be faster than the baseline. On the other hand, the weakness of DOD is that scenes identification is based only on the presence of relevant objects in frames that differ on a simple histogram analysis. This aspect may lead to misleading scene splittings, or may propose noisy frames. To improve our first proposal, we devise the approach schematized in Figure 3 and described by Algorithm 3.

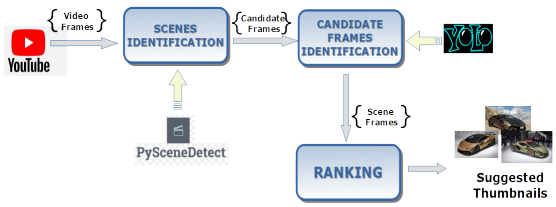


Figure 3: Schematic representation of the FSI approach.

With respect to the DOD approach, scene identification is performed *before* the object detection. We perform this step by relying on a well-known framework (PySceneDetect, see Section 4 for more details) which applies a blackframe filtering[4] and compares the brightness of each frame with a fixed threshold θ. Then it triggers a scene cut/break when this value crosses θ. For each identified scene, we simply consider the first frame as the scene frames, in this stage. Let us point out that methods of scene frames selection are currently under investigation. Subsequently, the set of scene frames X_S is analyzed by YOLO. Only frames containing relevant objects are selected as candidates. Ranking is performed like BFP and CFP approaches, sorting frames in descent order by their YOLO prediction scores.

[4]http://ffmpeg.org/ffmpeg-filters.html#blackframe

Algorithm 3: FSI approach.

Input: Video frames set (X_N, Domain (D))
Data:
X_S: scene frames (one frame per scene)
X_{YOLO}: frames containing YOLO objects
θ: brightness threshold
Output: Selected thumbnails (X_T)
begin
  $X_{YOLO} = \{\}, \{\}$
  `/* Scenes identification */`
  $X_S \longleftarrow$ **getSceneFrames**(X_N, θ)
  `/* Candidate frames selection */`
  for $x \in X_S$ **do**
    $obj \longleftarrow YOLO(\mathrm{x}, D)$
    if $obj \neq NULL$ **then**
      $X_{YOLO} \longleftarrow X_{YOLO} \cup \{x\}$
  `/* Ranking */`
  $X_T \longleftarrow$ **rank**(X_{YOLO})
  return $X_T.top(K)$

4 EXPERIMENTS

In this Section, after giving details about the adopted dataset and about the experimental settings, we describe which metric we used for evaluating our algorithms. Then, we report our preliminary results, together with a use case aimed at giving an example of thumbnail generation, compared with the chosen baseline.

4.1 Dataset

We manually selected a real-world dataset from YouTube portal. In particular, we selected videos belonging to 4 different domains (Motors, Tech, Food, and Animals). The selected dataset contains 31 videos. Let us point out that we report experiments on a small dataset only for a matter of time execution and evaluation. More experiments on a larger dataset are currently in progress. Details on the distribution

among categories and information about the video average length is reported in Table 1.

Table 1: Dataset details.

Domain	Number of videos	Average length (seconds)
Animals	6	136.5
Motors	8	96.5
Food	10	77.8
Tech	7	99.4
Total	*31*	*102.6*

4.2 Experiments Settings

YOLO Model. We used a pre-trained model of YOLO, trained with COCO (*C*ommon *O*bjects in *CO*ntext)[5] dataset, which contains more than 300k images of 91 objects types that "would be easily recognizable by a 4 year old" (Lin et al., 2014). To our purpose, we selected 4 subsets of the COCO objects to characterize our domains. Table 2 reports the details of each domain.

Table 2: Coco objects details.

Domain	COCO classes
Animals	bird, cat, dog, horse, sheep, cow, elephant, bear, zebra, giraffe
Motors	car, motorbike, truck, bus
Food	bottle, wine glass, cup, fork, knife, spoon, bowl, banana, apple, sandwich, orange, broccoli, carrot, hot dog, pizza, donut, cake
Tech	tvmonitor, laptop, mouse, remote, keyboard, cell phone, microwave

Implementation. All experiments have been performed by implementing several scripts written in Python language. In particular, for the FSI approach we chose to adopt an efficient framework for scene detection task, *PySceneDetect*[6], which is a Python module for detecting scene changes in videos, and automatically splitting the video into separate clips.

Parameters Settings. As we still are in a preliminary stage, we set the thresholds th (see Algorithms 1 and 2) and θ (in Algorithm 3) with typical values (further investigation is planned as future research activities). In particular, for BFP, CFP, and DOD we adopt the Pearson correlation (Pearson, 1895), and set the correlation threshold th with a value of 0.5. For FSI, we use the PysceneDetect default value of $\theta = 15$ for the threshold of brightness.

