Expert Systems With Applications 317 (2026) 131761

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Robust deepfake detection in compressed videos with scalable network strategies

Gianpaolo Perelli [a], Marco Micheletto [a,*], Sara Concas [a], Giovanni Puglisi [b], Gian Luca Marcialis [a]

[a] University of Cagliari, Department of Electrical and Electronic Engineering, Via Marengo 3, Cagliari, 09123, Italy
[b] University of Cagliari, Department of Mathematics and Computer Science, Via Ospedale 72, Cagliari, 09124, Italy

ARTICLE INFO

Keywords:
Deepfake detection
Biometrics
Pattern recognition
Computer vision

ABSTRACT

Deepfakes leverage artificial intelligence to generate highly realistic but falsified visual content, raising concerns for security and trust in digital media. Detecting such manipulations becomes more challenging when videos are compressed, as compression algorithms introduce artifacts that obscure forensic evidence. One possible solution is to train separate models for different compression levels; however, this approach increases computational costs and limits scalability.

To address this challenge, we introduce a unified framework designed to improve robustness against varying degrees of video compression. Our approach combines (i) a dedicated MPEG-based augmentation strategy tailored for compressed videos, and (ii) two architectural designs named Multi-Head (MHN) and the Multi-Branch Network (MBN). The MHN extends a standard backbone by appending lightweight output layers, or "heads", that jointly predict deepfake likelihood and compression level, enabling compression-aware detection with minimal architectural changes. The MBN combines multiple MHNs into a modular, parallel architecture, offering an alternative to conventional depth-based model scaling.

Experiments on the FaceForensics++ and Celeb-DF datasets show that both MHN and MBN improve detection performance in compressed scenarios. Notably, MHN applied to a lightweight backbone outperforms deeper and more complex models without the multi-head extension, making the proposed solution well-suited for deployment in resource-constrained settings.

1. Introduction

The term Deepfake (Heidari et al., 2024) refers to techniques that leverage artificial intelligence (AI) to create highly realistic images or videos by digitally manipulating facial features. Face manipulation can involve swapping faces, where the face of one person is replaced with another (Li et al., 2019), modifying facial attributes, such as altering expressions or ages (Thies et al., 2016), or creating entirely new faces that do not exist in reality (Thies et al., 2019). The advent of Generative Adversarial Networks (GANs) (Goodfellow et al., 2014) has significantly advanced these methods, enabling the production of hyper-realistic fake videos that pose significant threats to information security and digital trustworthiness. In the current digital era, the rise of deepfakes has compounded the complexity and prevalence of misinformation. News platforms and social media channels, driven by revenue models prioritizing clicks and views over content accuracy, often contribute to spreading fake news. Additionally, deepfakes have been weaponized to conduct scams, manipulate public sentiment, and tarnish reputations (Rancourt-Raymond & Smaili, 2023). These malicious uses can have severe consequences, including financial fraud, personal and corporate damage, and even geopolitical ramifications (Blauth et al., 2022; Mubarak et al., 2023). Consequently, the detection of deepfakes has become an essential area of research. Advanced detection methods, often underpinned by deep neural networks, have shown commendable performance in distinguishing fake content from real in controlled environments (Abbas & Taeihagh, 2024). However, in real-world scenarios, videos are often shared on social media platforms, where they undergo compression in order to save bandwidth and improve upload times. As a matter of fact, compression algorithms, such as MPEG-4 (Sikora, 1997) or H.264 (Wiegand et al., 2003), introduce artifacts such as blocking, ringing, blurring, or jagged edges, which degrade visual quality and can obscure the features leveraged by detection algorithms (Fig. 1). As a result, the accuracy

* Corresponding author.
E-mail addresses: gianpaolo.perelli@unica.it (G. Perelli), marco.micheletto@unica.it (M. Micheletto), sara.concas90c@unica.it (S. Concas), puglisi@unica.it (G. Puglisi), gianluca.marcialis@unica.it (G. Luca Marcialis).

