Fast AV1 Local Warped Motion Compensation Using Machine Learning

William Kolodziejski
wkolodziejski@inf.ufpel.edu.br
Federal University of Pelotas (UFPel)
Video Technology Research Group
Graduate Program in Computing
Pelotas, RS, Brazil

Leonardo Braga
lbraga@inf.ufpel.edu.br
Federal University of Pelotas (UFPel)
Video Technology Research Group
Pelotas, RS, Brazil

Marcelo Rezende
marcelo.pr@inf.ufpel.edu.br
Federal University of Pelotas (UFPel)
Video Technology Research Group
Pelotas, RS, Brazil

Marcelo Porto
porto@inf.ufpel.edu.br
Federal University of Pelotas (UFPel)
Video Technology Research Group
Graduate Program in Computing
Pelotas, RS, Brazil

Luciano Agostini
agostini@inf.ufpel.edu.br
Federal University of Pelotas (UFPel)
Video Technology Research Group
Graduate Program in Computing
Pelotas, RS, Brazil

ABSTRACT

The growing demand for high-efficiency video compression has driven the development of advanced codecs like AV1, which achieve superior compression rates but face challenges related to computational complexity. This paper addresses these challenges by proposing a machine learning-based optimization for the AV1 Local Warped Motion Compensation (LWMC) tool. This solution uses a Decision Tree model to skip unnecessary LWMC executions, reducing its processing time by 52% while maintaining a low impact on coding efficiency of only 0.21% in BD-BR. Compared to complete LWMC deactivation, our method demonstrates significantly better performance, particularly for content with complex motion patterns. To the best of the author's knowledge, this is the first work in the literature to explore machine learning-based solutions applied to the AV1 LWMC tool.

KEYWORDS

Video Coding, AV1, Local Warped Motion Compensation, Complexity Reduction, Machine Learning

1 INTRODUCTION

With the ever-increasing demand for high-quality, low-bandwidth video streaming, video compression is essential for optimizing multimedia content delivery. The last decade has witnessed a significant shift in media consumption toward streaming services like Netflix, Amazon Prime, Twitch, and YouTube. As the demand for video quality increases, compression becomes increasingly complex. Efficient encoders are essential for reducing storage requirements, transmission costs, and ensuring smooth playback.

The AV1 is an open and royalty-free video codec developed by the Alliance for Open Media (AOM) [2]. This consortium includes big tech companies such as Amazon, Cisco, Google, Microsoft, Netflix, and many others. Released in 2018 as the successor to Google's VP9 [26], AV1 was developed to target high-performance video delivery, achieving up to 30% better compression efficiency than its predecessor [13], while maintaining visual quality.

Despite its high compression gains, the AV1 encoding process is significantly more computationally expensive than its predecessors, taking about 55 times longer to encode than VP9 [20]. The AV1's complexity arises from its advanced prediction, transformation techniques, and motion compensation methods [3], which improve compression efficiency at the cost of increased encoding time and greater computational overhead.

Among AV1's many tools, Warped Motion Compensation modes represent a more advanced approach to inter-prediction, employing affine transformations to model complex motion patterns that go beyond simple translations, such as scaling, rotation, shearing, and perspective changes. This allows for more accurate predictions in scenarios with intricate object movements or camera motion. AV1 includes two warped motion modes: Global Warped Motion Compensation (GWMC) and Local Warped Motion Compensation (LWMC) [21].

This work proposes skipping the LWMC execution when it is not needed, thereby reducing encoding time and, consequently, the computational cost of the AV1 encoder. A Decision Tree-based machine learning model was trained to predict whether the tool would be used in the encoding process. The model fully complies with the AV1 standard and does not affect the bitstream syntax.

2 BACKGROUND

Video coding explores temporal redundancy between frames, with motion compensation being a key technique in modern codecs. This is typically implemented through block-based motion compensation, where a frame is divided into blocks predicted from corresponding blocks in a previously encoded frame. The motion assumed in this process is generally translational, as used in most modern codecs.

However, real-world motion in videos is often more complex. Rotations, such as those caused by handheld camera shake, and

In: Proceedings of the Brazilian Symposium on Multimedia and the Web (WebMedia'2025). Rio de Janeiro, Brazil. Porto Alegre: Brazilian Computer Society, 2025.

ISSN 2966-2753

distortions due to object movement, introduce motion components beyond simple translation. Panning and zooming also contribute to motion that a purely translational model cannot fully capture. While smaller block sizes allow for a closer approximation of such motion, they increase computational costs and reduce coding efficiency. Examples of these translational and non-translational motions are presented in Figure 1.

