A large-Scale TV Dataset for partial video copy detection

Van-Hao Le, Mathieu Delalandre, Donatello Conte

A large-Scale TV Dataset for partial video copy detection

Van-Hao LE, Mathieu Delalandre, and Donatello Conte

LIFAT Laboratory, Tours city, France
firstname.lastname@univ-tours.fr

Abstract. This paper is interested with the performance evaluation of the partial video copy detection. Several public datasets exist designed from web videos. The detection problem is inherent to the continuous video broadcasting. The alternative is then to process with TV datasets offering a deeper scalability and a control of degradations for a fine performance evaluation. We propose in this paper a TV dataset called STVD. It is designed with a protocol ensuring a scalable capture and robust groundtruthing. STVD is the largest public dataset on the task with a near 83k videos having a total duration of $10,660$ hours. Performance evaluation results of representative methods on the dataset are reported in the paper for a baseline comparison.

Keywords: partial video copy, detection, TV, dataset, evaluation

1 Introduction

This paper is interested with the Partial Video Copy Detection (PVCD). It aims to find one or more segments of a reference video which have transformed copies. It is a well-known topic in the computer vision field [12]. The recent works investigate the detection methods robust to the spatial & temporal deformations [6,5,15] or real-time [4,13,19]. A key aspect for any computer vision task is to design public datasets for performance evaluation. A few has been proposed in the literature for the PVCD [21,9,8]. They have been mainly designed from Web videos ensuring realistic degradations. However, this approach raises different problems such as **(i)** a huge user interaction **(ii)** errors in the groundtruth **(iii)** a low-level of scalability **(iv)** an unbalance distribution of test sets **(v)** a difficulty to challenge the methods on a particular detection problem.

The PVCD is inherent to the continuous video broadcasting. An alternative is to process with TV datasets offering meaningful data and having a low-level of noise. This ensures a deeper scalability, a robust groundtruthing with the help of TV metadata and a fine control of video degradations. We propose in this paper a large-Scale TV Dataset for the PVCD, called STVD. It is made public available for the needs of the research community[1]. Section 2 describes the related work. Section 3 presents our protocol and pipeline for the video capture

[1] http://mathieu.delalandre.free.fr/projects/stvd/index.html

and groundtruthing. Experiments to design the dataset are reported in section 4 with performance evaluation results of representative methods. Section 5 provides conclusions and perspectives. For convenience, Table 1 gives the meaning of the main symbols used in the paper.

Table 1: Main symbols and terms used in the paper

Symbols	**Meaning**
$\mathbf{t}, \hat{\mathbf{t}}$	the scheduled and detected timestamp for a TV program
$\mathbf{L} \in [\mathbf{L}_{\min}, \mathbf{L}_{\max}]$	$\mathbf{L} = \hat{\mathbf{t}} - \mathbf{t}$ is the latency, $\mathbf{L}_{\min} < \mathbf{0}$, $\mathbf{L}_{\max} > \mathbf{0}$ the min and max
$\mathbf{L}^- < 0, \mathbf{L}^+ > 0$	a negative and positive latency, respectively
$\mathbf{W} = \mathbf{W}^- + \mathbf{W}^+$	the capture window
$\mathbf{D} \in [\mathbf{D}_{\min}, \mathbf{D}_{\max}]$	$\mathbf{D} > 0$ is a program duration, $\mathbf{D}_{\min}$, $\mathbf{D}_{\max}$ the min and max
$\mathbf{T_0}, \ldots, \mathbf{T_6}$	the video degradations and transformations
$\mathcal{S}$	a $\mathbf{T_0}$ sequence starting at $\mathbf{s} = \mathbf{t} - \lvert\mathbf{L}^-\rvert$ and ending at $\mathbf{e} = \mathbf{t} + \mathbf{D} + \mathbf{L}^+$
α, β	the parameters to control the degradation level

2 Related work

Several datasets have been proposed in the literature for the performance evaluation of the PVCD. They are listed in Table 2. These datasets provide video files with a groundtruth. The groundtruth labels the partial video copies. The datasets can be used to characterize the tasks of video detection or retrieval. They are constituted by two main sets of (i) query and (ii) testing videos.

