Towards Secure Authentication: Detecting Replay Attacks via Compression Artifacts

Sahar Husseini, Jean-Luc Dugelay
Department of Digital Security
EURECOM
Sophia Antipolis, France
husseini, dugelay@eurecom.fr

***Abstract*—This study presents a novel machine learning-based approach to mitigating digital replay attacks in face verification systems by leveraging compression artifacts to differentiate between compressed and uncompressed video frames. While traditional methods rely on active liveness detection, which can be inconvenient and negatively affect user experience, this work addresses the lack of automated solutions for detecting replay attacks. Using raw, uncompressed video datasets and widely-used video compression algorithms, the proposed method trains a classifier to identify compression artifacts as distinguishing features. Experimental results validate the model's effectiveness in detecting injected content, highlighting the critical role of compression artifacts in enhancing the robustness of video authentication systems. This contribution represents a significant step toward advancing anti-spoofing techniques by exploring a previously underutilized aspect of video integrity.**

***Index Terms*—digital replay attacks, video injection, video compression, codecs**

I. Introduction

Advancements in biometric techniques and the increasing demand for seamless identity verification have led to the widespread adoption of remote face verification systems. As these systems become more popular, they also become increasingly attractive targets for malicious actors attempting to spoof the system and impersonate legitimate users. These attacks can occur on both the client side, where users interact with the system, and the server side, where data is processed and validated. In this paper, we assume that both the application running on the device and the server handling the data are securely protected. Our focus is on addressing client-side vulnerabilities, particularly injection attacks, where attackers bypass the sensor and directly inject malicious digital content into the data stream [2]. Injection attacks can be broadly classified into deepfake attacks and digital replay attacks:

1) **Deepfake Attacks**: In this type of attack, the adversary begins by obtaining images of the victim. Using a deepfake algorithm, the attacker generates a manipulated video in real time, replicating the facial expressions and head movements required by an active liveness detection system [25]. This deepfake video is then streamed to the authentication system via virtual camera software (e.g., OBS), which acts as an intermediary to bypass the physical camera sensor. Figure 1 illustrates the process of a deepfake injection attack.
2) **Digital Replay Attacks**: In these attacks, adversaries leverage authentic video footage of the victim, often sourced from publicly available platforms such as social media, and inject it into the system using virtual camera software. Since the video is genuine and lacks manipulation artifacts, it poses significant challenges for the system to distinguish between a live user and a replayed video.

To mitigate deepfake attacks researchers have proposed various detection methods, ranging from early handcrafted feature-based techniques to modern deep learning-based approaches [5], [9], [11], [12], [25]. These detection methods aim to identify subtle inconsistencies or artifacts introduced during video synthesis and they often rely on supervised learning, where models are trained to recognize known deepfake artifacts.

To address digital replay attacks, remote face authentication systems commonly implement active liveness detection, which requires users to perform specific actions such as nodding, blinking, or opening their mouth to verify their physical presence and confirm they are alive. While this approach is effective in countering replayed videos, it is not always user-friendly and may be perceived as inconvenient, potentially impacting the overall usability of the system. Another method for detecting replay injection attacks involves analyzing the metadata of the user's device and camera, flagging suspicious camera names as potential indicators of an injection. However, this method is highly vulnerable, as attackers can easily manipulate the camera metadata to bypass detection algorithms. This type of attack is the focus of our investigation in this paper. To the best of our knowledge, no machine learning-based approaches currently exist to mitigate digital replay attacks, as the injected video is authentic and lacks detectable artifacts.

