https://doi.org/10.2352/EI.2025.37.10.IPAS-233

HEVC stream fingerprinting for organic video: proof of concepts and illustrations

Mohamed Allouche, Mihai Mitrea, Carl De Sousa Trias; SAMOVAR, Telecom SudParis; Institut Polytechnique de Paris, Palaiseau, France

Abstract

While conventional video fingerprinting methods act in the uncompressed domain (pixels and/or directly derived representations from pixels), the present paper establishes the proof of concepts for compressed domain video fingerprinting. Thus, visual content is processed at the level of compressed stream syntax elements (luma/chroma coefficients, and intra prediction modes) by a homemade NN-based solution backboned by conventional CNN models (ResNet and MobileNet). The experimental validations are obtained out of processing a state of the art and a homemade HEVC compressed video databases, and bring forth Accuracy, Precision and Recall values larger than 0.9.

1. Introduction

Video content is ubiquitous, with more than 42,000 petabytes of data being exchanged worldwide monthly [1], and with continuous renewed applications or emerging usages benefiting from it.

Specifically, under the marketing framework, while the video advertising is still dominated by the paid content (content created by an advertiser who pays an announcer for the distribution), *organic video* is slowly but steadily advancing. Organic video content generally refers either to some user-generated content with implicit advertising value, or to some advertising content, spontaneously and freely distributed by a user on social networks [2]. In practice, such a content is subsequently shared by other users, on the same or on different social networks, thus creating a virtual distribution chain that is studied by marketing experts.

When transposed to the video processing framework, organic video distribution can be modelled by near duplicated[1] operations applied to a reference video content at each of its redistribution stages. Thus, the tracking of organic video can be ensured by video fingerprinting (also referred to as content-based copy detection, or near duplicate detection) that regroups research efforts devoted to retrieving duplicated and/or replicated versions of a given video sequence (query) in a reference video dataset [3, 4, 5].

Video fingerprints are compact digital representations extracted from the video itself. They are meant to uniquely identify video sequences even when their contents undergo a set of transformations. Two main properties apply to fingerprinting methods. Firstly, the unicity (or uniqueness) property assumes that semantically different contents result in different fingerprints (in the sense of a preestablished similarity measure and of its related threshold). Secondly, the robustness property relates to the possibility of identifying as similar sequences that are near-duplicated. These two main properties are generally evaluated under a binary decision framework (that is, the query is retrieved or not from a database), by computing the probabilities of false alarm (P_{fa}) and missed detection (P_{md}), or some derived entities like *Accuracy*, *Precision* and *Recall*.

Video fingerprinting is considered as a classification task where the same video and its altered version belongs to one and only one class. Video fingerprinting applicative field is related, yet complementary to video indexing that searches for different video content sharing some semantic similarities [6]: thus, video indexing is not supposed to feature the unicity property that is inner to video fingerprinting. With a two-decade history, video fingerprinting is now a mature research field, with approaches targeting a joint functional optimization of the fingerprinting extraction and retrieving procedures. To this end, various methodological frameworks, from information theory to machine learning (including deep learning) are explored [4, 5]. Despite their conceptual and applicative varieties, they most operate at the pixel level, after the stream decoding. This state-of-the art situation is different for video indexing, where previous attempts exploiting the compressed-domain information can be encountered [7, 8, 9].

The present study investigates the possibility of achieving compressed domain video fingerprinting, without exploiting any pixel-level information. To this end, we present an experimental study structured on two directions: (1) the selection of the compressed stream syntax elements representing the fingerprint, and (2) the design of the neural network structure achieving the fingerprinting retrieval task. The experiments correspond to HEVC video streams and are obtained out of processing two databases, namely UVG [10] and VID, a real-life organic video database organized in our study. The quantitative results show that Accuracy, Precision, and Recall values larger than 0.9 can be obtained, thus establishing the proof of concepts for compressed domain video fingerprinting. The final discussions open the door towards a deeper understanding of how neural networks (NN) work in compressed domain.