The AV1 codec introduces warped motion models to enhance prediction efficiency for sequences with strong non-translational motion. Its Global Warped Motion Compensation (GMWC) tool explicitly encodes frame-level transformations, particularly for camera motion, applying them selectively when beneficial. Additionally, the Local Warped Motion Compensation (LWMC) tool estimates transformations at the block level using neighboring motion vectors, allowing for an adaptive representation of object motion. Both tools compete with traditional translational Motion Compensation, ensuring they are only used when they provide a coding advantage [6]. In these cases, an affine motion model is applied to represent the transformation more accurately.

The transformation applied in both GWMC and LWMC follows a homography model. Given a pixel at a position (x, y) in the current frame, its warped coordinates (x', y') in the reference frame can be computed using the affine transformation in (1):

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (1)$$

The transformation matrix contains six degrees of freedom, enabling flexible warping, such as rotation, scaling, and shearing. The parameters h_{13} and h_{23} define the translational component of motion. To reduce computational complexity, these values are assumed to represent the motion vector (MV) of the current block (MV_1 in Figure 2), where $MV_1 = \binom{h_{13}}{h_{23}}$. This assumption simplifies the transformation to four degrees of freedom, restricting the mapping to a parallelogram while reducing processing costs.

Warped motion prediction is performed by projecting each pixel within a block to a new position in the reference frame. As shown in Figure 2, the motion vector $(u_x, u_y) = MV_1$ is applied to every pixel in the current block (blue square), producing the warped coordinates $(x', y') = (x + u_x, y + u_y)$. This projection is carried out for all neighboring blocks (green squares), referencing the same frame. Since the translational parameters h_{13} and h_{23} are omitted from explicit signaling, the source coordinates are shifted to the block center, and the motion vector is directly added to the projected destination. The model is then simplified to (2):

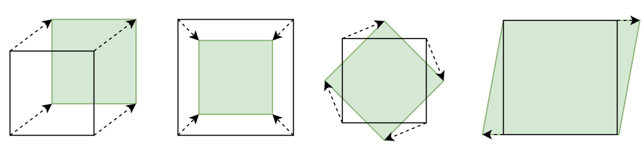

Figure 1: Example of translational and non-translational motion.

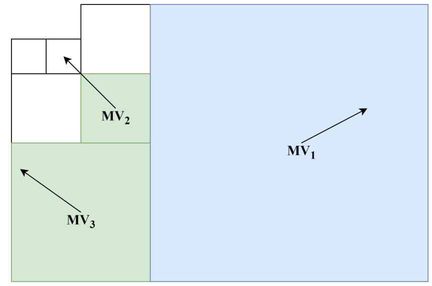

Figure 2: Example of MVs of neighboring blocks.

$$x' = h_{11}x + h_{12}y, \quad y' = h_{21}x + h_{22}y \quad (2)$$

To obtain the best-fit transformation, the parameters h_{11}, h_{12}, h_{21}, and h_{22} are estimated using a least-squares approach, minimizing the difference between reference pixels and warped projections.

To refine the motion model, AV1 LWMC decomposes the affine transformation into two sequential shear operations: one horizontal and one vertical. This structure facilitates the application of interpolation filters, improving accuracy. The decomposition process is defined in (3):

$$\begin{bmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ \gamma & 1+\delta \end{bmatrix} \begin{bmatrix} 1+\alpha & \beta \\ 0 & 1 \end{bmatrix} \quad (3)$$

The final transformation is then expressed as:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ \gamma & 1+\delta \end{bmatrix} \begin{bmatrix} 1+\alpha & \beta \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \quad (4)$$

The parameters α, β, γ, and δ are derived from h_{11}, h_{12}, h_{21}, and h_{22}, respectively, through subpixel interpolation. The AV1 codec utilizes 192 predefined 8-tap Finite Impulse Response filters with $1/64^{th}$-pel precision to refine the motion estimation. Warped Motion Compensation is applied to blocks of 8×8 pixels, with the final prediction constructed from individual sub-blocks.

The affine matrix for the LWMC is built by selecting candidate blocks that meet specific criteria: overlapping neighbors, a minimum size of 8×8 pixels, and warped motion mode enabled. A motion vector is then chosen from reference lists, typically the nearest or newly estimated ones [10].

Next, affine parameters are estimated by collecting motion vectors from neighboring blocks and filtering outliers. If insufficient valid samples remain, the candidate is discarded. Valid parameters are estimated, and candidates exceeding thresholds are rejected.