Content captured by cameras in practical scenarios often undergoes various digital image and video processing operations, including post-processing techniques such as stylization filters and beautification [18], before being disseminated. Recent studies have systematically evaluated the adverse effects of these operations on the performance of biometric algorithms [7], [13], [21]. Hence, in this work, we investigate whether providing uncompressed video access to face anti-spoofing service providers can improve the detection of injected versus authentic video streams. Building on this, we propose bypassing the compression step and directly captur-

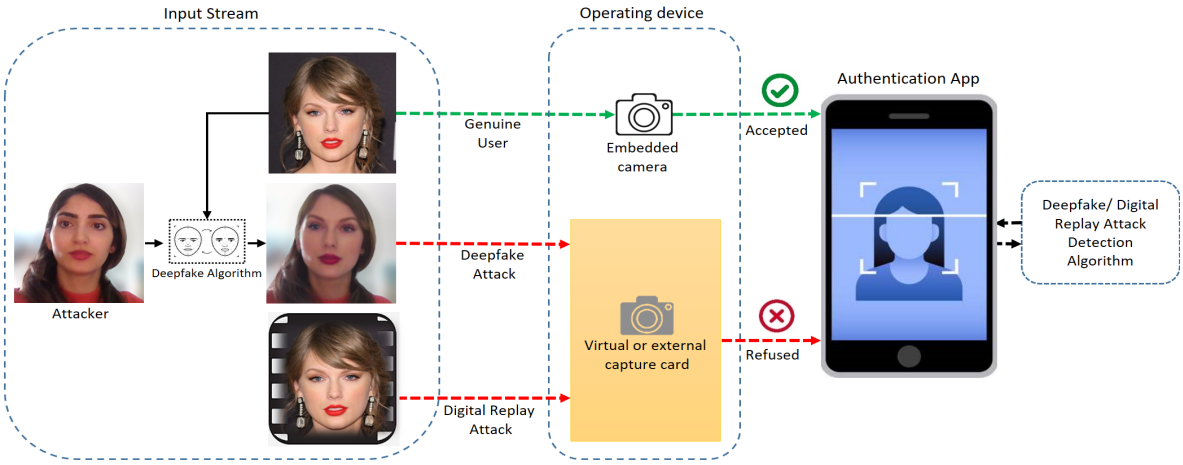

Fig. 1. **Illustration of various input streams to a remote face authentication service.** The input can originate from different scenarios. In the first scenario, the face of a genuine user is provided to the service, granting access to the application upon successful authentication. In the second scenario, a deepfake injection attack is performed. Here, the attacker generates a real-time video mimicking the victim's expressions and head movements using a single image. This video is streamed via virtual camera software to imitate a legitimate webcam feed, deceiving the authentication system. In the third scenario, the attacker uses either a single image or a pre-recorded video of the victim. The virtual camera streams a genuine video of the victim that lacks visible artifacts. Our main goal is to exploit compression artifacts for detecting digital replay attack.

ing uncompressed image data from the user's device during authentication, rather than relying on video content that has passed through the complete post-processing stages of the Image Signal Processing (ISP) pipeline [10]. This strategy enables the detection system to better differentiate between authentic and injected video streams, as injected videos are often sourced from the internet and are typically compressed using widely-used algorithms. By ensuring that the user's camera captures uncompressed images during authentication, the detection model can focus on identifying compression artifacts. The presence of such artifacts strongly indicates that the image frame has been injected, thereby significantly improving the system's ability to detect and prevent injection attacks.

For this purpose, we utilize raw video datasets in their original, uncompressed form to simulate real-world scenarios. To generate corresponding compressed versions, widely-used video compression algorithms are applied. A ResNet-50 classifier was then trained using both compressed and uncompressed frames to enable discrimination between the two. Through this process, we aim to evaluate the classifier's ability to detect compression artifacts and assess its effectiveness in accurately distinguishing between compressed and uncompressed frames. This analysis provides valuable insights into the role of compression artifacts as distinguishing features in video authentication tasks.

II. Related Work

A. Deepfakes Attack Detection

In response to the challenges posed by deepfakes, researchers have explored enhancing deepfake detectors from various perspectives. Some detection methods operate at the image level, identifying fake images by recognizing spatial artifacts within individual frames [4], [18], while others focus on the video level, leveraging temporal information by analyzing multiple frames to detect deepfake videos [8], [20]. Furthermore, certain detection methods utilize frequency information, which proves particularly effective on highly compressed videos. The LRL [3] and FRDM [22] combine representations from both RGB and frequency domains to learn inconsistencies in the video frames. Another direction in deepfake detection involves the use of training data synthesis, generating synthetic data that includes common deepfake artifacts. These techniques do not rely on existing fake data but generate their own. For instance, DSPFWA [16] focuses on identifying artifacts that arise during face warping, where a source face undergoes transformations such as scaling, rotation, and shearing to match the pose of target face it aims to replace. These transformations create artifacts and resolution inconsistencies between the warped face area and the surrounding context. During training, the algorithm generates synthetic data containing these affine face warping artifacts to improve detection accuracy.