2. State-of-the-art

2.1 Uncompressed domain video fingerprinting

The uncompressed video fingerprinting methods process the pixel-based representation of the video sequences. They emerged some 20 years ago [4, 5], and cover the large majority state-of-the-art approaches. From the methodological point of view, they can be structured into conventional methods and deep learning-based methods.

[1] Near-duplicated content is here understood as "*identical or approximately identical videos close to the exact duplicate of each other, but different in file formats, encoding parameters, photometric variations (color, lighting changes), editing operations (caption, logo and border insertion), different lengths, and certain modifications (frames add/remove)*." [11]

The conventional methods are composed of fingerprint extraction and fingerprint retrieving. On the one hand, the fingerprint extraction encompasses video pre-processing (frame resizing, frame dropping, key-frame detection, color modifications), local feature extraction, global feature extraction, local/global feature description, temporal information retrieval, and the means for accelerating the search in the dataset. Just for illustration, local feature extraction can be based on HOG (Histogram of Oriented Gradients), ORB (Oriented FAST and rotated BRIEF), SIFT (Scale Invariant Feature Transform), SURF (Speeded-Up Robust Features), while global information may relate to BoVW (Bag of Visual Words) or color histograms. Examples of temporal information include histogram correlation, optical flow or TIRI (Temporal Informative Representative Image). On the other hand, the detection procedure starts by ensuring some time-alignment operations (time origin synchronization, jitter cancelation), followed by information matching according to some similarity measures (Hamming, Euclidian norms, correlation coefficients, maximum *a posteriori* probability, etc.).

The deep learning-based methods leverage on the neural networks for implicitly learning the visual salient features of the content and for subsequently classify the queries in the corresponding classes. To this end, a large variety of models are considered, individually or combined. Just for illustration, spatial information can be addressed by AlexNet, VGG, ResNet, while temporal information by LSTM derived structure (Siamese LSTM, BiLSTM).

2.2 Towards compressed domain video fingerprinting

As video content is currently recorded, stored, and transmitted in compressed formats, performing video fingerprinting at the compressed stream level is expected to avoid a tremendous number of decoding operations.

To the best of our knowledge, the earliest study in this respect is a conventional method that computes the fingerprinting based on information computed both from the decompressed (pixel) domain and from MPEG-2 stream is presented in [12]. Specifically, the fingerprint is a combination of the frame color histograms, ORB descriptors and motion vector normalized histogram. The retrieving procedure is individually performed at the level of the three components (based on their individual appropriate matching criteria) and the overall decision is achieved through fusing decisions made on multiple features by a weighted additive voting model.

In [13], a method exploiting a vulnerability in the MPEG-DASH standard which causes distinct packet bursts related to content, even with encrypted streams is presented. This method represents the bursts of information delivered to the end user as the fingerprint. A specific CNN model is then used for the matching step. Inspired by these results, the study in [14] further investigates the capabilities of the data collected by a *Middleman* present in the network, by extracting the information from the Wi-Fi traffic.

Other examples are mentioned in [4, 5].

Note that the interest in identifying video streams currently exceeds the strict field of video fingerprinting, as for instance identifying the streams delivered on specific video platforms [15].

2.3 Summary

While not meant to be either exhaustive or detailed, the state-of-the-art study bring to light several guiding directions for our work. Firstly, although compressed domain fingerprinting is promising, it remains unexplored. Secondly, NN based methods are intensively used, with fingerprinting being by default modelled as a classification problem.

Yet, doubts about the practical relevance of such general conclusions can arise. On the one hand, the association of deep learning and compressed domain processing seems a conceptual contradiction: while the deep learning paradigm exploits data redundancy in order to achieve its applicative task, video encoders are designed to get rid of redundancy in order to reduce the size of the sequence. On the other hand, fingerprinting is a specific case of classification problem: while conventional classifiers are trained to achieve ultimate generalization, fingerprinting methods are expected to provide fine grain generalization, allowing for the near-duplicated copies to be correctly retrieved, and discriminating the semantically related, yet distinct visual content.

3. Method presentation

When specifying the compressed domain video fingerprinting method, two main difficulties are encountered: the selection of compressed stream syntax elements representing the fingerprint, and the specification of the NN based model *a priori* likely to achieve the retrieval task.

