https://doi.org/10.2352/EI.2025.37.10.IPAS-234

Stream encoder identification in green video context

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

Abstract

Video streaming hits more than 80% of the carbon emissions generated by worldwide digital technologies consumption that, in their turn, account for 5% of worldwide carbon emissions. Hence, green video encoding emerges as a research field devoted to reducing the size of the video streams and the complexity of the decoding/encoding operations, while keeping a preestablished visual quality. Having the specific view of tracking green encoded video streams, the present paper studies the possibility of identifying the last video encoder considered in the case of multiple reencoding distribution scenarios. To this end, classification solutions backboned by the VGG, ResNet and MobileNet families are considered to discriminate among MPEG-4 AVC stream syntax elements, such as luma/chroma coefficients or intra prediction modes. The video content sums-up to 2 hours and is structured in two databases. Three encoders are alternatively studied, namely a proprietary green-encoder solution, and the two by-default encoders available on a large video sharing platform and on a popular social media, respectively. The quantitative results show classification accuracy ranging between 75% to 100%, according to the specific architecture, sub-set of classified elements, and dataset.

1. Introduction

According to statistics published by French public organizations like ADEME and ARCEP [1], greenhouse gas emissions related to digital technology represent 5% of the worldwide global emissions. For a comparison, this is equivalent to the emission from the aviation industry as early as 2020 [2]. Also brought forth by the same organizations, data centers and networks account for 8 to 33% from the worldwide digital carbon footprint and have as main component the emissions related to video stream production, transmission, storage and consumption. Since the demand for video-related solutions will keep increasing during the next years, studying and understanding encoder profiles becomes valuable asset to reduce their environmental impact.

Green encoding for video is currently emerging as a scientific and technical field devoted to the carbon footprint reduction for video industry. Under this framework, one of the challenges is to specify an encoding profile minimizing (from theoretical and/or practical points of view) the size of the produced stream. Such an approach is mainly useful for video contents that are expected to be streamed many times (e.g. 10^8 views or more). This way, a reduction from 20% to 50% from the original carbon emission can be obtained with no impact in the quality of the experience perceived by the final user (i.e. with no reduction in the decompressed visual quality, no extra delay in decompression time, etc.).

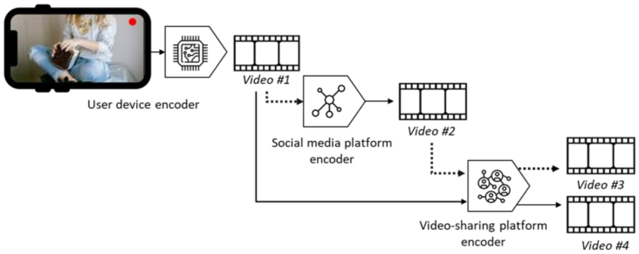

Figure 1. *Synopsis of the paper objective: automatically distinguish the last encoder used for a given video sequence.*

While video encoding can be a palliative to greenhouse gas emission, the nowadays usage of video content can reduce or even cancel such beneficial effect. For instance, when posted on a social media, the encoding properties of a video stream are changed. Moreover, such a phenomenon occurs for every successive sharing/posting in different social media or video distribution platforms and it is *a priori* likely to result in a regular video stream, as illustrated in Figure 1.

The present paper investigates the possibility of discriminating the video streams obtained from a same visual content but with different encoders and/or combination of encoders: green encoders, by-default social media encoders, video distribution platform encoders, *etc.*

Throughout the rest of the paper, no explicit reference to the green encoder technology and/or to the specific social media / video distribution platform will be made, in order to avoid any potential conflict of interest. Note that such platforms do not disclose the encoding strategies they deploy and consequently, identifying the encoder in a video stream pipeline becomes a complex task, with no *a priori* analytic solution and requiring a deep learning (DL)-based approach.

To this end, we start by considering a total of 46 minutes of original video content that is alternatively encoded by a proprietary green-encoding solution, and the two by-default encoders available on a large video sharing platform and on a popular social media, respectively. Then, the possibility of using for this task DL-based classifiers backboned by VGG, ResNet and MobileNet families is studied. The quantitative results are expressed in terms of *Accuracy, Precision*, and *Recall*.

This paper is structured as follows. Section 2 introduces the grounds of our study, from both applicative and support tools points of views. Section 3 describes the classification methods, emphasizing on the adaptation required for using popular DL pixel-based image classifiers for compressed stream typology identification. Experimental results are presented and discussed in Section 4 while Section 5 concludes the paper and opens further working directions.