Table 2: Comparison of datasets for the PVCD performance evaluation

Datasets	**TV_2007**	**CC_WEB**	**TRECVID**	**TV_2014**	**VCDB**	**SVD**	**STVD**
Reference	[10]	[21]	[16]	[2]	[9]	[8]	Ours
Year	2007	2009	2010	2014	2016	2019	2021
Query videos	100	24	1,608	N/A	28	1,206	243
Positive videos	500	3,481	134	20,000,000	528	10,211	19,280
Negative videos	N/A	9,309	7,866	N/A	100,000	26,927	64,040
Duration (h)	60,000	537	200	380,000	2,030	197	10,660
Annotation cost (m-h)	N/A	N/A	N/A	N/A	700	800	105
Source of capture	TV	Web	Web	TV	Web	Web	TV
Degradation methods	synthetic	real	synthetic	synthetic	real	synthetic	synthetic
Public available	no	yes	no	no	yes	yes	yes

The (h), (m-h) and N/A stand for (in hours), (in man-hours) and (not available), respectively.

The testing set groups negative and positive videos. The negative videos are not appearing in the query set. The positive videos contain copies of the queries. Some datasets have a small size [21,16,8]. Another limitation is the unbalanced distribution of positive / negative videos [16,9]. This is explained by the groundtruthing approaches requiring a huge user interaction [9,8]. Several datasets are not public available due to the intellectual property [10,16,2].

The positive videos are queries with degradations. Depending the datasets, the degradations could result from a real noise [21,9] or produced with synthetic methods [10,16,2,8]. The real noise results from the video processing pipeline (i.e., capture, networking, editing). The datasets could be obtained from a TV [10,2] or a Web [21,16,9,8] capture. As a general trend, the TV capture guaranties a lowest level of noise. Synthetic methods could be applied next for degradation. This allows a fine control for performance evaluation.

The main public dataset in the literature is VCDB [9]. It presents several limitations such as **(i)** an average scalability challenge **(ii)** a weak balance for the positive / negative videos **(iii)** a huge level of noise at the capture making unable to drive the performance evaluation on a particular detection problem.

We propose in this paper a new dataset and protocol for the TV video capture and groundtruthing. This dataset is public available for the needs of the research community[1]. It is the biggest public dataset in the literature with a near 83k videos and having a total duration of $10,660$ hours Table 2. Our capture is obtained with a low-level of degradation for a fine performance evaluation. Our protocol and dataset are presented in next section 3.

3 STVD: a large-Scale TV Dataset

We present in this section our protocol to design our large-Scale TV Dataset (STVD). Fig. 1 details our pipeline where 3 main components are used. We drive first a TV video capture (**C1**) that extracts positive/negative video candidates. This component processes with TV metadata. This requires a user interaction to constitute the query set and a video detection for verification driven in the component (**C2**). A final component (**C3**) is used for degradation.

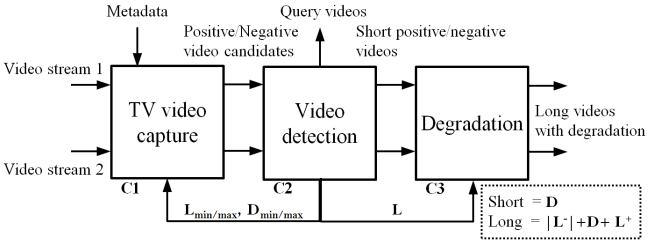

Fig. 1: Pipeline for constructing the STVD dataset

Our component (**C1**) is described into the publication [14]. It is mainly related to the hardware architecture and web crawling topics out of the scope of this paper. Section 3.1 reminds it for short. The components (**C2**) and (**C3**), for the video detection and degradation, constitute the new and key contributions of this paper. They are presented in details in sections 3.2 and 3.3, respectively.

3.1 TV video capture (C1)

Our component (**C1**) [14] captures the TV programs with a workstation [3,14,13]. This workstation can record daily video files on 8 TV channels simultaneously. We have captured 24 public channels during a period of 3 months. We have obtained a root database composed of $14,400$ hours of TV programs at a resolution 240×320 and having a total size of 3.46 TB. The resolution 240×320 constitutes a best tradeoff between the memory cost and video degradation.

We have processed next the TV metadata to capture positive/negative video candidates. These metadata are gathered by a Web crawler. This crawler targets only the daily and weekly programs having the maximum occurrence for the needs of the PVCD. A robust hashing method and user interaction are employed to guaranty a unique hash code for every TV program.