3.1 HEVC fingerprint specification

HEVC (a.k.a. H.265) standard [16] is a conventional video encoding scheme, consisting of five basic operations: (1) Partitioning of frames in GoP (Group of Pictures, composed of an I frame and possibly of a variable number of P and B frames) and of the pixels of frames in blocks; (2) Predicting the similarities among the blocks in a frame and among successive frames in a GoP; (3) Transforming the prediction errors; (4) Quantizing the coefficients thus obtained, and (5) Entropy Coding the result. The decoding process (Fig. 1) reverses the order of the operations.

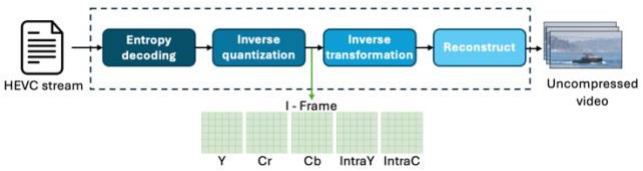

Figure 1. *HEVC decoding process and fingerprint extraction*

The fingerprint is defined by answering three questions, as follows.

Question 1: Where should the fingerprinting be extracted from? This question relates to the trade-off between the decoding operation complexity and the level of redundancy still existing in the compressed stream. From this point of view, we decide to extract the fingerprint after the Inverse quantization and before the Inverse transformation (the latter being the most complex decoding operation). This choice is also supported by previous results in compressed-domain video watermarking [17].

Question 2: What type of information from the GoP? To answer this question, the specificities of the organic video fingerprinting is considered. First, the usage of inter frame information is avoided, as it would not be able to distinguish between semantical related yet distinct video contents. Secondly, as various reencoding operations are likely to be encountered at each new posting, we shall extract the fingerprint from the I frames.

Question 3: What type of information from an I frame? As no a priori answer can be provided to this question, luma/chroma

coefficients, alongside with their intra prediction modes are considered. As no information related to the fingerprinting size is available either, three alternative configurations are considered, namely 32x32, 64x64, and 128x128. The selection of those elements is made according to the order of absolute values of the luma coefficients [16].

3.2 Fingerprinting NN model

In our study, we consider a structure composed a backbone and three additional layers, as illustrated in Fig. 2.

Candidate models for the backbone components are the widest used convolutional NN (CNN), namely the ResNet family [19] (ResNet18, ResNet50, ResNet101) and MobileNetV3small [20]. As usual in classification applications, a dense classification layer is positioned at the output (denoted by O3 in Fig. 2), to map the previously obtained features to the final output classes. This backbone should be completed by additional layers meant to solve the two previous identified issues.

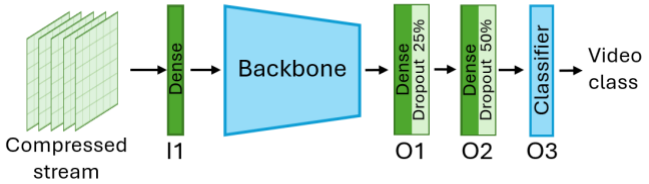

Figure 2. *Fingerprinting NN model: 1 input and 2 output layers (in green) are considered around the backbone and the final Classifier layer (in blue).*

On the one hand, prior to the backbone, an input preprocessing layer, denoted by I1, is added. I1 is dense and is expected to serve for the weighting of the heterogeneous information included in the fingerprint (Y, Cr, Cb, and their corresponding intra-prediction modes) to be learned. Note that in uncompressed domain fingerprinting, such a layer is not required, as the backbone is already designed for weighting the R, G, and B components of a pixel.

On the other hand, two postprocessing layers (denoted by O1 and O2 in Fig. 2) are included in the model. O1 and O2 are both dense, with dropout rates of 25% and 50%, respectively. These two layers are added for two complementary reasons. First, as the backbone is fed with a combination of compressed stream syntax elements, its task is more complex to be learned than the conventional (pixel) classification task for which it was designed. Secondly, the generalization trade-off between correctly retrieving near-duplicated contents and avoiding semantically related different contents becomes more complex, thus imposing two different dropout rates.