2. State-of-the-art

2.1 Video encoding

Video encoding deals with the design and development of an encoding/decoding system that fulfills the dilemma of reducing the weight of the video by large factors (beyond 10 000) while ensuring a prescribed visual quality for the decoded content. To this end, a large variety of methods, belonging to 4 types of operations (namely, *Prediction*, *Transformation*, *Quantization* and *Entropy coding*) are integrated into a processing pipeline, as illustrated in Figure 2 (for the decoding part).

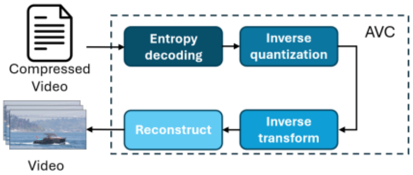

Figure 2. *Video decompression pipeline.*

Processing compressed video comes across with a conceptual contradiction: assuming the compression reached its goal, no more redundancy exists in the data, thus making it impossible for applications exploiting the visual redundancy to be performed.

Hence, as an *a priori* ground, we can expect better results to be obtained from intermediate representations, available in-between the compressed stream syntax elements and pixels. Moreover, we can expect the elements extracted in-between Inverse Quantizing and Transform to reach a convenient trade-off among the initial challenge of processing the compressed stream and the possibility of being processed by algorithms directly extended from the pixel domain.

2.2 Encoded stream identification

The problem of identifying video encoders and/or encoding parameters (be they related to a green approach or not) is studied from various points of view.

The various means of lowering the CO_2 emissions related to video consumption are summarized in [3] and relate to the ingested video source, the encoding parameters, the packaging, the delivery and the displaying conditions. Specifically related to the encoding parameters selection, it is shown that their dynamic matching to the visual content can be achieved by an AI-based estimation.

In [4], for each video segment, a home-made fully connected network is trained to find the CRF (Constant Rate Factor) enabling the highest possible encoding quality while adhering to Video Multi-Method Assessment Fusion (VMAF) [5] quality parameters.

With the similar target of identifying optimal encoding parameters, the study in [6] brings to light the possibility of identifying presets in per-title encoding. Thus, an experimental study investigates the 10 MPEG HEVC presets, and identifies optimal configurations that balance encoding time, energy consumption, and compression efficiency.

The reduction energy consumption on the display side is studied in [7]. To this end, additional metadata extracted from the original content can be inserted in the compressed video bitstream to provide detailed information about the decoding complexity of each segment. This way, the decoder efficiently controls the power consumption and distinguish between video streams based on their complexity profiles.

Complementary to [3], [4], [6], [7], several studies [8], [9] are devoted to identifying categories of encoded video contents, as follows. In [8], a neural network is defined to identify the alteration (defined as voluntary content modifications induced by a video editing software) and the source (defined as a manufacturer for the recording mobile device or as a social media platform) of a video stream. To this end, integrity information is extracted per group of pictures and then fused and fed to a classifier. Other work like [9] analyses the network's traffic to detect the content of a video-sharing platform.

This concise state of the art brings to light that the problem of green optimizing video compression is very broad, with no predilection solution: hence, identifying the last encoder opens the door to video tracking and potentially fosters for the needs of green video distribution.

2.3 Conventional DL classifiers

In a nutshell, a deep learning model classifier aims to associate some input data to a category or a class, based on a specific set of features. This generic definition is declined in myriads of practical solutions optimized to solve a huge variety of image processing applications, by processing the pixels of a visual content. Such solutions share a conventional processing pipeline, as illustrated in Figure 3: database creation, data loader, and classifier model.

Video data
Database
Dataloader
Neural Network
class:

Figure 3. *Processing pipeline of deep learning-based classification*

Nowadays, research studies benefit from large databases that are already preprocessed and annotated, like ImageNet for image classification, CelebA dataset for face segmentation and COCO for object detection, for instance. However, the original video content beneath these databases is not made available. When considering compressed-domain video processing, the situation completely changes with very few (if any) available databases.

The data loaders available today mainly perform basic preprocessing operations (like resizing or normalization) corresponding to the R, G and B channels. This way, a common data format can be presented to the model.