Every program in the metadata is delivered with a timestamp $\mathbf{t}$ to notify when it starts. However, no information is given about the exact location and duration of the repeated content. In addition, the TV broadcasting suffers from latency. To solve these problems, we have triggered the capture to get the jingles only appearing at the kickoff of programs Fig. 2. We have used a window having a size $\mathbf{W} = \mathbf{W}^- + \mathbf{W}^+$. The parameter $\mathbf{W}^-$ guaranties the minimum latency with the TV broadcasting $\mathbf{W}^- \geq |\mathbf{L}_{\min}|$. $\mathbf{W}^+$ is set with the maximum latency and jingle duration $\mathbf{W}^+ \geq \mathbf{D}_{\max} + \mathbf{L}_{\max}$. The capture is then done on the interval $[\mathbf{t} - \mathbf{W}^-, \mathbf{t} + \mathbf{W}^+]$. The $\mathbf{D}_{\max}$, $\mathbf{L}_{\min\,/\,\max}$ parameters are set with a loop-based methodology from the video detection (**C2**) as shown in Fig. 1.

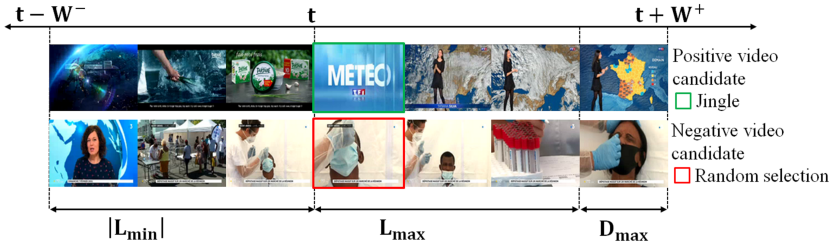

Fig. 2: TV video capture

Our component (**C1**) captures too the negative video candidates. These videos are not supposed to appear in the query and positive sets. For reliability, similar to the strategy deployed in [10] we have used two separate streams for the capture Fig. 1. For a better robustness, we have selected program contents apart of jingles. For every TV program, we have made idle for selection all the sequences where a jingle could appear in the range $[\mathbf{t} - \mathbf{W}^-, \mathbf{t} + \mathbf{W}^+]$. Any valid / not idle video sequence has been split into successive intervals having a duration $\mathbf{W}$. Within any interval, a selection is obtained at $\mathbf{t} = \mathbf{W}^-$ with a random duration $\mathbf{D} \in [\mathbf{D}_{\min}, \mathbf{D}_{\max}]$. For more details, please refer to [14].

3.2 Video detection (C2)

The component (**C1**) captures positive/negative video candidates. The negative video candidates are made consistent with the strategy deployed at the capture. For the positive video candidates, they have to be used to constitute the query set and validated. This is processed by our component (**C2**) Fig. 1.

A domain knowledge is required to detect the jingles. Indeed, repeated content not related to the programs could appear as the advertising. We have driven the detection with a user interaction and a GUI. Different error-prone cases could occur during the interaction: a jingle could be absent due to errors in the metadata, a jingle could present a different visual content Fig. 3 (a) the jingle could have a near-duplicate jingle appearing in a different channel Fig. 3 (b) or for a different program within a same channel Fig. 3 (c).

(a) (b) (c)

Fig. 3: Jingles (a) with a different visual content (b) (c) that are near-duplicate

This involves a large amount of visual inspection requiring an automatic video detection to support the interaction. A key constraint is the lack of a training database. We have considered the Zero-mean Normalized Cross-Correlation (**ZNCC**) for matching as a learning-free method [20]. The **ZNCC** fits well with the detection problem as it is robust to noise and contrast-invariant [4,13].

Our approach is illustrated in Fig. 4. For an accurate detection, we have matched the full frames ordered in the time domain. The **ZNCC** scores of frame matching are aggregated with weighted averaging to obtain a $\overline{\mathbf{ZNCC}}$. A subset of negative video candidates is used to fix the threshold for detection. The maximum score gives the timestamp for detection $\hat{\mathbf{t}}$. The difference with the scheduled timestamp is the latency $\mathbf{L} = \hat{\mathbf{t}} - \mathbf{t}$. The overall detection is supported with a GPU and time-efficient implementations suitable for the user interaction. As shown in section 4.1, we have obtained a separability with this approach. The $\mathbf{D}_{\min\,/\,\max}$, $\mathbf{L}_{\min\,/\,\max}$ parameters and the latency $\mathbf{L}$ are used for setting in the components (**C1**) and (**C3**) as shown in Fig. 1.