https://doi.org/10.1016/j.eswa.2026.131761
Received 6 August 2025; Received in revised form 10 February 2026; Accepted 17 February 2026
Available online 20 February 2026

of deepfake detectors significantly decreases in these compressed environments (Khormali & Yuan, 2022; Kohli & Gupta, 2022; Rossler et al., 2019). In current literature, video compression is primarily viewed as a robustness challenge rather than a target for enhancement. Most existing methods address generalization across different datasets or manipulation techniques while often neglecting the critical aspect of cross-compression generalization (Heidari et al., 2024). A commonly adopted and straightforward approach involves training multiple models to handle different compression domains (Khormali & Yuan, 2022). However, this strategy fails when encountering unknown compression levels and is computationally demanding regarding storage and memory usage.

To address the aforementioned problems, we propose a strategy that makes the model explicitly aware of compression conditions by introducing compression estimation as an auxiliary supervision task. This approach requires solving three fundamental challenges: (1) handling diverse compression levels without separate models, (2) maintaining computational efficiency for deployment, and (3) preserving detection accuracy across quality variations.

Our first step toward meeting these requirements is the design of a video compression augmentation framework that generates multiple compressed versions of each video while providing systematic coverage of the compression spectrum, mirroring typical social media video processing (Maiano et al., 2021). The goal was to expose the model to a comprehensive range of compression artifacts during training. Then, to fully exploit the diversity introduced by this framework, we propose two key architectural designs: the Multi-Head Network (MHN) and the Multi-Branch Network (MBN).

The MHN architecture refines a standard backbone by restructuring the terminal layer into two specialized heads: one dedicated to classifying deepfakes and the other focused on estimating the compression level. The joint formulation of these tasks enables the network to concurrently learn and differentiate between authentic and manipulated content while assessing the degree of compression, facilitating more stable predictions under varying compression domains. Building on the MHN, the Multi-Branch Network introduces a modular architecture composed of multiple parallel branches, each implemented as an independent MHN. The motivation for this design is to provide an alternative to conventional deep architectures, which increase capacity through depth at the cost of significant computational overhead. Since the backbone determines the MHN's complexity, adopting lightweight variants improves efficiency but may reduce performance. The MBN addresses this limitation by enabling controlled horizontal scaling by replicating smaller MHN units. Each branch is trained on a different bootstrap-resampled subset of the training data, promoting representational diversity. Our experiments show that an MBN with compact branches can outperform a deeper single-stream model with comparable or even higher parameter counts.

In summary, the key contributions of this work are:

- We design a video compression-aware data augmentation strategy specifically tailored for deepfake detection, which systematically exposes the model to a continuous range of compression levels while keeping the number of training frames per epoch comparable to standard baselines.
- We propose the Multi-Head Network, which jointly performs Deepfake classification and compression level estimation with minimal architectural overhead.
- We introduce the Multi-Branch Network, a scalable modular architecture that ensembles multiple MHNs trained on resampled subsets, effectively improving performance under compression while maintaining computational efficiency.

The rest of the paper is organized as follows: Section 2 reviews the related work on deepfake detection under varying compression conditions. Section 3 details the proposed framework, including the data augmentation strategy and the Multi-Head and Multi-Branch Network architectures. Section 4 presents the experimental results, analyzing the effectiveness of the proposed approach on multiple datasets. Finally, Section 5 concludes the paper.

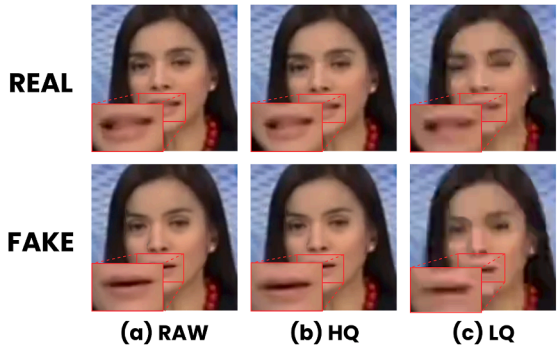


Fig. 1. Examples of compressed real and fake images across different compression levels. The top row shows real images, while the bottom row shows fake images. (a) RAW: Original images with no compression, where forged traces in fake images are visible. (b) HQ: High-quality compressed images where forged traces are still noticeable, but compression artifacts are minimal. (c) LQ: Low-quality compressed images with prominent blocking artifacts and significant blurring of forged traces.