For valid candidates, motion compensation generates the predicted block. Luminance and chrominance components are processed separately – larger blocks use direct warped prediction, while smaller ones require specialized chroma compensation. With multiple reference frames, weighted blending combines predictions. The final transformation uses subpixel interpolation filters to refine accuracy.

Finally, the rate-distortion cost determines if LWMC improves compression. If beneficial, it is selected as the motion mode, and the parameters are encoded.

3 RELATED WORKS

Few works in the literature target the AV1 Warped Motion Compensation tool. The reference work [21] introduces two coding modes: Global and Locally Adaptive Warped Motion. Results show significant coding gains, especially for videos with strong non-translational motion, making this a key contribution to AV1's advanced motion compensation.

The work [7] presents two high-throughput, multiplierless hardware designs for the AV1 LWMC interpolation filters. This is the first work in the literature focusing on an AV1 LWMC specialized solution.

The works [17, 18] focus on the AV1 GWMC. The first paper investigates the impact of modifying the standard *libaom* GWMC implementation by replacing algorithms with others used in the field of computer vision. The second work proposes a machine learning-based approach to terminate the GWMC execution early to reduce computational costs.

To the best of our knowledge, no other works targeting AV1 Warped Motion Compensation have been published.

On the other hand, some works are targeting the VVC Affine Motion Estimation (AME), a tool with the same function as the AV1 LWMC. For example, the works [9, 24, 25] use machine learning models to reduce the VVC AME computational effort. However, since the encoding tools are different in VVC and AV1, the solutions targeting VVC AME are not relevant in this work, since the presented ideas cannot be applied to the AV1 LWMC machine learning models implemented in our work.

4 PROPOSED SOLUTION

The LWMC tool in AV1 competes with other inter-prediction methods like translational Motion Compensation and Overlapped Block Motion Compensation. The encoder evaluates all options and selects the one with the best Rate-Distortion trade-off. This work proposes a machine learning model to predict when LWMC will not be chosen, allowing it to be skipped. This reduces encoding time and power consumption while maintaining AV1 compatibility and minimizing impacts on coding efficiency.

A Decision Tree (DT) model was selected to minimize computational overhead due to its efficiency, hardware-friendly binary structure, and low processing cost – ideal for real-time encoding. A DT [5, 12] recursively partitions the feature space using binary splits on attributes, where each leaf node represents a class prediction. For our binary classification task, the tree predicts whether to skip LWMC execution.

To achieve enhanced results, the DT was trained using CatBoost [8, 22], a gradient boosting algorithm that natively handles categorical features, eliminating pre-processing needs while improving performance. The model was designed to be independent of video spatial and temporal resolutions, quantization levels, block sizes, and other encoder settings. Additionally, the model was tuned to use 100 iterations, with a max depth of 7, keeping the tree small and efficient.

Feature selection was conducted using CatBoost's importance scoring to identify and remove features. To ensure robustness, it was combined with cross-validation [23], training models on different dataset splits and selecting only the most consistently essential features for the final model. Over 600 features – listed in [14] – were evaluated from the motion mode selection data. After feature selection, 32 were retained for the final model. Due to space restrictions, the features are listed in [16].

The training dataset contained 36 video sequences across five resolutions, while the test dataset included 10 other sequences with the same resolution distribution. Both datasets were balanced by class, and all sequences were selected by resolution and content type according to [1] and sourced from the XIPH database [19].

To construct the datasets, the video sequences were encoded using a modified AV1 encoder (*libaom* v3.9.2) to extract features, processing each sequence at four Constrained Quality (CQ) levels (20, 32, 43, 55) in Random-Access mode. After model training, the model was implemented in a modified *libaom* encoder. This encoder processed a separate set of 15 sequences (Table 2), matching those used in the reference work [21].

It is important to emphasize that no video sequence was reused across datasets to prevent data leakage. The sequences were randomly selected while ensuring a balanced resolution and content type distribution, as categorized in [1]. A comprehensive list of sequences in each dataset can be found in [15].

Figure 3 illustrates our method integrated into AV1's motion mode decision flow. The process begins with the "Handle Inter Mode" function, which evaluates the Translational Motion Compensation (MC), Overlapped Block Motion Compensation (OBMC), and Local Warped Motion Compensation (LWMC) modes to select the option with the lowest Rate-Distortion (RD) cost. These modes compete with each other and, although they are executed sequentially in the current *libaom* implementation, they could theoretically be processed in parallel.