The user interaction could be time consuming. We have adopted a strategy for bounding. All the hashcodes of TV programs are marked first as unlabelled. The hashcode with the maximum number of occurrence is still selected for inspection. It is labelled when a jingle is detected and validated by the user. The detection cases of Fig. 3 serve to correct the hashcodes. The case (a) splits the hashcode whereas the cases (b) (c) merge two hashcodes. This strategy guaranties a low-level of interaction compared to the other approaches Table 2.

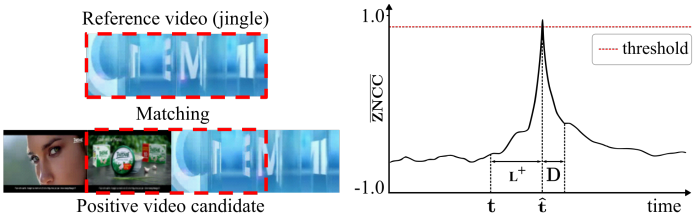

Fig. 4: Video detection

3.3 Video degradation (C3)

The positive/negative videos obtained with the components (**C1**), (**C2**) correspond to true-life captures with real noisy conditions. For the needs of performance evaluation, a common issue is to apply additional synthetic methods to degrade the videos [10,16,2,8]. By performing transformations, a fine performance evaluation can be handled and more challenging datasets can be designed in order to stress the methods for detection.

Similar to the works [10,16,2,8], we have selected a set of representative methods detailed in Table 3 labelled $\mathbf{T_0}$ to $\mathbf{T_6}$. These are applied both to the positive and negative videos. For the performance evaluation of PVCD methods, we use first a transformation $\mathbf{T_0}$ to get long videos embedding the positive videos. Then, the methods enter in two categories Fig. 5 for pixel attack $\mathbf{T_{1-2}}$ (b) and global transformations $\mathbf{T_{3-5}}$ (c). A final transformation $\mathbf{T_6}$ is used for video speeding.

Table 3: Degradation methods for video transformation

Label	Method	Parameters
$\mathbf{T_0}$	video cut	uses the latency distribution to cut segments before / after the video and having a duration $\|\mathbf{L}^-\|, \mathbf{L}^+$, respectively
$\mathbf{T_1}$	down-scaling	applies a random down-scaling $\alpha \in [0.1, 0.9]$ to get frames from 24×32 up to 216×306 for a robust matching with time optimization [18]
$\mathbf{T_2}$	compression	processes with a parameter $\frac{1}{\beta}$ with $\beta \in [1, 80]$ applied to the recommended kbps $\in \{140, 280, 420\}$ for capture [1] such as $\frac{1}{\beta} \times$ kbps
$\mathbf{T_3}$	flipping	applies randomly (yes/no) a flipping transformation to the video
$\mathbf{T_4}$	rotating	applies a random vertical/horizontal rotation $\in \{0, \frac{\Pi}{2}, \Pi, \frac{3}{2}\Pi\}$
$\mathbf{T_5}$	black border & stretching	selects an aspect ratio $\frac{w}{h} \in \{0.46, 0.56, 0.63, 0.75, 1.33, 1.6, 1.78, 2.17\}$ to introduce left / right borders ($\frac{w}{h} < 1$) or to stretch the image ($\frac{w}{h} > 1$)
$\mathbf{T_6}$	video speeding	speeds down the videos at a FPS $\in [15, 25]$

For the needs of the PVCD, short negative/positive videos must be embedded into longest sequences $\mathcal{S}$. We use a specific transformation $\mathbf{T_0}$ in our approach designed with our latency measure $\mathbf{L}$ Fig. 4. $\mathbf{T_0}$ extracts additional left/rigth video segments within the window of size $\mathbf{W}$ Fig. 6 (a). The duration of $\mathcal{S}$ must be fixed, we have set $\mathbf{T_0}$ with the latency distribution obtained with the component (**C2**) as illustrated in Fig 1. Considering a short negative video timestamped at $\mathbf{t} = \mathbf{W}^-$ in (**C1**), or a query / jingle detected at $\hat{\mathbf{t}}$ in (**C2**), $\mathcal{S}_\mathbf{i}$ is obtained

Fig. 5: Degradation (a) reference (b) pixel attack (c) global transformations

by cutting a long video at $\mathbf{s_i} = \mathbf{t_i} - |\mathbf{L}^-|$ and $\mathbf{e_i} = \mathbf{t_i} + \mathbf{D_i} + \mathbf{L}^+$ (and with $\hat{\mathbf{t}}_\mathbf{i}$ respectively) with $\mathbf{L}^-, \mathbf{L}^+$ random negative/positive latency values.