2. Related works

Deepfake detection has made significant progress in recent years, driven by the release of numerous datasets and the development of state-of-the-art (SOTA) detection methods. While many recent works have focused on improving generalization across deepfake generation techniques (Guarnera et al., 2022), relatively less attention has been given to robustness against compression artifacts, a key factor in real-world deployment scenarios, where media is shared through social platforms or messaging apps, which apply compression processes that degrade the data. Compression severely impacts detection reliability, yet most existing methods are optimized under clean or lightly compressed conditions. To address these challenges, researchers have explored a range of approaches, focusing primarily on spatial and frequency domains. Spatial-based methods analyze image and video data directly in the pixel domain, aiming to detect inconsistencies introduced by manipulation. For example, Mitra et al. (2020) introduced a Convolutional Neural Network (CNN) followed by a classifier trained on images with intermediate compression levels to improve system resilience. Similarly, Zhou et al. (Zhou et al., 2020) proposed a 3D-ResNext model that focuses on learning relevant features from selected video frames while also employing data enhancement techniques during preprocessing to mitigate the information loss caused by compression. In this context, attention mechanisms have proven effective. The authors of Zhao et al. (2021) utilized a multi-attentional network (MAT) to enhance shallow textures and integrate them with deeper semantic features, dynamically adapting attention across facial regions. Ref. Sun et al. (2022) introduced a Self-Information Attention (SIA) module to focus on high-information regions, improving detection accuracy across various models. In Cao et al. (2022), a different approach was taken by employing reconstruction learning, multi-scale graph reasoning, and attention mechanisms to model genuine face features and detect unknown forgeries, even in highly compressed media.

The second group of methods is frequency-based. These methods were developed under the hypothesis that compression artifacts manifest more distinctly in the frequency domain. For example, Ref. Qian et al. (2020) used two techniques: Frequency-aware Decomposition (FAD) to detect hidden forgery signs, and Local Frequency Statistics (LFS) to identify abnormal frequency patterns. Combined in a collaborative framework, they significantly improved detection accuracy, particularly on low-quality images and videos. Liu et al. (2021) employed phase spectrum analysis to detect upsampling artifacts, enhancing robustness

against compression-induced distortions. In Li et al. (2021), authors further integrated frequency analysis with metric learning techniques, introducing a Single-Center Loss (SCL) to improve feature separation between real and fake faces, along with an Adaptive Frequency Feature Generation Module (AFFGM) to capture forgery patterns dynamically. In addition to the above two groups of methods, relational learning approaches emerged as a promising direction, utilizing both spatial and frequency cues to analyze the relationships between different facial regions. They were particularly effective when compression distorts these connections. Chen et al. (2021) introduced a Multi-scale Patch Similarity Module (MPSM) to capture subtle differences between real and forged local face regions. To further enhance the representation of local features, the authors also proposed the RGB-Frequency Attention Module (RFAM), which combines information from both the RGB color space and the frequency domain. The relationships between facial regions are also exploited in Yang et al. (2023). The framework includes a spatio-temporal Attention module to capture features from facial regions and a Masked Relation Learner (MRL) to propagate relational information.