The classifier model architectures vary in the number and types of layers, the number of neurons of each layer, and the topology between the layers. Currently, 3 families of convolutional models are intensively considered, namely VGG [10], ResNet [11], and MobileNet [12]. The first in chronological order, VGG presented outstanding results on the ImageNet challenge. However, its limitations become evident when applied on more complex tasks, that cannot be fulfilled even when increasing the number of hidden layers beyond 19. ResNet tackles this issue by introducing residual connections in-between layers, thus reaching up to 152, while also stabilizing the loss function during convergence. Finally, MobileNet was designed to feature the same applicative advantages while reducing the number of parameters.

While the paragraphs above are consensual when processing pixel representations of the visual content, no hint about their validity exist for processing elements extracted from the compressed representations.

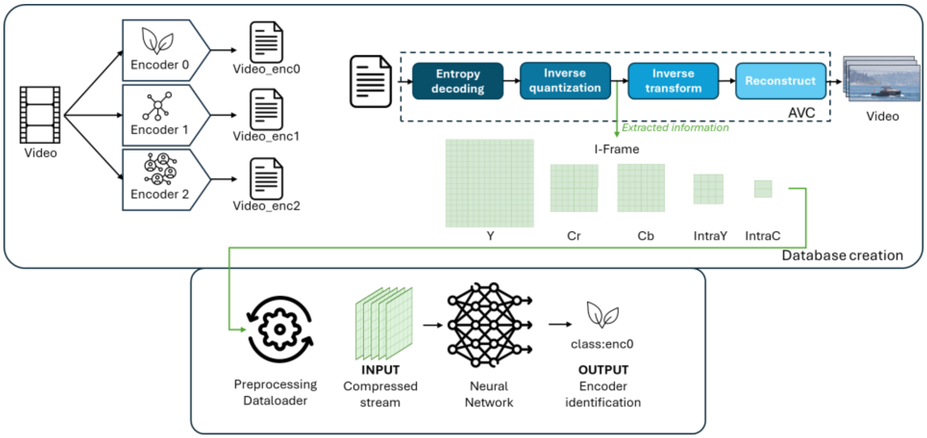

Figure 4. *Overall workflow of the proposed method.*

3. Method presentation

As our study targets compressed domain processing, the three elements presented in Figure 3 should be reconsidered for integration into a functional pipeline. The overall process of our method can be seen in Figure 4.

3.1 Dataset

The original video content corresponds to two databases, namely the state-of-the-art UVG uncompressed sequences [13] and a home-made industrial database. Note that our task needs access to the compressed stream and not a given representation (*e.g.* tensors in Youtube-8M [14]).

The UVG dataset is composed of 16 raw videos sequences, 14 of them of 24 seconds and 2 of them of 12 seconds. While this dataset is designed as benchmark dataset for the end-to-end video encoding algorithms, it is also valuable in our study, as it represents a large variety of content.

The Industrial dataset is composed of 96 video sequences provided by an industrial partner. The video content is business oriented, any mainly composed of commercials, tutorials, internal communications, and news. All video sequences have a duration between 15 and 50 seconds, while summing up to 40 min.

Regardless of the video source, for the purpose of this study, three variants of encoded content using different encoding parameters of MPEG-4 AVC. The first category is obtained by encoding the original content with a proprietary encoding configuration designed to decrease the CO_2 consumption, referred to as green-encoded. The second category is obtained by uploading the original content on a video distribution platform and by subsequently downloading it. Finally, the third category is obtained by uploading the original content on a social media platform and by subsequently downloading the resultant video.

Figure 5 presents the syntax elements extraction during the decoding process of the video. In our study, we developed a parsing module integrated with the MPAEG-4 AVC reference software [14]. The output of the implemented method is a group of syntax elements: luminance (Luma), chrominance (Chroma), and intra-frame prediction modes (Intra Prediction) per I frame. Among the 4 main steps of video decoding, only the entropy decoding and the inverse quantization are required in our method.

After this extraction process, 176 I-frames and 4120 I-frames are obtained for the UVG dataset and the Industrial dataset, respectively. The UVG dataset accounts for 59 I-frames belonging to the green-encoded class, 74 I-frames to the video sharing platform class, and 46 I-frames to the social media platform class. In its turn, the Industrial dataset results in 1410 I-frames belonging to the green-encoded class, 1872 I-frames to the video sharing platform class, and 838 I-frames to the social media platform class.