4. Experimental Results

4.1 Database specification

Uncompressed domain video fingerprinting solutions benefit from already available databases. Specifically, conventional methods can be benchmarked on challenge databases (e.g. TRECVID) while deep-learning solutions on general purposes computer vision datasets, e.g. YLI-MED [21] or Youtube-8M [22]. However, such databases cannot be considered in our study, as they are either already encoded with legacy encoders (like MPEG-2 or MPEG-4 AVC) or presented as tensors and do not allow to obtain back a video. Consequently, our experimental study starts by organizing the reference content to be processed, and to this end, we shall consider two complementary databases.

The UVG database [10] is available in raw format. In the present study, it is encoded by the VideoLAN implementation of HEVC [23]. UVG is composed of 16 natural 3840x2160 video sequences, with variable lengths ranging from 2.5 to 12 sec. The encoding process provides an average of 3.2 I frame per video (min. 2, max. 12). A total of 4983 frames is generated from this dataset. Note that UVG is designed as a generic end-to-end video encoding benchmarking database, with no direct relevance for the organic video applications. The organic video content is represented in our study by home-made database, referred to as VID and composed of 164 video excerpts, 1920x1080, from advertising content provided by an industrial company. This content was edited so as to not include duplicated/reused content in different sequences. VID sequences durations range from 5 to 180 sec. and they contain an average of 6.5 I frame per video (min. 1, max. 41). A total of 117769 frames are included in this dataset.

These two reference databases are subsequently subjected to a set of 10 different near-duplicated transformations, as follows. Firstly, 5 luminance/colorimetry modifications are applied: brightness, contrast, Gamma, hue, and saturation modifications. For each individual type of modification, 10 different relative increasing/decreasing parameters of maximum 33% are considered. Secondly, 3 types of video editing operations are performed, namely insert logo (image size equals to 200x500 pixels, randomly placed in the frame), insert subtitle, and central zoom (by 10 values between 10% and 30%). Finally, two video encoding modifications are considered, namely CRF (Constant Rate Factor) and QP (Quantizing Parameter) changes. The former was applied 10 times, with parameters ranging between 20 and 40, while the latter by 10 values ranging from 8 to 35.

4.2 Experimental setup

The experiments are performed on in-premises servers, with Xeon E5-1650 v3 @ 3.50GHz, 4 threads CPU, 32 GB of RAM and GeForce 1080Ti GPU.

The NN models are implemented in Python 3.9 using the TensorFlow v2.10.0 framework. For the backbone, we use ResNet50, ResNet101 and MobileNetV3small proposed by Keras while ResNet18 implementation is available in [24].

The complete set of experiments is composed of 120 configurations: 2 databases (UVG, VID), 4 backbones (ResNet18, 50 and 101, MobileNetV3small), 3 fingerprinting sizes (32x32, 64x64, and 128x128) and 5 NN configurations.

Models have been trained for 100 epochs, with a 64 batch size. The initial learning rate is set to 0.1 and kept unchanged during the first 10 epochs; then, an exponential relative decay of 0.15 each three epochs is considered. The training dataset is composed of 80% of the dataset. For each content in the training dataset, a Monte Carlo simulated version of the near duplicated modification is also considered.

The validation is achieved by considering 20% of the database (without any Monte Carlo simulation). Additionally, for MobileNet, if the results are not stable, 50 more epochs are considered.

5 configurations are considered during: Baseline (backbone and O3), End-to-End (I1, backbone, O1, O2, O3) and three intermediate models obtained by combining a subset of the elements presented in Fig. 2.

4.3 Result illustrations

A selection of quantitative results is illustrated in Figs. 3 and 4, as well as in Tabs. 1 and 2.

Fig. 3 and 4 consider the End-to-End model, the case of 128x128 fingerprinting size, and focus on ResNet18 and MobileNet. They show the training loss and the validation Accuracy for the UVG and VID databases, respectively. The visual analysis of the results presented in Figs. 3 and 4 show that although the same convergence value tends to be reached, this process is faster and smoother for ResNet18. The same behavior is encountered for all the investigated configurations.