While many current methods perform well in detecting known manipulations, certain studies [5], [27] have identified limitations in their ability to generalize to fake faces forged by unknown manipulations. This is often due to overfitting to method-specific artifacts observed during training.

B. Digital Replay Attack Detection

Digital replay attacks involve injecting a genuine video stream of the victim into a facial recognition system, often using webcam simulation tools like OBS Studio on computers. On mobile devices, more sophisticated software solutions are required to achieve similar results [1], [2]. Detecting these attacks is challenging, as the biometric data used is authentic and lacks detectable anomalies. As a result, defense strategies against digital replay attacks remain underexplored,

with limited research addressing this issue within the biometric community. Some studies have proposed methods to counter digital replay attacks by leveraging external signals to verify the presence of a live user in front of the camera. The authors of [28] and [6] suggest using a smartphone screen to emit randomly flashing colors onto the user's face. These flashes serve as a dynamic watermark within the video sequence, as the light is reflected off the user's face and captured by the camera. By analyzing the reflected light colors, these methods aim to distinguish between a live face and a replayed video or image. Similarly, [32] and [19] propose utilizing specific light patterns as an authentication mechanism for the captured content. However, the effectiveness of these approaches is limited due to their reliance on external signals, which are often too weak to detect, especially under strong ambient lighting conditions or on individuals with low skin reflectance. These constraints highlight the need for more robust and practical solutions to combat digital replay attacks.

C. Compression Detection in Digital Video Forensics

A significant area of research in digital video forensics involves the detection and analysis of compression artifacts, which provide valuable insights into the editing history of videos. These artifacts are typically examined through spatial statistics within individual frames and temporal statistics embedded in the Group of Pictures (GOP) structure. The GOP defines the types and sequence of frames in a video, establishing the foundation for compression analysis.

Video manipulation often involves decompression, editing, and recompression, making the detection of double compression artifacts particularly important. These artifacts serve as crucial evidence for identifying the sequence of edits and determining the presence of tampering in the video. Techniques such as the analysis of quantization artifacts or blockiness patterns have been developed to detect traces of recompression in both images and videos. For example, the authors of [24] propose a Support Vector Machine (SVM)-based classifier to determine the number of compression steps applied to a video sequence. Their method relies on Benford's law, analyzing the statistics of the most significant digit in quantized transform coefficients. Similarly, Jiang et al. [14] apply Markov statistics to identify double quantization artifacts in MPEG-4 videos. Other studies, such as [15] and [34], focus on periodicity analysis and the GOP structure to detect double compression in videos.

Despite the significant body of research on detecting double or multiple compressions, most existing methods are limited to identifying double-compressed videos for specific codecs or compression parameters. Moreover, these methods typically rely on processed RGB image or video data. However, in authentication systems where the data is directly captured by the service provider, the input data can be more tightly controlled.

Both replay and deepfake injection attacks remain unresolved challenges. The ideal solution to this challenge is to cryptographically sign biometric data at the hardware level, enabling hardware manufacturers to verify the authenticity of the captured content [37]. However, implementing this approach requires seamless collaboration among hardware manufacturers, operating system developers, software providers, and face anti-spoofing service providers, a level of coordination that has not yet been achieved.

III. Proposed Method

This work investigates whether providing uncompressed video access to face anti-spoofing service providers can improve the detection of injected versus authentic video streams. We hypothesize that uncompressed video frames from a user's device would lack compression artifacts, while injected videos, such as deepfakes or replays, would show detectable artifacts due to compression during their creation or transmission. The aim of this study is to analyze these compression artifacts for effective differentiation.