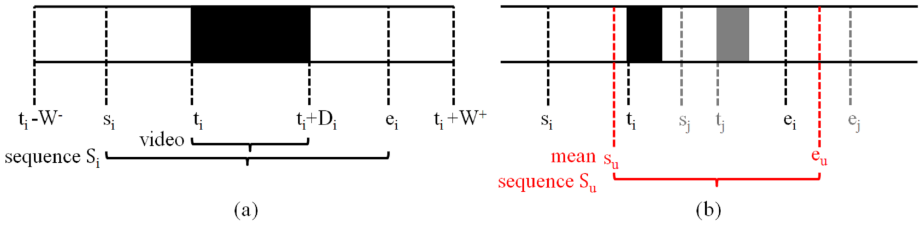

Fig. 6: (a) a sequence $\mathcal{S}$ (b) covering case

A sequence $\mathcal{S}$ could be extracted for any short negative video. Indeed, a selection at $\mathbf{t} = \mathbf{W}^-$ within a window of size $\mathbf{W}$ by (**C1**) cannot result in a covering case while using the latency in $\mathbf{T_0}$. However, such a case could appear with the short positive videos Fig. 6 (b). Considering two videos timestamped at $\mathbf{t_i}, \mathbf{t_j}$ with $\mathbf{t_i} < \mathbf{t_j}$, we could met a case where $\mathbf{t_j} - \mathbf{t_i} \ll \mathbf{W}^+$. A mean sequence $\mathcal{S}_\mathbf{u}$ must be computed and preserved if $\forall \mathcal{S}_\mathbf{i} \in \mathcal{S}_\mathbf{u}$, $\mathbf{s_u} < \mathbf{t_i} + \mathbf{D_i} < \mathbf{e_u}$. That is, a long positive video for testing could embed several queries.

We apply next a set of baseline video processing $\mathbf{T_{1-6}}$ for degradation. $\mathbf{T_1}, \mathbf{T_2}$ are set with recommended parameters for robust low-resolution video processing [18] and capture [1]. Two parameters α, β control the level of degradation. $\mathbf{T_3}$ and $\mathbf{T_4}$ apply realistic geometric transformations for video rendering as the flipping and the horizontal/vertical rotations. The aspect ratio parameters in $\mathbf{T_5}$ have been fixed using the standard screen resolutions[2]. $\mathbf{T_6}$ speeds down the videos with predefined FPS similar to [21,8].

We have combined the degradations $\mathbf{T_0}$ to $\mathbf{T_6}$ to generate the test sets A to F as detailed in Table 4. The test set A gives a root capture while applying only $\mathbf{T_0}$. It is given for the needs of tuning a performance evaluation task. The test sets B and C apply a pixel attack with $\mathbf{T_1}$ and $\mathbf{T_2}$ at two levels of degradation with the control of parameters α, β. The test set B has a low-level of distortion and scalability and constitutes a "hello world" benchmark. The test set C presents

[2] For desktop, tablet and phone https://gs.statcounter.com/

a hard pixel attack. The test set D is related to the global transformations with $\mathbf{T_3}$ to $\mathbf{T_5}$ whereas the test set E applies $\mathbf{T_6}$ for video speeding. For storage optimization, $\mathbf{T_1}$ and $\mathbf{T_2}$ are used with predefined parameters α, β ensuring a negligible degradation. At last, the test set F combines the sets C, D and E.