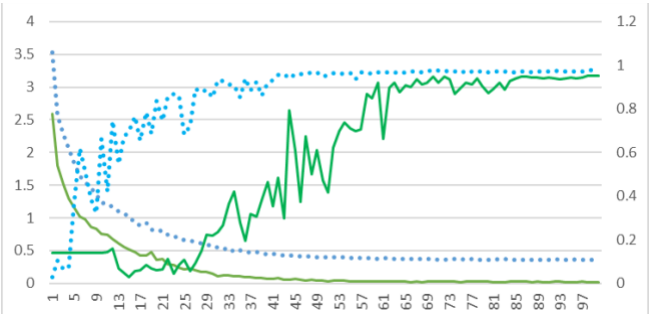

Figure 3. *End-to-End, 128x128, UVG: training loss and validation Accuracy: MobileNet (solid green) and ResNet18 (dotted blue)*

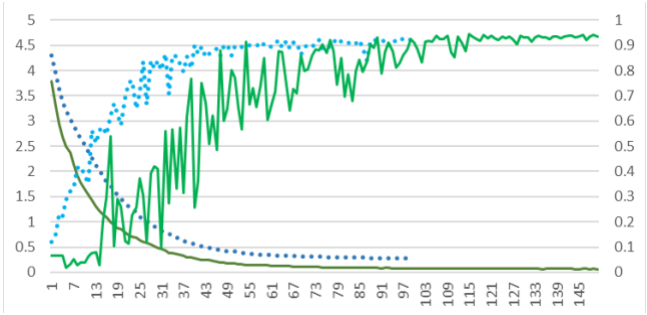

Figure 4. *End-to-End, 128x128, VID: training loss and validation Accuracy: MobileNet (solid green) and ResNet18 (dotted blue)*

Tab. 1 also focusses on ResNet18 and MobileNet, and investigates the Accuracy (Acc), Precision (Prec) and Recall (Rec), by presenting the corresponding values multiplied by 100. The columns are grouped in three areas, according to the three fingerprinting sizes discussed in the first part Method presentation section. The rows are organized at three recursive levels: firstly, according to the two databases (UVG and VID), then to the backbone NN component (ResNet18 or MobileNet), and finally according to the 5 model configurations discussed in the second part of the Method presentation section.

The values reported in Tab. 1 represent the proof of concept for achieving video fingerprinting by using only data extracted from the compressed domain.

When ResNet18 is included in the backbone, Acc values for the UVG database are higher than 0.9, irrespective to the NN configuration. Two exceptions are encountered, namely 64x64 fingerprints and B, I1 configuration (that is, when removing the O1 and O2 layers), and for 128x128 fingerprints and B, I1, O2 configuration (that is, when removing the O1 layer). It can also be noticed that the Prec and Rec values are well balanced, with average relative differences lower than 3%. When considering the VID database, the same general trend is followed, yet the configurations resulting in Acc values lower than 0.9 are different. In its turn, when included in the backbone, MobileNet results in Acc values larger than 0.8 for the UVG database while featuring some values as low as 0.567 in the case of VID database.

Tab. 1 also provides information about the usefulness of the I1, O1 and O2 layers added over the Baseline model. When considering the ResNet18 as backbone component and the UVG database, the baseline model is always outperformed by a configuration including at least one additional layer. This is not the case for the VID database where the baseline is the better solutions for 32x32 and 128x128 fingerprints, while being outperformed by the configuration including O1 and O2 in the case of 64x64 fingerprint. Also notice that the End-to-End configuration is never the better choice. When considering MobileNet as backbone, the conclusions change: the End-to-End is the best solution, with a singular exception: the 64x64 fingerprint and the UVG dataset, when it is outperformed by 0.5% by the B, I1, O2 configuration.

When comparing UVG and VID datasets, we would have expected better results on UVG, as it is a priori simpler, covering only 16 reference sequences. This result is confirmed for the End-to-End configuration but not for all the other four investigated configurations.