To achieve this, we propose a machine learning-based model trained on both compressed and uncompressed video frames. Videos are compressed using four widely used algorithms—H.264, H.265, VP8, and VP9—chosen for their popularity. Both compressed and uncompressed video versions are converted into individual frames.

For training, random patches of size 224×224 are extracted from the frames, ensuring balanced representation from both compressed and uncompressed frames. By focusing on image patches rather than full frames, the model captures localized compression artifacts, which are key for accurate detection. These patches are then used to train a binary classifier. The model is optimized using cross-entropy loss, L, defined as:

$$L = -\frac{1}{N} \sum_{i=0}^{N-1} \{t_i \log F(x_i) + (1 - t_i) \log(1 - F(x_i))\}$$

where $F(x)$ represents the probability of classifying a patch x as compressed, and t_i is the binary label associated with the input patch, where $t_i = 1$ for compressed and $t_i = 0$ for uncompressed.

IV. Experiment

A. Experimental Setup

To train and test a classifier for compression detection, we utilize six well-established video datasets that are commonly employed in video quality assessment and coding algorithm evaluation.

- **Xiph.org Video Test Media Dataset** [36]: contains a diverse collection of video clips with varying resolutions (240 to 2160), frame rates (25–60 fps). A subset of 47 videos, featuring resolutions of CIF (352×288), HD (1280×720), and Full HD (1920×1080), is selected from this dataset.
- **SJTU-4K Video Sequence Dataset** [30]: contains 15 4K (3840×2160) sequences captured with a Sony F65 camera at 30 fps. For our experiments, we utilize the 8-bit YUV 4:2:0 format videos.

TABLE I
MODEL PERFORMANCE ON TRAINED (H.264, H.265, VP8, VP9) AND UNSEEN (MPEG-4) CODECS ON THE MCL-JCV CROSS-TEST DATASET.

Codec	AUC	Precision	Recall	F1
H.264	0.979	0.868	0.970	0.916
H.265	0.986	0.870	0.986	0.924
VP8	**0.993**	**0.871**	**0.998**	**0.930**
VP9	0.988	0.869	0.982	0.922
Mpeg4	0.965	0.864	0.939	0.900

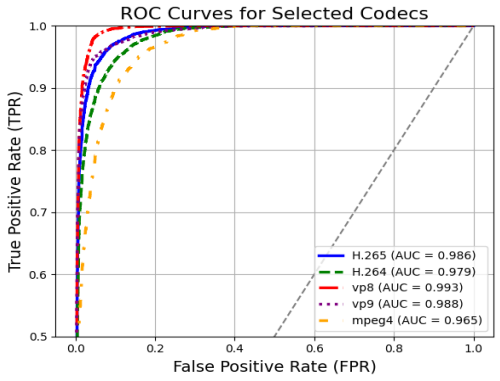

Fig. 2. ROC curves for different codecs

- **SJTU-HDR Video Sequence Dataset** [29]: contains 16 High Dynamic Range video sequences, captured at 60 fps using Sony F65 and F55 cameras. The sequences, originally provided in 16-bit OpenEXR format. We convert videos to 8-bit YUV 4:2:0 for our experiments.
- **UVG-Dataset** [23]: comprises 16 4K video sequences captured at 50 or 120 fps in raw 8-bit and 10-bit YUV 4:2:0 formats. We utilize the 4:2:0 YUV format for this study.
- **USTC-TD Video Dataset** [17]: This dataset contains 10 video sequences, captured at 30 fps using Nikon D3200 and Nikon Z-fc cameras. The videos are provided in Full HD and were converted to the YUV 4:2:0 format using FFmpeg library [33].
- **MCL-JCV** [35]: Comprises 30 HD/Full HD uncompressed video sequences. Additionally, it includes encoded videos produced using the H.264/AVC codec, with their quality determined by the quantization parameter (QP), which varies from 1 to 51. For this study we utilize HD videos.