Subsequently, the encoder may perform additional procedures – such as transform search [11] and further RD-cost evaluations – before updating internal statistics with the best RD result obtained up to that point. This motion mode search is repeated for each available reference motion vector associated with the current block. Finally, the motion mode with the best RD performance is selected and transmitted in the bitstream.

The key innovation is the orange 'Skip?' decision step, where our machine learning model predicts whether to execute LWMC. By activating LWMC only when beneficial, this approach reduces computation while preserving coding efficiency. The prediction is based

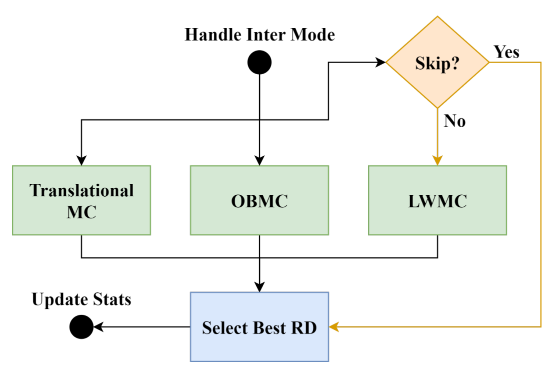

Figure 3: Proposed Method Diagram.

on the estimated likelihood of LWMC improving RD cost, allowing selective bypass of unnecessary warped motion compensation.

Table 1 presents the results using standard machine-learning metrics, where `False` indicates a *no-skip* decision and `True` represents a *skip* decision. Correct `False` predictions prevent discarding effective predictors, maintaining coding efficiency, while accurate `True` predictions enable encoding time savings. Mispredictions in either class may compromise coding efficiency or fail to achieve computational gains.

The results shown in Table 1 indicate that the model performs consistently across both classes, with the same F1-Score (0.85), suggesting a balanced trade-off between precision and recall. The `False` class achieves a slightly higher recall, while the `True` class has a marginally better precision, implying that the model is more effective at correctly identifying negative instances. The overall accuracy is 0.85. These results were considered satisfactory, and then a complete evaluation in a real scenario was done, and the results are presented in the next section.

5 RESULTS

The proposed method was implemented in AV1's reference software (*libaom* v3.9.2) and tested on 30 frames from 15 XIPH database sequences [19] that were excluded from the training and testing datasets. Encodings used Random-Access configuration at four CQ levels (20, 32, 43, 55), with all CPU cores fixed at 100% utilization to ensure consistent timing measurements.

To better determine the limits of gains and losses of our solution, an ablation experiment was conducted to investigate the relevance of the LWMC in terms of coding efficiency. For this, the LWMC was completely disabled, and the coding efficiency was measured. Effectively, the skip decision in Figure 3 was consistently enforced, meaning a 100% time reduction.

Table 2 shows coding efficiency (BD-BR) for each sequence, considering both experiments. BD-BR [4] measures bitrate variation at equivalent quality versus an anchor encoder, where negative values indicate improved efficiency.

Encoding time gains are also presented in Table 2. In this case, the results demonstrate the gain percentage when comparing our LWMC implementation to the original AV1 version. Negative values show faster LWMC execution.

The ablation experiment showed a 3.41% average BD-BR increase, demonstrating the content-dependent behavior of LWMC, with significant BD-BR variation, particularly in high-motion sequences like *Station 2* and *Blue Sky*. In contrast, low-motion sequences like *BQ Square* and *Party Scene* exhibited minimal coding efficiency degradation. Then, the standard deviation (SD) of LWMC coding efficiency is high, as presented in Table 2.

The results of our method, presented in Table 2, showed an average BD-BR penalty of 0.21%, outperforming the complete deactivation of LWMC by 16 times. Besides, our method also reached a lower standard deviation. Surprisingly, sequences *BQ Square* and *Party Scene* presented small coding efficiency gains, which were not expected.