As a final remark related to the values in Tab. 1, we would have expected to notice a significant impact of the size of the fingerprint in the Acc: intuitively, the larger the size of the fingerprint the better the Acc. However, the results show that such a tendency is not always confirmed. Using a bigger size might drop the performance by using non relevant information (mainly composed of zeros and ones). Tab. 2 complements the detailed information provided in Tab. 1 with a global information about the cases in which different components are considered in backbone, namely ResNet50 and ResNet101. Tab. 2 presents the Acc, Prec and Rec values multiplied by 100, corresponding to the End-to-End case. The results show that the global trend brought forth by the values in Tab. 1 is kept for the new backbone components. However, Tab; 2 also shows that the claim of ResNet architecture being robust against degradation [19] seems false in the compressed domain, as ResNet50 and ResNet101 are always outperform either by ResNet18 or by MobileNet. The impact of the fingerprinting size in the Acc value is now confirmed for the ResNet18 and MobileNet, while being contradicted by ResNet50 and ResNet101.

5. Conclusion

The present study establishes the proof of concepts for HEVC-compressed domain video fingerprinting, with illustrations on two complementary databases, namely one reference database for end-to-end video encoding, and one homemade organic video database. This statement is based on applicative relevant fingerprinting performances: Acc values larger than 0.9, with well-balanced (relative differences lower than 0.03) underlying Prec and Rec.

The results correspond to fingerprints represented by I frame syntax elements (Y, Cr, Cb coefficients and their underlying prediction modes) and to NN-based models obtained by complementing widely used CNN models with pre- and postprocessing layers.

Beyond the targeted proof of concepts, the results reported in this paper open the door towards further investigations on how neural networks work in compressed domain. Hence, new transformation will be considered in the future such (*i.e.* additional

Table 1. Accuracy (Acc.), Precision (Pre.), and Recall (Rec.) according to the fingerprinting size, database, and model configuration.

			32×32			64×64			128×128		
			Acc.	*Prec.*	*Rec.*	*Acc.*	*Prec.*	*Rec.*	*Acc.*	*Prec.*	*Rec.*
UVG	ResNet18	Baseline (B)	91.2	92.2	90.6	95.9	96	95.7	96.1	96.2	95.9
		B, O1, O2	93	93.5	92.7	**98**	98.1	97.9	**98.2**	98.5	97.9
		B, I1	**95.7**	95.9	95.4	88.2	88.5	87.9	96.3	96.4	96.1
		B, I1, O2	92.8	93.1	92.4	96.6	96.8	96.4	83.9	94.5	60.8
		End-to-End	94.2	94.8	93.7	97	97	96.8	97.7	98.2	97.4
	MobileNet	Baseline (B)	81	81.5	80.2	87.1	87.1	86.8	86.4	86.7	86.2
		B, O1, O2	80	81.9	79.5	85.3	86.5	85.1	91.3	91.3	91.3
		B, I1	85.4	86.6	84.9	87	87.4	86.7	88.8	88.6	89.1
		B, I1, O2	85.4	86.6	84.9	**91.5**	92.2	91.2	89.9	92.5	89.1
		End-to-End	**87**	88.5	84.2	91	91.6	90.8	**95.2**	95.3	94.5
VID	ResNet18	Baseline (B)	**94**	96	93.1	94.6	96.3	89.6	**95.3**	97	94.6
		B, O1, O2	93.7	95.8	92.7	**95.8**	98.9	95	92.9	97.9	92.1
		B, I1	90.7	93.4	89.8	90.9	93.8	89.6	91.2	93.7	90.2
		B, I1, O2	88.3	91.5	87.4	94.3	96.5	93.4	92.7	95.9	91.8
		End-to-End	88.7	91.2	87.6	91.4	94.2	90.2	92.2	94.9	91.3
	MobileNet	Baseline (B)	59.4	66.8	55.2	88.1	90.8	86.9	90.3	92.2	89.3
		B, O1, O2	60.3	75.6	52.9	86.9	90.2	85.3	82.5	86.9	81.3
		B, I1	68.9	76.3	65.7	89.3	91.5	88.3	91.9	93.8	91.1
		B, I1, O2	56.3	71.1	48.9	90.9	92.9	90	91.9	94	91
		End-to-End	**86.6**	93.7	83	**91.8**	94.2	91	**94.2**	96.2	93.3

Table 2. Accuracy (Acc.), Precision (Pre.), and Recall (Rec.) according to the fingerprinting size, database, the End-to-End configurations and with different backbone components.

