

Demo: Real-Time Deep Video Watermarking on the Edge

Kosta Pavlovic
kosta@ucg.ac.me
Computer Science Center, Faculty of
Natural Sciences and Mathematics,
University of Montenegro
Podgorica, Montenegro

Roberto Cavicchioli
roberto.cavicchioli@unimore.it
Department of Communication and
Economics, University of Modena and
Reggio Emilia
Modena, Italy

Alessandro Capotondi
alessandro.capotondi@unimore.it
Department of Physics, Informatics
and Mathematics, University of
Modena and Reggio Emilia
Modena, Italy

Abstract

This demo presents a deployment-oriented system for real-time deep video watermarking on edge hardware. It addresses a central practical limitation of recent learning-based watermarking methods which offer strong robustness and high visual quality, but are often too computationally demanding for live execution on embedded platforms. Starting from a state-of-the-art baseline, we obtain a compact embedder suitable for edge deployment through architecture simplification, progressive distillation, and hardware-in-the-loop optimization. The live demonstration runs on NVIDIA Jetson AGX Orin and shows an end-to-end pipeline comprising live camera capture, on-device watermark embedding, real-time display, video recording, and post-hoc watermark recovery from the recorded stream. In the implemented setup, the full demonstration pipeline operates at approximately 30 FPS on 720p input which illustrates the feasibility of real-time video watermarking on the edge for trustworthy media processing and content attribution.

CCS Concepts

• **Applied computing** → **Computer forensics**; • **Computing methodologies** → *Model development and analysis*; • **Security and privacy** → *Digital rights management*.

Keywords

Video watermarking, neural network compression, progressive knowledge distillation, edge deployment.

ACM Reference Format:

Kosta Pavlovic, Roberto Cavicchioli, and Alessandro Capotondi. 2026. Demo: Real-Time Deep Video Watermarking on the Edge. In *The 24th Annual International Conference on Mobile Systems, Applications and Services (MobiSys Workshop '26)*, June 21–25, 2026, Cambridge, United Kingdom. ACM, New York, NY, USA, 2 pages. <https://doi.org/10.1145/3812836.3814795>

1 Introduction

Digital watermarking is becoming increasingly important as a mechanism for media provenance, attribution, forensic tracking and trustworthy content processing. Deep-learning-based watermarking can learn embedding strategies that better balance watermark imperceptibility and robustness to realistic distortions.



This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License.

MobiSys Workshop '26, Cambridge, United Kingdom

© 2026 Copyright held by the owner/author(s).

ACM ISBN 979-8-4007-2712-2/2026/06

<https://doi.org/10.1145/3812836.3814795>

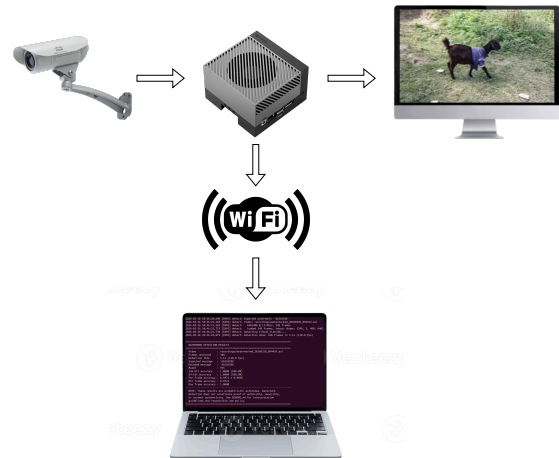


Figure 1: End-to-end demo workflow: live video capture, on-device watermarking, display, and remote verification.

However, these benefits usually come at the cost of substantially higher computational complexity, which limits practical deployment in real-time and edge scenarios where watermarking must operate during capture, generation, or transmission of video streams. This makes model compression a necessary step toward making deep watermarking feasible on embedded and resource-constrained platforms while preserving the robustness and visual quality that make deep approaches attractive.

2 Demo Overview

The goal of this demo is to show that deep-learning-based video watermarking can be integrated into a practical edge pipeline rather than remaining only an offline research result. The demonstrated workflow is presented in Figure 1.

The system consists of two logical parts: an online embedding pipeline and an offline detection pipeline. The online pipeline captures live video from a USB camera on the NVIDIA Jetson AGX Orin, resizes each frame, converts it to the appropriate color space and tensor representation, and runs the optimized embedder on-device. The predicted watermark pattern is resampled to the original resolution and blended with the original frame using a predefined embedding strength. The resulting watermarked frame is then converted back to display format, shown on an external display, and recorded to disk.

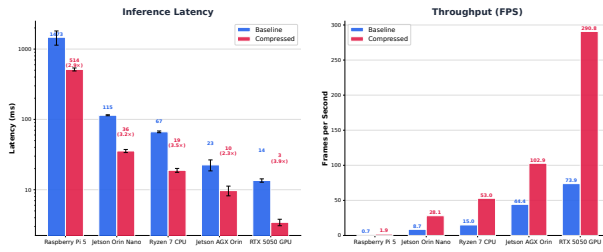


Figure 2: Inference latency and throughput of the baseline and compressed embedder across the evaluated platforms.