Table 4: Test sets

Test set	$\mathbf{T_0}$	$\mathbf{T_{1-2}}$	$\alpha \in$	$\beta \in$	$\mathbf{T_{3-5}}$	$\mathbf{T_6}$	**Description**
Set A	√						Root capture for tuning
Set B	√	√	$[0.25, 0.9[$	$[1, 40[$			"Hello world" test set
Set C	√	√	$[0.1, 0.25]$	$[40, 80]$			Pixel attack with scalability
Set D	√	√	0.6	20	√		Global transformations with scalability
Set E	√	√	0.6	20		√	Video speeding with scalability
Set F	√	√	$[0.1, 0.25]$	$[40, 80]$	√	√	Combination of sets C, D and E

4 Experiments

4.1 Dataset and groundtruthing

We report in this section experiments to generate the dataset with the groundtruth. Table 5 details the dataset organization where the components (**C1**), (**C2**) and (**C3**) have been used to generate the positive/negative videos with degradations.

Table 5: STVD dataset

	Root capture		**C1, C2**		**C3**		
	Channels	**Duration**	**Videos**	**Duration**	**Test sets**	**Videos**	**Duration**
Positive videos	8	4,800 h	3,780	6 h	6	19,280	2,515 h
Negative videos	16	9,600 h	12,165	21 h	6	64,040	8,145 h

We have split the root database obtained with (**C1**) into two subsets of $4,800$ and $9,600$ hours for the positive/negative videos. We have captured then a near 3k and 12k positive/negative video candidates with the metadata[3]. The positive video candidates have been processed with the component (**C2**). We have extracted 243 distinct jingles with the GUI. They have been matched against the positive video candidates and a subset of negative videos as detailed in section 3.2. We have observed a separability between the interclass / intraclass $\overline{\mathbf{ZNCC}}$ distributions $\in [0.79, 0.90]$ Fig. 7 (a). This ensures none false positive case.

For a further investigation, Fig. 7 (b) gives a characterization of the interclass $\overline{\mathbf{ZNCC}}$ distribution in terms of compression noise and contrast deviation. We have employed the standard metrics MSE[4] [7] and CNR[4] [11], respectively. The distributions are compact with a MSE < 20 and CNR < 0.01 for most of the

[3] A full analysis and experiments with the metadata are reported into [14].
[4] Mean Square Error, Contrast Noise Ratio

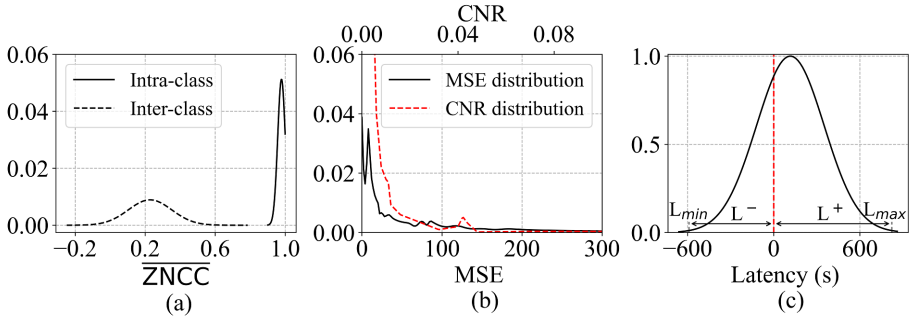

Fig. 7: Distributions of (a) $\overline{\mathbf{ZNCC}}$ (b) MSE / CNR (c) latency

matching cases. This reflects the $\overline{\mathbf{ZNCC}}$ distributions obtained in Fig. 7 (a) and the sources of degradation for the TV capture.

We have analysed then the two time aspects illustrated in Fig. 4. We have obtained a total video duration of 6 hours Table 5 as the jingles have a short duration $\mathbf{D} \in [1, 25]$ seconds. For the latency, we have observed an almost-Gaussian distribution with $\mathbf{L} \in [-590, 820]$ seconds at $\pm\mathbf{3}\sigma$ Fig. 7 (c). We have used the duration and latency distributions to set the $\mathbf{D}_{\min\,/\,\max}$, $\mathbf{L}_{\min\,/\,\max}$ parameters and the random model in $\mathbf{T_0}$ for (**C1**) and (**C3**) as shown in Fig. 1.

We have then applied the component (**C3**) to get the 6 test sets as discussed in section 3.3. For the test sets A, C, D, E and F we have obtained $5 \times 3,780 = 18,900$ and $5 \times 12,165 = 60,825$ positive/negative videos, respectively. The test set B has been generated in balance for a low scalability with a total number of $2 \times 3,780 = 7,560$ videos. We have observed a $\simeq 15\%$ of covering cases with the positive videos Fig. 6 (b). We have then obtained a total amount of $\simeq$ 83k videos composed of $19,280$ and $64,040$ positive/negative videos Table. 5.