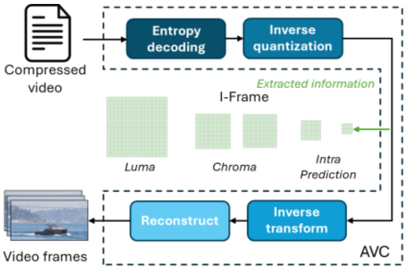

Figure 5. *Pipeline for the data extraction process from the compressed video using the reference software and a parser.*

3.2 Data loader

The 5 syntax elements extracted from an I-frame present a variety of sizes (as illustrated in Figure 4) that rises an additional challenge compared to a typical deep learning classifier. To be processed by a data loader, the input data should be a matrix of fixed size for all the 5 channels. The luminance (Y) is a 16×16 matrix, both chrominances (Cr) (Cb) are 8×8 matrixes, and each intra prediction modes (IntraY) (intraC) have their own size ranging from 1 to 4×4. Hence, a smart padding is applied to associate each luminance element with the right chrominances and intra predictions modes. Each padded macroblock is then inserted in its corresponding position in the final matrix. The label for the obtained input corresponds to its category of video encoding.

The obtained labeled data is then down sampled to the size of the smallest I-frame in each dataset. No additional normalization or data augmentation is applied since the nature of the input data differs from the pixel-domain task.

3.3 Classifier model

The given task is considered to be a 3-class classification problem. This study investigates the VGG, MobileNet, and ResNet families. Specifically, ResNet18, ResNet50 and ResNet101 are considered to study the impact of the DL depth in the results.

While the main task adaptation is produced by the data loader, the number of input channels of the first layer needs to be modified. As hyperparameters for training, we used the cross-entropy loss, the stochastic gradient descent optimizer with a learning rate adapted to each model.

4. Experimental results

The experimental setup considers 5 architectures namely VGG11, MobileNetV3small, ResNet18, ResNet50 and ResNet101, as well as 3 configurations for each dataset obtained by changing the type of the processed syntax elements: luminance features (Y), luminance and chrominances features (Y, Cr, Cb), and luminance, chrominances and intra prediction modes features (Y, Cr, Cb, IntraY, IntraC). For each configuration, the datasets are split between trainset (80%) and testset (20%).

Each model is trained for 25 epochs using an SGD optimizer. For the MobileNet and ResNet configurations, the initial learning rate was set to 0.01 and decreased by 10 after the 15th and 20th epochs. For the VGG network, the initial learning rate was set to 0.0001 and decreased the same way. The experiments are performed using an in-premises server with the following configuration: i9-11950H processor @2.60GHz, 8 threads CPU, 32GB of RAM and RTX A4000 GPU.

A selection of quantitative results is illustrated in Figures 6 and 7, as well as in Tables 1 and 2.

Figure 6 displays the training loss evolution for ResNet18 and MobileNet models on the (Y, Cr, Cb) UVG configuration. Despite having similar learning curves, ResNet18 achieves *Acc.* = 1 while MobileNet achieves *Acc.* = 0.75. Figure 7 goes into detail for this situation and displays the MobileNet confusion matrix: it can be noticed that the social media encoder cannot be identified by this model.

Table 1 and Table 2 show the results corresponding to the UVG and Industrial datasets, respectively. Note that the reported Accuracy (*Acc.*), Precision (*Prec.*) and Recall (*Rec.*) values are multiplied by 100 and are computed as unweighted mean values. The columns correspond to the selected models. The rows are organized at two imbricated levels: firstly, according to the three-dataset configuration (Y; Y, Cr, Cb; Y, Cr, Cb, IntraY, IntraC), then to the three metrics.

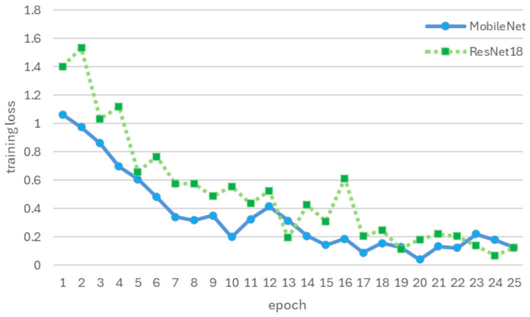

Figure 6. Training loss evolution of ResNet18 and MobileNet on the UVG dataset with the (Y, Cr, Cb) components.