The full live pipeline runs at approximately 30 FPS on 720p input. For presentation clarity, the payload is represented as a 64-bit message corresponding to an 8-character ASCII string. The 64-bit payload is then encoded using error-correcting codes (ECC) to form the 256-bit watermark used by the model.

After recording is completed, the saved video is processed by the offline detection pipeline, running on a PC. The detector loads the video, resizes the frames for evaluation, converts them into the detector input representation, and runs inference in chunks. It then computes frame-level watermark estimates and aggregates the information across the video sequence. The resulting encoded watermark is decoded using the ECC decoder to recover the original 64-bit message. Finally, the recovered message is converted back into text and presented together with summary decoding statistics.

3 Methodology

The main technical contribution behind the demonstration is a compressed and deployment-oriented watermark embedder derived from a state-of-the-art baseline model. We followed an analysis-driven redesign procedure guided by redundancy measurements on the baseline architecture. In particular, we examined redundancy at the *block* and *layer* level. At the block level, we examined the redundancy of the paired convolutions within residual blocks and found that this redundancy was consistently high. At the layer level, we analyzed redundancy among convolutional filters and found that it supported moderate channel-width reduction.

Based on these observations, the compressed embedder replaces the heavier building blocks with lighter alternatives while preserving the overall encoder-decoder structure and the original message-processing interface. The redesign was further combined with progressive distillation to stabilize training and preserve watermark recoverability. Hardware-in-the-loop benchmarking was used to guide low-level architectural choices, particularly in the resampling path, to improve deployment efficiency on representative edge platforms. The resulting compressed model reduces the embedder from 23.66M to 6.37M parameters and from 28.34G to 7.42G MACs/frame.

4 Benchmark and Results

The compressed model was evaluated against the baseline in terms of robustness, visual quality, and runtime on a held-out subset of SA-V dataset under photometric, geometric, video-compression, and temporal distortions.

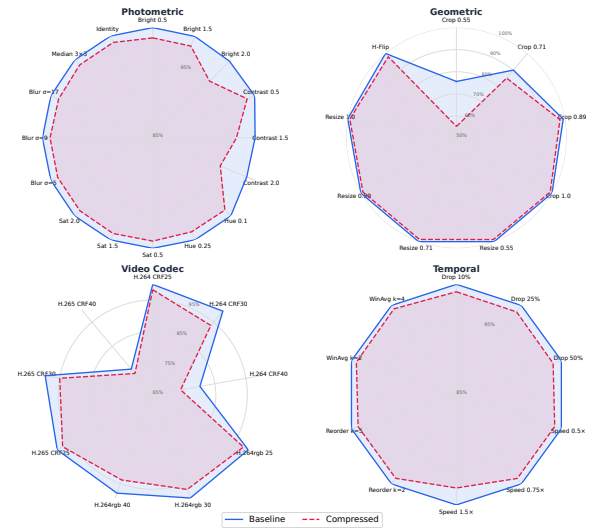


Figure 3: Robustness of the baseline and compressed embedder across distortion classes.

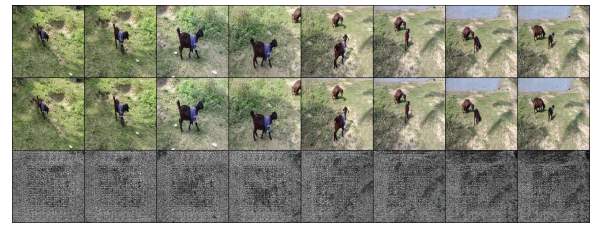


Figure 4: Visual-quality example: original, watermarked, and amplified-difference frames.

Although Jetson AGX Orin is the main target platform for the final demonstration, runtime was also measured on several representative platforms: Raspberry Pi 5, Jetson Orin Nano, an AMD Ryzen 7 CPU, and an RTX 5050 GPU. Figure 2 summarizes inference latency across the evaluated hardware. These latency results correspond to *model inference only* at 256×256 input resolution and exclude preprocessing and postprocessing outside the model, data movement, display, recording, or other application-level overheads.

The compressed model makes real-time edge execution practical while retaining 97.7% of the baseline robustness on average. It also preserves high visual quality, achieving 43.12 dB PSNR and 0.9928 SSIM, compared with 45.39 dB and 0.9971 SSIM for the baseline. Figures 3 and 4 summarize robustness and visual quality.

Acknowledgments

This work was supported by the dAIEDGE project, funded by the European Union under Grant Agreement No. 101120726, through the dAIEDGE Exchange Programmes Open Call 1, Sub-grant Agreement No. dAIIOC06. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Commission. Neither the European Union nor the granting authority can be held responsible for them.