Considering the latency distribution Fig. 7 (c), the application of $\mathbf{T_0}$ has resulted in an average duration $|\mathbf{L}^-| + \mathbf{L}^+$ of 7.5 minutes. The total duration of the dataset is $10,660$ hours with $2,515$ hours and $8,145$ hours for the positive/negative videos, respectively Table. 5. Each test set C to F contains $\simeq 1,960$ hours of testing video for scalability competitive with the VCDB dataset [9] Table 2. Considering the all test sets A to F, STVD is the largest dataset of the literature $\times 5$ larger than VCDB. STVD is made public available[1] for the needs of the research community.

4.2 Performance Evaluation

We present in this section performance evaluation results on the STVD dataset of representative PVCD methods [24,23,22]. These methods process in two steps for key-frame extraction and matching. The key-frame extraction selects candidate frames for matching based on sampling methods [23,22] or temporal features [24]. The matching processes with features (SIFT [24], BRIEF [23] and CNN [22]) and optimization components for the time processing requirement.

We have applied a protocol for a fair comparison. We have normalized the key-frame extraction step within all the methods. The SIFT and BRIEF features are not supporting the global transformations. We have bounded the evaluation to the test sets B and C only. We have characterized the methods in a learning-free / pre-trained mode. Only the SIFT and BRIEF features of query frames have been stored for comparison. The CNN features have been obtained from a pre-trained VGG16 network from the ILSVRC dataset [17]. We have also removed the optimization components for a strongest accuracy. The F_1 score has been used as it is common to characterize the PVCD methods [9,4,19,6,5,15].

We have evaluated first the method [23] on the test set B. We have obtained a score $F_1 = 0.98$ highlighting the "hello world" ability. Further experiments have been investigated on the test set C Fig. 8. We have constituted first a subset with in balance 3k +3k positive/negative videos. Fig. 8 (a) gives the F_1 scores against the normalized thresholds for all the methods. We have obtained optimum scores $F_1 \in [0.73, 0.83]$ with a top $F_1 = 0.83$ for the method [23]. A gap $\simeq 0.15$ appears for [23] between the test sets B and C due to the pixel attack. Fig. 8 (b) reports the results of the top methods [23,22] on the full test set C while increasing the negative videos up to 12k. We have observed a gap $\simeq 0.25$ for the F_1 score due to the scalability with a better robustness for the CNN features [22].

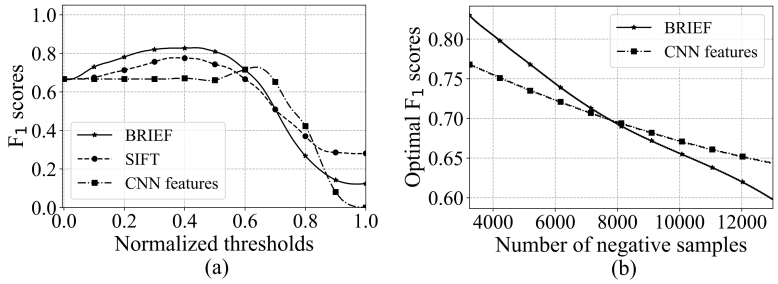

Fig. 8: F_1 scores on the test set C
(a) comparison of methods [24,23,22] (b) performance with scalability for [23,22]

5 Conclusions and perspectives

We propose in this paper a new dataset to evaluate the PVCD methods called STVD. This dataset is designed with a protocol ensuring a scalable capture and robust groundtruthing. STVD is today the largest public dataset on the task. It covers a near 83k videos for a total duration of $10,660$ hours. Performance evaluation results of representative methods on the dataset are reported in the paper for a baseline comparison. A key issue next will be to promote the dataset in the research community. Additional test sets should be included to address specific PVCD tasks such as the real-time or near-duplicate detection.