We use the videos from the first five datasets for training. The data is split into 70% for training, 20% for evaluation, and 10% for testing. To assess the model's ability to generalize to datasets beyond the training set, we use the **MCL-JCV** [35] dataset as a cross-test set. We leverage both compressed and uncompressed video frames to train our classifier. Uncompressed frames are generated by converting raw videos into .png format without any compression. Compressed frames are obtained by applying compression algorithms (H.264, H.265, VP8, and VP9) to the raw videos, followed by saving the resulting frames in .png format. The default compression parameters of FFmpeg for each algorithm are used to simulate typical compression scenarios. We ensure an equal number of uncompressed and compressed frames for training. To introduce greater variability during training, we randomly crop image patches of size 224x224 and apply both horizontal and vertical flipping as augmentations. The ResNet-50 architecture, pretrained on ImageNet, is employed and fine-tuned for 20 epochs using the Adam optimizer. A batch size of 128 is used, with an initial learning rate of 0.001, decayed by a factor of 0.1 every 10 epochs to ensure stable convergence. ResNet-50 is chosen for its deep residual structure, which enables effective learning of subtle compression artifacts present in the data.

B. Experimental Results

In our experiment, the MCL-JCV dataset is employed as a cross-test dataset to evaluate the model's performance on previously unseen data. Table I highlights the model's performance on the codecs it was trained on, using the default parameters of the FFmpeg library. Furthermore, the table evaluates the model's generalization capability by testing its performance on an unseen codec, specifically the MPEG-4 compression method.

The results in Table I and AUC curves in Figure 2 reveal outstanding performance for the H.265, VP8, and VP9 codecs, with AUC values near or equal to 0.99. Among these, the VP8 codec achieves the best results across all metrics, suggesting that its compression artifacts are the most distinguishable by the detector. In contrast, the H.264 codec demonstrates slightly lower performance across all metrics, indicating that its compression artifacts are less prominent and harder for the model to detect. For the unseen MPEG-4 codec, the results show a decline in performance compared to the seen codecs. Nevertheless, the model maintains a reasonably high level of accuracy, showcasing its adaptability to compression methods it was not explicitly trained on.

Table II and Figure 3 showcase the model's capacity to generalize to compressed frames across a range of quantization parameters, despite being trained with FFmpeg's default quantization values. The H.264 codec is tested with Quantization Parameters (QP) ranging from 1 to 50. A QP value of 1 represents the highest image quality, while 50 corresponds to the lowest. The results in Table II reveal that at QP=1, the model struggles to distinguish between compressed and uncompressed frames, as the quality is nearly indistinguishable from uncompressed images. From QP=20 onward, the model's performance improves significantly, with metrics approaching near-perfect values. This trend indicates that higher compression levels introduce more noticeable artifacts, making them easier for the model to detect.

To gain qualitative insights, Figure 4 visualizes the results of guided backpropagation [31], which highlights all contributing features that influence the prediction. Additionally, we use Grad-CAM [26] to visualize the regions where the model's

TABLE II
PERFORMANCE METRICS OF THE MODEL FOR H.264 COMPRESSION ACROSS VARYING QUANTIZATION PARAMETERS (QP) ON THE MCL-JCV CROSS-TEST DATASET.

Codec-QP	AUC	Precision	Recall	F1
H.264-Q01	0.498	0.492	0.143	0.221
H.264-Q20	0.948	0.859	0.903	0.881
H.264-Q30	0.984	0.869	0.981	0.922
H.264-Q45	0.994	0.871	0.997	0.930
H.264-Q50	**0.995**	**0.871**	**0.999**	**0.931**

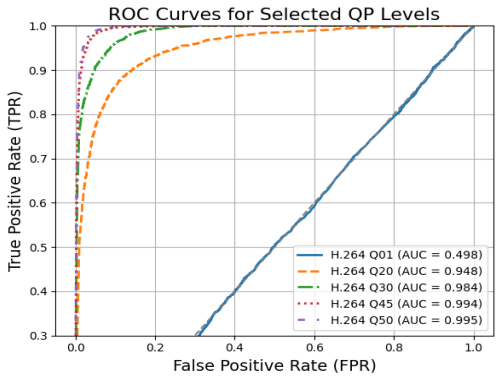

Fig. 3. ROC curves for H.264 codec with different QP levels

attention is concentrated, specifically for the compressed category. As seen in the visualizations, the model's attention is more sparse when analyzing uncompressed frames, whereas on compressed frames (e.i. H.264, H.265, VP8, VP9 and Mpeg4) the attention is more concentrated and coarser, focusing on areas where compression artifacts are most prominent.