		32×32			64×64			128×128		
		Acc.	*Prec.*	*Rec.*	*Acc.*	*Prec.*	*Rec.*	*Acc.*	*Prec.*	*Rec.*
UVG	ResNet18	**94.2**	94.8	93.7	**97**	97	96.8	**97.7**	98.2	97.4
	ResNet50	81.5	87.8	80.2	87.4	89	87.2	93.2	93.5	92.5
	ResNet101	87.1	88.8	86.6	91.8	92.6	91.7	85.4	88.1	84
	MobileNet	87	88.5	84.2	91	91.6	90.8	95.2	95.3	94.5
VID	ResNet18	**88.7**	91.2	87.6	91.4	94.2	90.2	92.2	94.9	91.3
	ResNet50	87.1	90.2	86	88.7	91.1	87.7	86.3	88.7	85.4
	ResNet101	74.5	80.1	72.4	88.3	90.7	87.5	84.6	87.8	83.3
	MobileNet	86.6	93.7	83	**91.8**	94.2	91	**94.2**	96.2	93.3

codec) Note that several a priori expectations have not been confirmed by our study and future work will be done on explaining why larger fingerprinting sizes or deeper ResNet structures do not necessary result in improved results. A different research direction relates to the explanation of the relationship between the content semantic and the NN in the backbone: note that a finer comparison of the impact of the complementary layers (I1, O1 and O2) in the results show different trends for UVG and VID and, at the same time, for ResNet18 and MobileNet.

Acknowledgment

Mohamed Allouche's PhD program was financed by VIDMIZER (https://cutz.cloud/) under the CIFRE framework. The present research study was conducted in the absence of any commercial or financial constraint that could be considered as a potential conflict of interest.

References

[1] Statista. (last visited: March 2022). Global Data Volume of Internet Video to TV Traffic. *https://www.statista.com/statistics/267222*

[2] Hubspot (last visited: March 2024). Organic Marketing vs. Paid Marketing. https://blog.hubspot.com/marketing/organic-marketing

[3] M. Douze, A. Gaidon, H. Jegou, M. Marszałek, and C. Schmid, "INRIA-LEAR's Video Copy Detection System" TRECVID, 2008.

[4] Y. G. Jiang, and J. Wang, "Partial Copy Detection in Videos: A Benchmark and an Evaluation of Popular Methods," IEEE Transactions on Big Data, vol. 2, no. 1, pp. 32–42, 2016. doi:10.1109/TBDATA.2016.2530714

[5] M. Allouche, and M. Mitrea, "Video Fingerprinting: Past, Present, and Future," Frontiers in Signal Processing, 2, article 984169, 2022.

[6] F. Idris and S. Panchanathan, "Review of image and video indexing techniques," Journal of Visual Communication and Image Representation, vol. 8, no. 2, pp. 146–166, 1997. doi:10.1006/jvci.1997.0355

[7] J. Benois-Pineau, "Indexing of compressed video: Methods, challenges, applications," in International Conference on Image Processing Theory, Tools and Applications, pp. 3–4, 2010.

[8] F. Manerba, J. Benois-Pineau, R. Leonardi, and B. Mansencal, "Multiple Moving Object Detection for Fast Video Content Description in Compressed Domain," EURASIP Journal on Advances in Signal Processing, pp. 1–15, 2008.

[9] V. Mezaris, I. Kompatsiaris, N. V. Boulgouris, and M. G. Strintzis, "Real-time compressed-domain spatiotemporal segmentation and ontologies for video indexing and retrieval," IEEE Transactions on Circuits and Systems for Video Technology, vol. 14, no. 5, pp. 606–621, May 2004.

[10] A. Mercat, M. Viitanen, and J. Vanne, "UVG Dataset: 50/120fps 4K Sequences for Video Codec Analysis and Development," in Proceedings of the ACM Multimedia Systems Conference, Istanbul, Turkey, June 2020.