Recent research has also explicitly addressed the problem of detecting deepfakes in compressed environments, recognizing the growing importance of this issue. Zhang et al. (2021) proposed a self-supervised decoupling network (SSDN) that uses compression ratios as self-supervised signals to enhance the model's ability to learn both authenticity and compression features. Techniques such as "bleaching" preprocessing, as seen in (Li et al., 2023), simulate compressed images to improve detector robustness without the need for retraining. Other studies have sought to identify features less affected by compression, such as the learned visibility matrix in Chhabra et al. (2023), which forces the model to focus on imperceptible artifacts, and the FAMM framework in Liao et al. (2023), which analyzes facial muscle motions across video frames, demonstrating resilience to compression. Some approaches, such as the deceptive model in Chen and Hsu (2023), aim to improve generalization capability and protection against adversarial attacks, addressing the challenge of compressed media head-on. Innovative methods like the anti-counterfeit label mechanism (Zhao et al., 2023) protect media by embedding watermarks into facial identity features, making them sensitive to malicious modifications and robust against resizing and compression.

A second group of methods targets low-quality social-network content more directly. Li et al. (2023) proposed the Spatial Restore Detection Framework (SRDF), which combines restoration blocks and attention modules. Gao et al. (2024) introduced a high-frequency enhancement network (HiFE) designed for highly compressed content, where local and global high-frequency components are strengthened in order to recover forgery cues degraded by lossy coding. Chen et al. (2024) proposed a detector based on 3D spatiotemporal trajectories that aggregates motion patterns across frames. Overall, the majority of these works, despite representing significant advancements in deepfake detection, still exhibit substantial limitations in effectively addressing the challenge of varying compression levels, since they have relied predominantly on the FaceForensics++ dataset, which includes only two predefined compression levels (Humidan et al., 2022). As a result, two main strategies have been adopted to mitigate this problem: training separate models for each compression level, which is computationally expensive and impractical for deployment, or training on a single compression level, typically the higher-quality compression, to mitigate performance loss. However, this latter strategy presumes an inherent robustness to compression artifacts that does not hold in practice, especially under compression levels not explicitly seen during training.

To the best of our knowledge, no prior work explicitly incorporates compression level as a structured learning dimension within the model architecture. Existing approaches tend to treat compression as a nuisance factor, often mitigated by preprocessing steps or fixed-level training, rather than modeling it explicitly. In contrast, the approach introduced in this work encodes compression-aware representations within the architecture itself, enabling robust detection across a wide spectrum of compression conditions without requiring multiple compression-specific models. The proposed method is outlined in the following section.

3. The proposed approach

The goal of our proposed framework is to enhance deepfake detection, specifically targeting facial deepfakes distributed in compressed media. The primary challenge we aim to address is the variability of the performance of deepfake detectors at different compression levels. Our framework is thus specifically designed to mitigate performance degradation caused by varying degrees of compression typically encountered in practical scenarios. The overall structure of our approach is illustrated in Fig. 2. First, input videos undergo our proposed compression-based data augmentation step (Section 3.1), in which multiple compressed versions of each video are generated by randomly selecting compression levels from a predefined interval. Subsequently, a preprocessing step consisting of face detection and cropping is performed directly on these compressed videos. The resulting cropped facial frames are then used for training. The preprocessed frames are fed into multiple instances of the Multi-Head Network (MHN) (Section 3.2), each independently performing deepfake classification and compression level estimation. Finally, the MHN instances are combined into a Multi-Branch Network (MBN) (Section 3.3), enabling the framework to leverage multiple models in parallel for more robust detection. A detailed description of the building blocks of the proposed deepfake detection framework is presented in the following sections.

3.1. Compression-based data augmentation

It is well established that for deepfake images, certain data augmentation techniques prove beneficial in terms of robustness and cross-dataset or cross-manipulation generalization (Bondi et al., 2020; Wang et al., 2020). These methods typically apply transformations to samples during training, such as compression, rotation, scaling, and mirroring, but they have not yet been specifically tailored to address compression artifacts. Moreover, their application to video data remains problematic.