Based on the quantitative and qualitative results, we can conclude that all compression methods introduce artifacts that are distinguishable from uncompressed frames. Notably, even under moderate compression with a quantization parameter (QP) of 20—where image quality remains nearly flawless—the model reliably detects compression artifacts.

If the authentication service can access uncompressed frames directly from the device, it can focus exclusively on identifying compression artifacts. The presence of such artifacts would indicate that the frame is injected, simplifying the detection process. This approach eliminates the need for the algorithm to recognize specific artifacts left behind by various deepfake generators. Furthermore, in digital replay attacks, where virtual cameras often apply video compression, injected videos should be distinguishable from genuine videos due to the compression artifacts they inevitably contain. This distinction significantly facilitates the task of detecting injected content.

V. CONCLUSION

This study introduces a novel approach for mitigating replay attacks by utilizing compression artifacts to differentiate between compressed and uncompressed video frames. These artifacts act as reliable indicators of injected content, thereby facilitating the detection process. In the context of digital replay attacks, where virtual cameras typically apply video compression, injected videos can be distinguished from genuine ones based on the inherent compression artifacts. By using raw video datasets and applying common video compression algorithms, a classifier was trained to differentiate between compressed and uncompressed frames. Experimental results demonstrate that the model effectively performs this distinction, highlighting the significance of compression artifacts in video authentication.

REFERENCES

[1] K. Carta, C. Barral, N. El Mrabet, and S. Mouille, "On the pitfalls of videoconferences for challenge-based face liveness detection," in *proceedings of World Multi-Conference on Systemics, Cybernetics and Informatics*, vol. 2021, 2021.

[2] ——, "Video injection attacks on remote digital identity verification solution using face recognition," in *in 13th International Multi-Conference on Complexity, Informatics and Cybernetics*, 2022.

[3] S. Chen, T. Yao, Y. Chen, S. Ding, J. Li, and R. Ji, "Local relation learning for face forgery detection," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 35, no. 2, 2021, pp. 1081–1088.

[4] J. Choi, T. Kim, Y. Jeong, S. Baek, and J. Choi, "Exploiting style latent flows for generalizing deepfake detection video detection," *arXiv preprint arXiv:2403.06592*, 2024.

[5] S. Dong, J. Wang, R. Ji, J. Liang, H. Fan, and Z. Ge, "Implicit identity leakage: The stumbling block to improving deepfake detection generalization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 3994–4004.

[6] H. Farrukh, R. M. Aburas, S. Cao, and H. Wang, "Facerevelio: a face liveness detection system for smartphones with a single front camera," in *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking*, 2020, pp. 1–13.

[7] S. Gal and B. Bulgurcu, "Exploring factors influencing internet users' susceptibility to deepfake phishing," 2024.

[8] Q. Gu, S. Chen, T. Yao, Y. Chen, S. Ding, and R. Yi, "Exploiting fine-grained face forgery clues via progressive enhancement learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 1, 2022, pp. 735–743.

[9] A. Heidari, N. Jafari Navimipour, H. Dag, and M. Unal, "Deepfake detection using deep learning methods: A systematic and comprehensive review," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 14, no. 2, p. e1520, 2024.

[10] S. Husseini, P. Babahajiani, and M. Gabbouj, "Color constancy model optimization with small dataset via pruning of cnn filters," in *2021 9th European Workshop on Visual Information Processing (EUVIP)*. IEEE, 2021, pp. 1–6.

[11] S. Husseini and J.-L. Dugelay, "A comprehensive framework for evaluating deepfake generators: Dataset, metrics performance, and comparative analysis," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 372–381.

[12] ——, "Alignface: Enhancing face verification models through adaptive alignment of pose, expression, and illumination," in *2024 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2024, pp. 3243–3249.

[13] S. Husseinil and J.-L. Dugelay, "Raw data: A key component for effective deepfake detection," in *ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2025, pp. 1–5.