[11] X. Wu, W. Zhao, and C. W. Ngo, "Near-Duplicate Keyframe Retrieval with Visual Keywords and Semantic Context," in Proceedings of the 6th ACM International Conference on Image and Video Retrieval (CIVR'07), pp. 162–169, 2007.

[12] C. W. Ngo, M. Yu-Fei, and J. Z. Hong, "Video Summarization and Scene Detection by Graph Modeling," IEEE Transactions on Circuits and Systems for Video Technology, vol. 15, pp. 296–305, 2005.

[13] R. Schuster, V. Shmatikov, E. Tromer, "Beauty and the Burst: Remote Identification of Encrypted Video Streams", 26th USENIX Security Symposium (USENIX Security 17). USENIX Association, Vancouver, BC, pp. 1357–1374, 2017.

[14] Y. Li, Y. Huang, R. Xu, S. Seneviratne, K. Thilakarathna, A. Cheng, D. Webb, G. Jourjon, "Deep Content: Unveiling Video Streaming Content from Encrypted WiFi Traffic", 2018 IEEE 17th International Symposium on Network Computing and Applications (NCA). Cambridge, MA, pp. 1–8, 2018.

[15] W. Afandi, S. M. A. H. Bukhari, M. U. S. Khan, T. Maqsood and S. U. Khan, "Fingerprinting Technique for YouTube Videos Identification in Network Traffic," in IEEE Access, vol. 10, pp. 76731-76741, 2022.

[16] ISO/IEC 23008-2:2013 High Efficiency Video Coding.

[17] M. Hasnaoui, and M. Mitrea, "Multi-symbol QIM Video Watermarking," Signal Processing: Image Communication, 29(1), 107-127. ISSN 0923-5965, 2014.

[18] A. Garboan, and M. Mitrea, "Live Camera Recording Robust Video Fingerprinting," Multimedia Systems, vol. 22, pp. 229–243, 2016.

[19] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, pp. 770-778, 2016.

[20] A. Howard, M. Sandler, G. Chu, L.-C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan et al. "Searching for MobileNetV3," 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Korea (South), pp. 1314-1324, 2019.

[21] J. Bend, "The YLI-MED Corpus: Characteristics, Procedures, and Plans," ICSI Technical Report TR-15-001, pp. 1–46, 2015.

[22] S. Abu-El-Haija, N. Kothari, J. Lee, A. Natsev, G. Toderici, B. Varadarajan, et al., "YouTube-8M: A Large-Scale Video Classification Benchmark," ArXiv, abs/1609.08675, 2016.

[23] VideoLan (last visited: March 2024). x265. https://www.videolan.org/developers/x265.html

[24] Jerett, "Resnet.py", Keras-CIFAR10. Main branch (last visited: March 2024). https://github.com/jerett/Keras-CIFAR10/blob/master/classifiers/ResNet.py

Author Biography

Mohamed Allouche received his PhD degree from Institut Polytechnique de Paris (December 2024), and the MS degree from National Engineering School of Sfax (2020). His research interests cover green video encoding and its subsequent tracking, as well as cloud-based video processing platforms. Mohamed Allouche actively contributes to MPEG standardization activities.

Mihai Mitrea is a Professor at Telecom SudParis, Institut Polytechnique de Paris. He holds an HDR degree from Pierre and Marie Curie University in Paris (2010) and a PhD from Politechnica University in Bucharest (2003). He is vice-president of the Cap Digital's Technical Commission on Digital Content and serves as advisor for the French delegation at ISO/IEC JTC1 SC29 (a.k.a. MPEG). Inside MPAI standardization organization, he is coordinating Neural Network Watermarking efforts.

Carl De Sousa Trias received both the M.S. degree in High Tech Imaging from Telecom SudParis and the M.S degree in Virtual & Augmented Reality from Institut Polytechnique de Paris in 2021. He is currently pursuing a Ph.D. degree in neural network watermarking and video compression at Institut Polytechnique de Paris and is an active contributor to MPAI Neural Network Watermarking standardization efforts.