Table 1: Model Classification Report

Class	Precision	Recall	F1-Score	Accuracy
False	0.84	0.87	0.85	0.85
True	0.86	0.83	0.85	

Table 2: Results of the Proposed Method

Group	Sequence	BD-BR (%)		LWMC (%)
		Ablation	Ours	
lowres	BQ Square	+0.02	-0.06	-75.87
	Flowervase	+2.71	+0.66	-63.15
	Tempete	+0.75	+0.17	-59.71
	Waterfall	+0.51	+0.02	-61.58
	Blue Sky 360p	+6.67	+0.07	-67.34
midres	Aspen	+1.01	+0.08	-38.19
	Into Tree 480p	+2.10	+0.09	-30.23
	Party Scene	+0.18	-0.01	-50.39
	Station 2 480p	+7.66	+0.24	-34.19
highres	Driving POV	+1.51	+0.49	-46.21
	Into Tree 720p	+3.21	+0.38	-49.77
	Into Tree 1080p	+2.32	+0.32	-48.93
	Mobcal	+2.70	+0.34	-68.21
	Blue Sky 1080p	+4.05	+0.14	-43.94
	Station 2 1080p	+15.77	+0.72	-44.05
AVG		**+3.41**	**+0.21**	**-52.12**
SD		**4.07**	**0.23**	**13.37**

When considering encoding time, our method reduced the average LWMC execution time by more than 52%, with consistent results across the evaluated video sequences. These values already consider the overhead of the developed method. The execution cost of our method was measured and was approximately 11% of the LWMC execution.

The encoding time and coding efficiency results demonstrate the effectiveness of the proposed method. To the best of the authors' knowledge, no prior work in the literature has specifically addressed the reduction of computational effort for the AV1 LWMC, as discussed earlier. Most related works, such as [9, 24, 25], focus on optimizing the affine motion compensation tool in the VVC standard. Then, a fair comparison is impossible.

6 CONCLUSION

This work presented a machine learning-based method to reduce the AV1 Local Warped Motion Compensation (LWMC) computational effort while preserving coding efficiency. Using a Decision Tree model, a 52% LWMC time reduction was achieved with only 0.21% BD-BR increase, significantly outperforming complete LWMC deactivation (3.41% BD-BR penalty). Our adaptive approach maintains AV1 compliance and shows particular effectiveness in complex-motion sequences. To our knowledge, this work represents the first machine learning-based optimization specifically targeting the AV1 LWMC tool.

ACKNOWLEDGMENTS

The authors of this work would like to thank the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES), Finance Code 001. and CNPq for funding this research.