[5]http://cocodataset.org/

[6]https://pyscenedetect.readthedocs.io/en/latest/

4.3 Evaluation

In literature, although several attempts have been made, there are no standard criteria to evaluate the performance of a thumbnail generation algorithm. Therefore, to assess the performances of our algorithms, we performed extensive manual assessment by humans. An assessor assigned a relevance score to each generated thumbnail, each score belonging to a three-point relevance scale with the following meaning:

- **non-relevant (score 1):** the thumbnail is totally useless in providing a clear idea of video content (e.g., no relevant objects, dark images, noisy frames);
- **somewhat relevant (score 2):** the thumbnail is not strictly related to the video content, but the image would provide a related concept (for example, a knife in a video of a specific recipe), which may be attractive for user's interest;
- **relevant (score 3):** the thumbnail perfectly capture the video content.

4.4 Results

A quantitative comparison between the proposed models and the baseline(s) is shown in Table 3, in which the average relevance score is reported for each domain, and also for the entire dataset. We will discuss these results in Section 5.

Table 3: Average relevance.

Domain	BFP	CFP	DOD	FSI
Animals	2.110	1.867	2.443	2.200
Motors	1.333	1.917	2.583	2.540
Food	1.467	1.967	2.300	2.600
Tech	1.427	1.713	2.473	2.473
Average	*1.547*	*1.880*	*2.440*	*2.483*

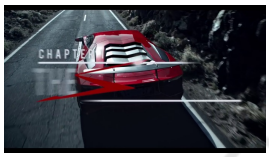
As we are also interested in comparing the computational effort of the approaches, in Table 4 we report the comparison of the execution time. Let us note that experiments have been performed in a scenario which does not rely on powerful hardware. In particular, no GPUs have been used and all the approaches have been compared with same hardware and settings.

Table 4: Average execution time (in seconds).

Domain	BFP	CFP	DOD	FSI
Animals	551.4	526.7	208.6	94.8
Motors	393.1	380.6	178.7	73.3
Food	261.6	285.3	118.7	61.1
Tech	359.2	353.3	161.4	64.0
Total	*373.6*	*372.0*	*161.2*	*72.7*

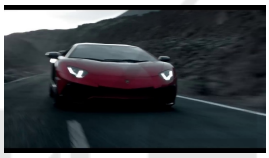
Use Case. For the sake of completeness, let us give an example of thumbnail selection for all the approaches. Due to paper limits, we report here only the first ranked thumbnail of a video, selected from our dataset, regarding the domain "Motors".

(a) BFP



(b) CFP

(c) DOD



(d) FSI

Figure 4: Example: best ranked thumbnail for each approach.

To let the reader understand the content, the title of the video is *Lamborghini Aventador LP 750-4 SV*. The generated thumbnails are depicted in Figure 4. Let us note that, for the BFP approach the suggested thumbnail does not contain a car (Figure 4a). This behavior confirms our expectation: the pruning phase, in our opinion, discarded many meaningful frames; the reported thumbnail is just a false positive (YOLO recognized the frame as one very likely to contain a car). Furthermore, also the CFP approach suggests a thumbnail of poor quality (there is a blurred text that overlaps the car), as reported in Figure 4b. Figure 4d reports the generated thumbnail by FSI approach, and clearly shows how the quality is good, and the car is perfectly recognized.

5 DISCUSSION

As clearly highlighted in Table 3, our approaches, at least in this preliminary stage, outperform the baselines models. Looking at the average evaluation, DOD and FSI have similar performances, but significantly higher than the baseline systems. Although there is the need of performing deeper experiments on a larger dataset, preliminary results, in our opinion, confirm expectations that the pruning process may discard meaningful frames, and encourage to deeper investigate our algorithms. To remark that, the use case is a clear indicator on how our proposals may improve the thumbnail selection (as Figure 4a is totally unrelated to the video topic).

Furthermore, the comparison of the execution times clearly points out that our proposed methods are faster than BFP and CFP. This is potentially useful for medium and small companies, which usually require reducing hardware and infrastructural costs. In particular, FSI is a very time-efficient choice, and it could be a starting point to refine the algorithm, by introducing more complex analysis, in particular for scene frame selection, with a tolerable increase of time execution.