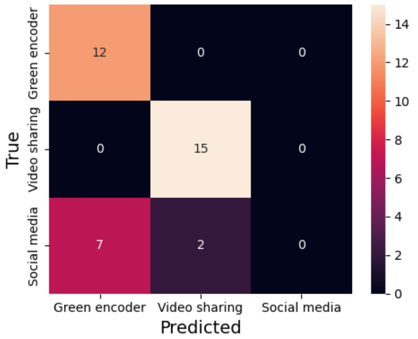

Figure 7. Confusion matrix of MobileNet on the UVG dataset with the (Y, Cr, Cb) configuration.

In Table 1, VGG11, ResNet18, ResNet50 bring to light that the performance increases (or stay constant) when considering additional syntax elements during the classification. On the contrarily, ResNet101 and MobileNet show the performance depreciation when adding the intra prediction modes as classification data.

This can be explained by the fact that intra-prediction modes are intrinsically linked to the content, and consequently less dependent with the encoding configuration.

TABLE 1. Classification performance on UVG dataset

		Architectures				
	metrics	*VGG11*	*MobileNet*	*ResNet18*	*ResNet50*	*ResNet101*
Y	*Acc.*	77.77	38.89	69.44	94.44	86.11
	Prec.	84.13	45.10	81.66	94.10	88.10
	Rec.	74.81	37.77	71.66	95.55	88.33
(Y, Cr, Cb)	*Acc.*	83.33	**75.00**	**100**	**100**	**94.44**
	Prec.	87.78	50.47	100	100	94.10
	Rec.	77.78	66.67	100	100	95.56
(Y, Cr, Cb, IntraY, IntraC)	*Acc.*	**91.66**	47.22	**100**	**100**	77.80
	Prec.	93.30	46.23	100	100	80.74
	Rec.	88.8	44.44	100	100	80.74

TABLE 2. Classification performance on Industrial dataset

		Architectures				
	metrics	*VGG11*	*MobileNet*	*ResNet18*	*ResNet50*	*ResNet101*
Y	*Acc.*	**99.62**	97.57	98.98	**99.10**	**98.46**
	Prec.	99.71	97.66	98.98	99.15	98.54
	Rec.	99.60	98.11	98.98	99.25	98.72
(Y, Cr, Cb)	*Acc.*	98.59	98.33	**99.48**	98.72	97.83
	Prec.	98.62	98.31	99.61	98.70	97.67
	Rec.	98.88	98.66	99.47	98.91	98.23
(Y, Cr, Cb, IntraY, IntraC)	*Acc.*	98.47	**98.43**	99.23	96.42	95.27
	Prec.	98.59	98.36	99.22	96.2	95.92
	Rec.	98.61	98.72	99.41	97.26	94.96

Also note that such a behavior also depends on the processed content: in Table 2, MobileNet is the only architecture increasing its performance with the typology of the processed syntax elements. For VGG11, ResNet50, and ResNet101 the best *Acc.* is obtained when Y is solely processed, while ResNet18 provides the best *Acc.* value for (Y, Cr, Cb).

The values reported in Table 1 and Table 2 proves that encoders can be classified according to the information extracted in the compressed domain:

- for the UVG database, an *Acc.* = 100 is obtained for ResNet18 and ResNet50 and for (Y, Cr, Cb) or (Y, Cr, Cb, IntraY, IntraC),
- for the Industrial dataset, an *Acc.* = 99.62 is obtained for VGG11 and only the Y component.

These *Acc.* Values come across with well balances *Prec.* and *Rec*. values.

While the results presented in Tables 1 and 2 achieve our task, they also bring to light some insights on how popular DL solutions designed for processing pixels work when applied to compressed domain.

First, VGG11 needs a hundred times lower learning rate compared to the other models, while, as rule of thumb, only a ten-times lower rate is enough in the pixel domain. Note that for UVG database, the MobileNet cannot solve this 3-class classification problem, with the best *Acc.* = 0.75 corresponding to (Y, Cr, Cb) syntax elements. Yet, the small size of the database cannot be invoked as an explanation, as architectures with larger number of parameters were successfully trained (*e.g.* ResNet101). Also note that ResNet101 is always outperformed by ResNet18 and ResNet50.

5. Conclusion

The present study establishes the proof of concepts for the possibility of detecting the last video encoder considered in the case of multiple reencoding distribution scenarios. This PoC is based on a study investigating 5 DL classifiers acting directly in the compressed domain, processing more than 2h of video content and resulting on the *Acc.* values larger than 0.99 (even 1), with well-balanced *Prec*. and *Rec* (larger than 0.98).