[14] X. Jiang, W. Wang, T. Sun, Y. Q. Shi, and S. Wang, "Detection of double compression in mpeg-4 videos based on markov statistics," *IEEE Signal processing letters*, vol. 20, no. 5, pp. 447–450, 2013.

[15] Y. Li, M. Gardella, Q. Bammey, T. Nikoukhah, J.-M. Morel, M. Colom, and R. G. von Gioi, "A contrario detection of h. 264 video double compression," in *2023 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2023, pp. 1765–1769.

[16] Y. Li and S. Lyu, "Exposing deepfake videos by detecting face warping artifacts."

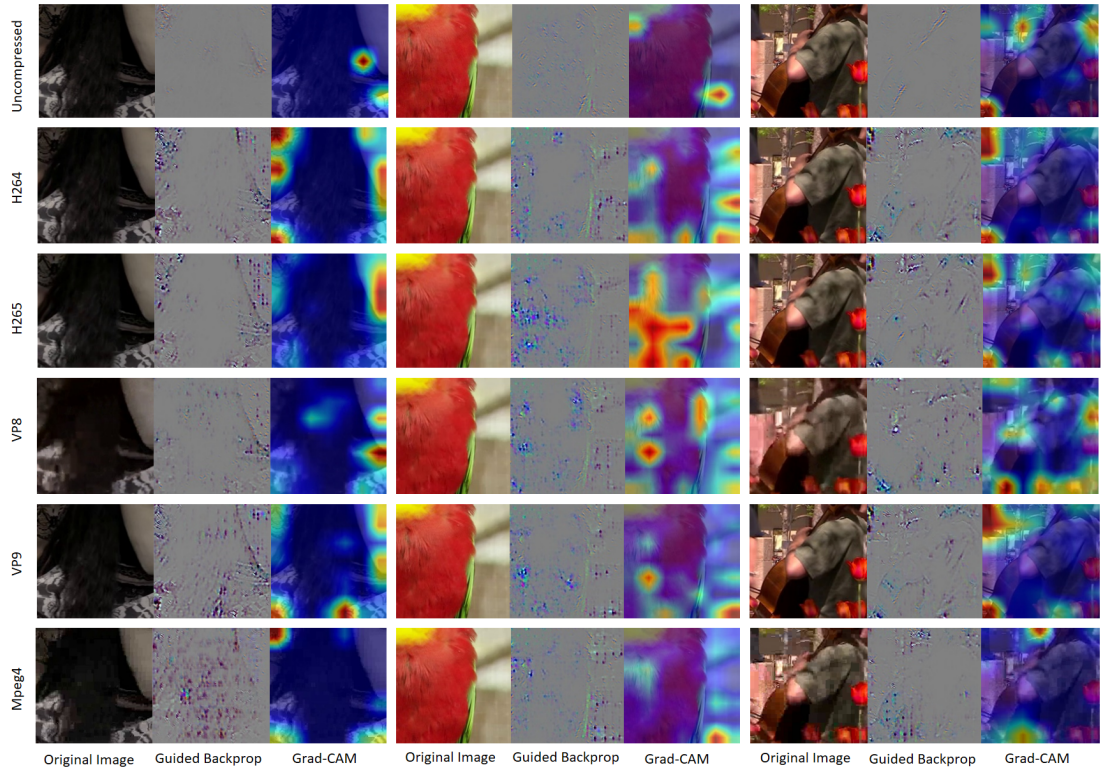

Fig. 4. Guided Backpropagation and Grad-CAM visualizations for uncompressed and compressed video frames, highlighting the areas of support for the compressed category.

[17] Z. Li, J. Liao, C. Tang, H. Zhang, Y. Li, Y. Bian, X. Sheng, X. Feng, Y. Li, C. Gao *et al.*, "Ustc-td: A test dataset and benchmark for image and video coding in 2020s," *arXiv preprint arXiv:2409.08481*, 2024.

[18] A. Libourel, S. Husseini, N. Mirabet-Herranz, and J.-L. Dugelay, "A case study on how beautification filters can fool deepfake detectors," in *IWBF 2024, 12th IEEE International Workshop on Biometrics and Forensics*, 2024.