In fact, compression is straightforward in the case of images, as it simply involves applying a function directly to the image. For video frames instead, compression is more complex as it requires processing the entire video to account for both spatial and temporal redundancies across multiple frames (Ma et al., 2019). To the best of our knowledge, no existing augmentation techniques explicitly address these complexities for deepfake detection in videos. To achieve this, we designed a novel augmentation strategy tailored specifically for deepfake detection in videos, ensuring that the network is exposed to a broad spectrum of compression levels during training. To formally address the design of our augmentation strategy, let N_{tot} represent the total number of frames to be used for training. This can be expressed as:

$$N_{tot} = N_v \cdot N_f \cdot N_q \tag{1}$$

where N_v is the number of videos, N_f is the number of frames sampled per video, and N_q is the number of *compressed* variants generated for each sampled frame by randomly selecting N_q compression levels from a predefined interval $\mathcal{I} = [c_{\min}, c_{\max}] \cap \mathbb{Z}$, where c denotes a generic compression-control parameter. We define c as a general codec control variable governing the strength of re-encoding, so the same sampling scheme applies to any MPEG-style video encoder by selecting the corresponding quality parameter; the specific instantiation adopted in our experiments is reported in Section 4.2.

Under this definition, N_{tot} counts only frames extracted from the compressed replicas. When the RAW version is also included in the training pool, the corresponding total becomes $N_v \cdot N_f \cdot (1 + N_q)$. Specifically, for each original video v, we generate a set of compressed versions $\mathcal{V}$ defined as:

$$\mathcal{V} = \{v_c \mid c \in \mathcal{I}\} \tag{2}$$

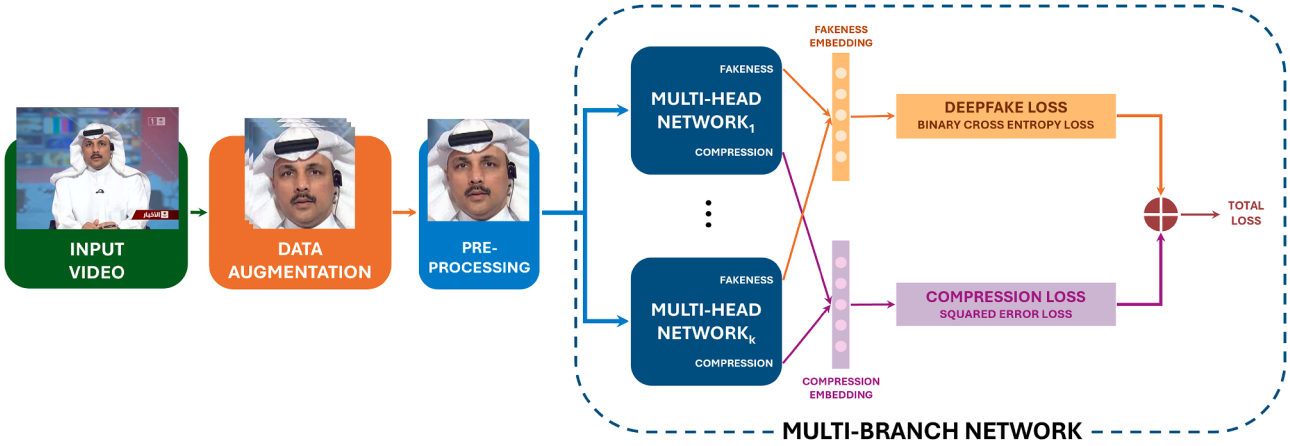


Fig. 2. Overview of the proposed deepfake detection framework. Each video is first augmented at multiple compression levels, then face regions are cropped and processed by parallel Multi-Head Networks (MHNs) for joint deepfake detection and compression estimation. The MHNs form the Multi-Branch Network (MBN).