References

1. AVerMedia: Avermedia capture card software developement kit. Tech. Rep. 4.2, AVerMedia Technologies, Inc. www.avermedia.com (2015)
2. Chenot, J., Daigneault, G.: A large-scale audio and video fingerprints-generated database of tv repeated contents. In: Workshop on Content-Based Multimedia Indexing (CBMI). pp. 1–6 (2014)
3. Delalandre, M.: A workstation for real-time processing of multi-channel tv. In: International Workshop on AI for Smart TV Content Production (AI4TV). pp. 53–54 (2019)
4. Guzman-Zavaleta, Z., Uribe, C.: Partial-copy detection of non-simulated videos using learning at decision level. Multimedia Tools and Applications **78**(2), 2427–2446 (2019)
5. Han, Z., He, X., Tang, M., Lv, Y.: Video similarity and alignment learning on partial video copy detection. In: ACM International Conference on Multimedia (MM). pp. 4165–4173 (2021)
6. Hu, Y., Mu, Z., Ai, X.: Strnn: End-to-end deep learning framework for video partial copy detection. Journal of Physics: Conference Series **1237**(2), 022112 (2019)
7. Ieremeiev, O., Lukin, V., Okarma, K., Egiazarian, K.: Full-reference quality metric based on neural network to assess the visual quality of remote sensing images. Remote Sensing **12**(15), 2349 (2020)
8. Jiang, Q., al: Svd: A large-scale short video dataset for near-duplicate video retrieval. In: International Conference on Computer Vision (ICCV). pp. 5280–5288 (2019)
9. Jiang, Y., Wang, J.: Partial copy detection in videos: A benchmark and an evaluation of popular methods. IEEE Transactions on Big Data **2**(1), 32–42 (2016)
10. Joly, A., Buisson, O., Frélicot, C.: Content-based copy retrieval using distortion-based probabilistic similarity search. IEEE Transactions on Multimedia **9**(2), 293–306 (2007)
11. Kanmani, M., Narsimhan, V.: An image contrast enhancement algorithm for grayscale images using particle swarm optimization. Multimed Tools Appl **77**(3), 23371–23387 (2018)
12. Law-To, J., al: Video copy detection: a comparative study. In: Conference on Image and Video Retrieval (CIVR). pp. 371–378 (2007)
13. Le, V., Delalandre, M., Conte, D.: Real-time detection of partial video copy on tv workstation. In: International Conference on Content-Based Multimedia Indexing (CBMI). pp. 1–6 (2021)
14. Le, V., Delalandre, M., Conte, D.: Une large base de données pour la détection de segments de vidéos tv. In: Journées Francophones des Jeunes Chercheurs en Vision par Ordinateur (ORASIS) (2021)
15. Liu, X., Feng, X., Pan, P.: Gann: A graph alignment neural network for video partial copy detection. In: Conference on Big Data Security on Cloud (BigDataSecurity), Conference on High Performance and Smart Computing (HPSC), Conference on Intelligent Data and Security (IDS). pp. 191–196 (2021)
16. Over, P., al: Trecvid 2010 - an overview of the goals, tasks, data, evaluation mechanisms and metrics. NIST, https://www.nist.gov/ (2010)
17. Russakovsky, O., al: Imagenet large scale visual recognition challenge. International Journal of Computer Vision (IJCV) **115**(3), 211–252 (2015)
18. Su, J., Vargas, D., Sakurai, K.: One pixel attack for fooling deep neural networks. Transactions on Evolutionary Computation (TEVC) **23**(5), 828–841 (2019)

19. Tan, W., Guo, H., Liu, R.: A fast partial video copy detection using knn and global feature database. In: Preprint arXiv. No. 2105.01713 (2021)
20. Wang, X., Wang, X., Han, L.: A novel parallel architecture for template matching based on zero-mean normalized cross-correlation. IEEE Access **7**, 186626–186636 (2019)
21. Wu, X., Ngo, C., Hauptmann, A., Tan, H.: Real-time near-duplicate elimination for web video search with content and context. IEEE Transactions on Multimedia **11**(2), 196–207 (2009)
22. Zhang, C., al: Large-scale video retrieval via deep local convolutional features. Advances in Multimedia (7862894), 1687–5680 (2020)
23. Zhang, Y., Zhang, X.: Effective real-scenario video copy detection. In: International Conference on Pattern Recognition (ICPR). pp. 3951–3956 (2016)
24. Zhu, Y., Huang, X., Huang, Q., Tian, Q.: Large-scale video copy retrieval with temporal-concentration sift. Neurocomputing **187**, 83–91 (2016)