6 CONCLUSIONS AND FUTURE WORK

In this paper, we proposed two novel video thumbnail generation approaches, named DOD – Dynamic Object Detection-based, and FSI – Fast Scene Identification. Both methods are aimed at providing relevant thumbnails with the minimum effort in terms of time execution and computational resources. We rely on an object recognition framework (YOLO) which identifies relevant objects with the goal of identifying the most topic-related frames of a video. The methods select the thumbnails from the resulting frames sets. We performed preliminary experiments aimed at assessing and validating our models, and we compared them with a baseline system, proposed in two variants (BFP – Blur-based Frame Pruning, and CFP – Colorfulness-based Frame Pruning) compliant with the state-of-the-art. The assessments confirm our expectations, i.e., DOD and FSI outperform BFP and CFP in terms of relevance of the suggested thumbnails and time execution. These preliminary results encourage future refinements of the algorithms. We are currently experimenting the proposed algorithms, together with baseline comparison, on a larger dataset of videos, in order to provide a more strong and robust assessments.

Further modifications of algorithms are currently under investigation. First, for DOD approach, we are studying a more efficient way of performing the "dynamic" scene identifications, with, at the same time, the development of further refinements of rank-

ing step. Regarding the FSI approach, which is still in its preliminary stage, many directions are looking forward. We expect to provide more performative methods in (i) identifying scenes with further algorithms, (ii) selecting scene frames, and (iii) ranking candidate frames.

ACKNOWLEDGEMENTS

This research has been partially supported by the "Bando Aiuti per progetti di Ricerca e Sviluppo"—POR FESR6832014-2020—Asse 1, Azione 1.1.3. Project VideoBrain- Intelligent Video Optmization.

REFERENCES

Ames, M. and Naaman, M. (2007). *Why We Tag: Motivations for Annotation in Mobile and Online Media*. CHI '07. Association for Computing Machinery, New York, NY, USA.

Gao, Y., Zhang, T., and Xiao, J. (2009). Thematic video thumbnail selection. In *Proc. of the 16th IEEE Int. Conf. on Image Processing*, ICIP'09, pages 4277–4280, Piscataway, NJ, USA. IEEE Press.

Hasler, D. and Süsstrunk, S. (2003). Measuring colourfulness in natural images. *Human Vision and Electronic Imaging*.

Kang, H.-W. and Hua, X.-S. (2005). To learn representativeness of video frames. In *Proceedings of the 13th Annual ACM International Conference on Multimedia*, MULTIMEDIA '05, page 423–426, New York, NY, USA. Association for Computing Machinery.

Lee, Y. J., Ghosh, J., and Grauman, K. (2012). Discovering important people and objects for egocentric video summarization. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1346–1353.

Li, H., Yi, L., Liu, B., and Wang, Y. (2014). Localizing relevant frames in web videos using topic model and relevance filtering. *Mach. Vis. Appl.*, pages 1661–1670.

Lin, T.-Y., Maire, M., Belongie, S., Bourdev, L., Girshick, R., Hays, J., Perona, P., Ramanan, D., Zitnick, C. L., and Dollár, P. (2014). Microsoft coco: Common objects in context.

Liu, C., Huang, Q., and Jiang, S. (2011). Query sensitive dynamic web video thumbnail generation. In *2011 18th IEEE International Conference on Image Processing*, pages 2449–2452.

Liu, W., Mei, T., Zhang, Y., Che, C., and Luo, J. (2015). Multi-task deep visual-semantic embedding for video thumbnail selection. In *CVPR*, pages 3707–3715. IEEE Computer Society.

Pearson, K. (1895). Note on regression and inheritance in the case of two parents. *Proceedings of the Royal Society of London*, 58:240–242.

Potapov, D., Douze, M., Harchaoui, Z., and Schmid, C. (2014). Category-specific video summarization. In Fleet, D., Pajdla, T., Schiele, B., and Tuytelaars, T., editors, *ECCV - European Conference on Computer Vision*, volume 8694 of *Lecture Notes in Computer Science*, pages 540–555, Zurich, Switzerland. Springer.

Rav-Acha, A., Pritch, Y., and Peleg, S. (2006). Making a long video short: Dynamic video synopsis. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 1, pages 435–441.

Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. (2015). You only look once: Unified, real-time object detection. cite arxiv:1506.02640.

Song, Y., Redi, M., Vallmitjana, J., and Jaimes, A. (2016). To click or not to click: Automatic selection of beautiful thumbnails from videos. In Mukhopadhyay, S., Zhai, C., Bertino, E., Crestani, F., Mostafa, J., Tang, J., Si, L., Zhou, X., Chang, Y., Li, Y., and Sondhi, P., editors, *CIKM*, pages 659–668. ACM.

Wang, Y., Han, B., Li, D., and Thambiratnam, K. (2018). Compact web video summarization via supervised learning. In *2018 IEEE International Conference on Multimedia Expo Workshops (ICMEW)*, pages 1–4.

Zhang, K., Chao, W., Sha, F., and Grauman, K. (2016). Summary transfer: Exemplar-based subset selection for video summarization. *CoRR*, abs/1603.03369.