REFERENCES

[1] Codec Working Group Alliance for Open Media. 2021. *AOM Common Test Conditions v2.0.* https://aomedia.org/docs/CWG-B075o_AV2_CTC_v2.pdf
[2] AOM. 2019. *AV1 Video Codec.* https://aomedia.org/av1
[3] Isis Bender, Alex Borges, Luciano Agostini, Bruno Zatt, Guilherme Correa, and Marcelo Porto. 2023. Complexity and compression efficiency analysis of libaom AV1 video codec. *J. Real-Time Image Process.* 20, 3 (may 2023), 14 pages. https://doi.org/10.1007/s11554-023-01308-5
[4] Gisle Bjontegaard. 2008. Improvements of the BD-PSNR model. ITU-T SC16/Q6. In *35th VCEG Meeting, Berlin, Germany, Doc. VCEG-AI11.*
[5] L. Breiman, J. Friedman, C.J. Stone, and R.A. Olshen. 1984. *Classification and Regression Trees.* Taylor & Francis. https://books.google.com.br/books?id=JwQx-WOmSyQC
[6] Yue Chen, Debargha Mukherjee, Jingning Han, Adrian Grange, Yaowu Xu, Zoe Liu, Sarah Parker, Cheng Chen, Hui Su, Urvang Joshi, Ching-Han Chiang, Yunqing Wang, Paul Wilkins, Jim Bankoski, Luc Trudeau, Nathan Egge, Jean-Marc Valin, Thomas Davies, Steinar Midtskogen, and Peter Rivaz. 2018. An Overview of Core Coding Tools in the AV1 Video Codec. 41–45. https://doi.org/10.1109/PCS.2018.8456249
[7] Robson Domanski, William Kolodziejski, Wagner Penny, Marcelo Porto, Bruno Zatt, and Luciano Agostini. 2023. High-Throughput and Multiplierless Hardware Design for the AV1 Local Warped MC Interpolation. In *2023 IEEE International Conference on Image Processing (ICIP).* 2680–2684. https://doi.org/10.1109/ICIP49359.2023.10222657
[8] Anna Veronika Dorogush, Vasily Ershov, and Andrey Gulin. 2018. CatBoost: gradient boosting with categorical features support. arXiv:1810.11363 [cs.LG] https://arxiv.org/abs/1810.11363
[9] Adson Duarte, Paulo Gonçalves, Luciano Agostini, Bruno Zatt, Guilherme Correa, Marcelo Porto, and Daniel Palomino. 2022. Fast Affine Motion Estimation for VVC using Machine-Learning-Based Early Search Termination. In *2022 IEEE International Symposium on Circuits and Systems (ISCAS).* 1–5. https://doi.org/10.1109/ISCAS48785.2022.9937973
[10] Alliance for Open Media. 2022. *Local Warped Motion Compensation.* https://github.com/AmusementClub/SVT-AV1/blob/mod/Docs/Appendix-Local-Warped-Motion.md
[11] Alliance for Open Media. 2022. *Transform Type and Transform Size Search.* https://gitlab.com/AOMediaCodec/SVT-AV1/-/blob/master/Docs/Appendix-TX-Search.md
[12] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. 2000. Special Invited Paper. Additive Logistic Regression: A Statistical View of Boosting. *The Annals of Statistics* 28, 2 (2000), 337–374. http://www.jstor.org/stable/2674028
[13] Jingning Han, Bohan Li, Debargha Mukherjee, Ching-Han Chiang, Adrian Grange, Cheng Chen, Hui Su, Sarah Parker, Sai Deng, Urvang Joshi, Yue Chen, Yunqing Wang, Paul Wilkins, Yaowu Xu, and James Bankoski. 2021. A Technical Overview of AV1. *Proc. IEEE* 109, 9 (2021), 1435–1462. https://doi.org/10.1109/JPROC.2021.3058584
[14] W. Kolodziejski. 2025. *List of Features.* https://github.com/WKolodziejski/WebMedia-2025/blob/main/features.txt
[15] W. Kolodziejski. 2025. *List of Sequences.* https://github.com/WKolodziejski/WebMedia-2025/blob/main/sequences.txt
[16] W. Kolodziejski. 2025. *List of Used Features.* https://github.com/WKolodziejski/WebMedia-2025/blob/main/model.txt
[17] William Kolodziejski, Robson Domanski, and Luciano Agostini. 2024. Speeding Up the AV1 Global Warped Motion Compensation. In *2024 IEEE 15th Latin America Symposium on Circuits and Systems (LASCAS).* 1–5. https://doi.org/10.1109/LASCAS60203.2024.10506160
[18] William Kolodziejski, Robson Domanski, and Luciano Agostini. 2025. FastGW: A Machine Learning-Based Early Skip for the AV1 Global Warped Motion Compensation. *IEEE Transactions on Circuits and Systems I: Regular Papers* 72, 3 (2025), 977–988. https://doi.org/10.1109/TCSI.2024.3486243
[19] Xiph.org Test Media. 2004. *Collection of test sequences and clips for evaluating compression technology.* https://media.xiph.org/video
[20] Tung Nguyen and Detlev Marpe. 2018. Future Video Coding Technologies: A Performance Evaluation of AV1, JEM, VP9, and HM. In *2018 Picture Coding Symposium (PCS).* 31–35. https://doi.org/10.1109/PCS.2018.8456289
[21] Sarah Parker, Yue Chen, David Barker, Peter Rivaz, and Debargha Mukherjee. 2017. Global and locally adaptive warped motion compensation in video compression. 275–279. https://doi.org/10.1109/ICIP.2017.8296286
[22] Liudmila Prokhorenkova, Gleb Gusev, Aleksandr Vorobev, Anna Veronika Dorogush, and Andrey Gulin. 2019. CatBoost: unbiased boosting with categorical features. arXiv:1706.09516 [cs.LG] https://arxiv.org/abs/1706.09516
[23] Payam Refaeilzadeh, Lei Tang, and Huan Liu. 2009. *Cross-Validation.* Springer US, Boston, MA, 532–538. https://doi.org/10.1007/978-0-387-39940-9_565
[24] Fernando Sagrilo, Marta Loose, Ramiro Viana, Gustavo Sanchez, Guilherme Corrêa, and Luciano Agostini. 2023. Learning-Based Fast VVC Affine Motion Estimation. In *2023 IEEE International Symposium on Circuits and Systems (ISCAS).* 1–5. https://doi.org/10.1109/ISCAS46773.2023.10181659
[25] Ramiro Viana, Marta Loose, Rafael Ferreira, Marcelo Porto, Guilherme Corrêa, and Luciano Agostini. 2024. A Hardware-Friendly Acceleration of VVC Affine Motion Estimation Using Decision Trees. In *2024 37th SBC/SBMicro/IEEE Symposium on Integrated Circuits and Systems Design (SBCCI).* 1–5. https://doi.org/10.1109/SBCCI62366.2024.10703987
[26] WebM. 2013. *VP9 Video Codec.* https://www.webmproject.org/vp9