The results of this study for each type of architecture provide a foundation for future research into how CNN models behave in a compressed domain.

Note that identifying the encoder in a video stream pipeline can serve as a method to assess the emissions of a video platform thus serving as an asset in reducing the end-to-end video delivery carbon footprint.

Beyond video encoder detection, the results presented in this paper will be reconsidered and extended in future work for video content tracking: the capability of the investigated classifiers to discriminate the content suggests that the compressed domain implicitly brings a particular kind of information that is lost during the decoding procedure.

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] "The Environment," Arcep. Accessed: May 24, 2024. Available: https://en.arcep.fr/news/press-releases/view/n/the-environment-190122.html

[2] "Why your internet habits are not as clean as you think." Accessed: May 24, 2024. Available: https://www.bbc.com/future/article/20200305-why-your-internet-habits-are-not-as-clean-as-you-think

[3] R. Seeliger, C. Muller, and S. Arbanowski, "Green streaming through utilization of AI-based content aware encoding," in 2022 IEEE International Conference on Internet of Things and Intelligence Systems (IoTaIS), Nov. 2022, pp. 43–49. doi: 10.1109/IoTaIS56727.2022.9975919.

[4] F. Mico-Enguidanos, W. Moina-Rivera, J. Gutierrez-Aguado, and M. Garcia-Pineda, "Per-title and per-segment CRF estimation using DNNs for quality-based video coding," Expert Systems with Applications, vol. 227, p. 120289, Oct. 2023, doi: 10.1016/j.eswa.2023.120289.

[5] N. T. Blog, "Toward A Practical Perceptual Video Quality Metric," Medium. Accessed: May 24, 2024. [Online]. Available: https://netflixtechblog.com/toward-a-practical-perceptual-video-quality-metric-653f208b9652

[6] H. Amirpour, V. V. Menon, S. Afzal, R. Prodan, and C. Timmerer, "Optimizing Video Streaming for Sustainability and Quality: The Role of Preset Selection in Per-Title Encoding," presented at ICME 2023, IEEE Computer Society, Jul. 2023, pp. 1679–1684. doi: 10.1109/ICME55011.2023.00289.

[7] F. C. Fernandes, X. Ducloux, Z. Ma, E. Faramarzi, P. Gendron, and J. Wen, "The Green Metadata Standard for Energy-Efficient Video Consumption," IEEE MultiMedia, vol. 22, no. 1, pp. 80–87, Jan. 2015, doi: 10.1109/MMUL.2015.18.

[8] Y. Li et al., “Learning Hierarchical Fingerprints via Multi-Level Fusion for Video Integrity and Source Analysis,” IEEE Transactions on Consumer Electronics, vol. 70, no. 1, pp. 3414–3424, Feb. 2024, doi: 10.1109/TCE.2024.3357977.

[9] 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,” IEEE Access, vol. 10, pp. 76731–76741, 2022, doi: 10.1109/ACCESS.2022.3192458.

[10] K. Simonyan, A. Zisserman. “Very Deep Convolutional Networks for Large-Scale Image Recognition.” in ICLR 2015, Computational and Biological Learning Society, 2015, pp. 1–14.

[11] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” in CVPR 2016, Las Vegas, NV, USA: IEEE, Jun. 2016, pp. 770–778. doi: 10.1109/CVPR.2016.90.

[12] A. G. Howard et al., “MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications.” arXiv, Apr. 16, 2017. doi: 10.48550/arXiv.1704.04861.

[13] A. Mercat, M. Viitanen, and J. Vanne, “UVG dataset: 50/120fps 4K sequences for video codec analysis and development,” in Proc. ACM Multimedia Syst. Conf., Istanbul, Turkey, June 2020.

[14] 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.

[15] “jvet / JM · GitLab,” GitLab. Accessed: May 24, 2024. Available: https://vcgit.hhi.fraunhofer.de/jvet/JM

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.

Elliot Cole 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 2024.

Mateo Zoughebi 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, France in 2024. He is currently pursuing a Ph.D. degree in deepfake detection in 6G context at Institut Polytechnique de Paris.

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.

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 and JPEG). Inside MPAI standardization organization, he is coordinating Neural Network Watermarking efforts.