[19] H. Liu, Z. Li, Y. Xie, R. Jiang, Y. Wang, X. Guo, and Y. Chen, "Livescreen: Video chat liveness detection leveraging skin reflection," in *IEEE INFOCOM 2020-IEEE Conference on Computer Communications*. IEEE, 2020, pp. 1083–1092.

[20] H. Liu, X. Li, W. Zhou, Y. Chen, Y. He, H. Xue, W. Zhang, and N. Yu, "Spatial-phase shallow learning: rethinking face forgery detection in frequency domain," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 772–781.

[21] Y. Lu and T. Ebrahimi, "Assessment framework for deepfake detection in real-world situations," *EURASIP Journal on Image and Video Processing*, vol. 2024, no. 1, p. 6, 2024.

[22] Y. Luo, Y. Zhang, J. Yan, and W. Liu, "Generalizing face forgery detection with high-frequency features," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 16 317–16 326.

[23] A. Mercat, M. Viitanen, and J. Vanne, "Uvg dataset: 50/120fps 4k sequences for video codec analysis and development," in *Proceedings of the 11th ACM Multimedia Systems Conference*, 2020, pp. 297–302.

[24] S. Milani, P. Bestagini, M. Tagliasacchi, and S. Tubaro, "Multiple compression detection for video sequences," in *2012 IEEE 14th International Workshop on Multimedia Signal Processing (MMSP)*. IEEE, 2012, pp. 112–117.

[25] A. Rossler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Nießner, "Faceforensics++: Learning to detect manipulated facial images," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 1–11.

[26] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 618–626.

[27] K. Shiohara and T. Yamasaki, "Detecting deepfakes with self-blended images," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 18 720–18 729.

[28] D. F. Smith, A. Wiliem, and B. C. Lovell, "Face recognition on consumer devices: Reflections on replay attacks," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 4, pp. 736–745, 2015.

[29] L. Song, Y. Liu, X. Yang, G. Zhai, R. Xie, and W. Zhang, "The sjtu hdr video sequence dataset," in *Proceedings of International Conference on Quality of Multimedia Experience (QoMEX 2016)*, 2016, p. 100.

[30] L. Song, X. Tang, W. Zhang, X. Yang, and P. Xia, "The sjtu 4k video sequence dataset," in *2013 Fifth International Workshop on Quality of Multimedia Experience (QoMEX)*. IEEE, 2013, pp. 34–35.

[31] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller, "Striving for simplicity: The all convolutional net," *arXiv preprint arXiv:1412.6806*, 2014.

[32] D. Tang, Z. Zhou, Y. Zhang, and K. Zhang, "Face flashing: a secure liveness detection protocol based on light reflections," *arXiv preprint arXiv:1801.01949*, 2018.

[33] S. Tomar, "Converting video formats with ffmpeg," *Linux journal*, vol. 2006, no. 146, p. 10, 2006.

[34] D. Vazquez-Padin, M. Fontani, T. Bianchi, P. Comesaña, A. Piva, and M. Barni, "Detection of video double encoding with gop size estimation," in *2012 IEEE International Workshop on Information Forensics and Security (WIFS)*. IEEE, 2012, pp. 151–156.

[35] H. Wang, W. Gan, S. Hu, J. Y. Lin, L. Jin, L. Song, P. Wang, I. Katsavounidis, A. Aaron, and C.-C. J. Kuo, "Mcl-jcv: a jnd-based h. 264/avc video quality assessment dataset," in *2016 IEEE international conference on image processing (ICIP)*. IEEE, 2016, pp. 1509–1513.

[36] Xiph.org. (2024) Video test media. Xiph.org Foundation. [Online]. Available: https://media.xiph.org/video/derf/. Accessed: 2024-11-13. [Online]. Available: https://media.xiph.org/video/derf/

[37] X. Xu, T. Zhao, Z. Zhang, Z. Li, J. Wu, A. Achille, and M. Srivastava, "Principles of designing robust remote face anti-spoofing systems," *arXiv preprint arXiv:2406.03684*, 2024.