where v_c denotes the compressed version of video v obtained by applying the compression level c. The corresponding compressed clips are produced in an offline pre-processing stage and stored on disk, so no re-encoding is required during network training. The augmentation strategy determines which elements of $\mathcal{V}$ are instantiated for each video and how compression levels are assigned and distributed across videos and frames. Two strategies are considered in this work: a video-centric strategy and a frame-centric strategy. In the *video-centric* strategy, for each video v, we randomly select N_q compressed versions from the set $\mathcal{V}$. Then, we randomly sample N_f frames from these selected compressed versions, resulting in a total of N_{tot} frames for training. As illustrated in Fig. 3a, this strategy may yield a non-uniform distribution of compression levels across the sampled frames, as the representation of each compression level depends on the random frame selection process. For instance, consider the configuration used in the figure, where the compression interval is defined as $\mathcal{I} = [15, 45]$, with $N_q = 2$ and $N_f = 100$. In this case, each video is associated with only two randomly selected CRF values (e.g., 18 and 33), and the 100 frames used for training are sampled across these two compressed versions. This can result in an over- or under-representation of certain compression levels in the training set.

Conversely, in the *frame-centric* strategy, we first select N_f frames from each original video v. Then, for each selected frame, we randomly choose N_q compressed versions of the same frame from the corresponding videos in the set $\mathcal{V}$. This again results in N_{tot} frames. Unlike the video-centric approach, the frame-centric strategy ensures that each frame individually covers multiple, randomly selected compression levels, potentially leading to a more uniform representation of the compression interval. This is confirmed by the example shown in Fig. 3b, which uses the same configuration as above and reveals a significantly flatter and more balanced distribution across the compression spectrum.

It is worth noting that both strategies generate the same total number of frames N_{tot}, so the amount of data that enters the training pipeline and the computational load per epoch remain comparable to standard baselines. Instead, storage requirements differ across strategies because the number of encoded versions per video depends on how compression levels are assigned. In the video-centric setting, each video v is encoded at N_q CRF values, therefore the number of stored compressed copies per video equals N_q. In the frame-centric setting, frame-level sampling relies on the full set $\mathcal{V}$ of compressed variants for each video, since MPEG compression operates on the complete sequence. The encoder produces $|\mathcal{I}|$ compressed versions per video in a single offline step and stores them for reuse. Under the assumption of a comparable average size for all encoded clips, the storage associated with the frame-centric and video-centric strategies satisfies

$$\frac{M_{\mathrm{fc}}(v)}{M_{\mathrm{vc}}(v)} \approx \frac{|\mathcal{I}|}{N_q}. \tag{3}$$

In the configuration adopted in this work, $\mathcal{I} = \{15, \ldots, 45\}$ and $N_q = 2$, so the factor is approximately 15.5, which corresponds to an overhead of about 15-16 compressed copies per video in the frame-centric configuration with respect to the video-centric one. A detailed empirical analysis comparing the two strategies in terms of their impact on deepfake detection performance will be presented in the Experimental Section (Section 4.4.2). Additionally, to maximize the effectiveness of this augmentation strategy, we experimented with adding a head to the neural network to detect the level of compression applied to the input, as detailed in the subsequent section.

3.2. Multi-head network (MHN)

To integrate compression-related information into the learning process, we introduce the Multi-Head Network (MHN), a two-branch architecture designed to address deepfake classification and compression level estimation jointly. In practical terms, the MHN can be viewed as a standard deepfake classifier that receives one input frame and produces two outputs: a probability that the frame is fake and a numerical estimate of the compression level. Both outputs are computed from the same feature map produced by the backbone, so the internal representation is encouraged to remain informative for authenticity while retaining sensitivity to compression. As illustrated in Fig. 4, the MHN extends a standard backbone by appending two parallel output heads: a classification head that distinguishes between deepfake and authentic samples, and a regression head that estimates the level of compression applied to the input frame.

This dual-head design is inspired by multi-task and multi-label learning, which are well-established concepts in machine learning. Multi-task learning scenarios often utilize multi-head models where each head is responsible for a different task, improving the overall performance by leveraging shared representations (Zhang & Yang, 2021). Similarly, in multi-label classification, different heads can handle distinct labels or categories, optimizing the model's capacity to learn complex relationships within the data (Tarekegn et al., 2024).

Based on these principles, we can view the problem of deepfake detection under compression as a specific case of multi-task learning. Each head follows a fixed architectural structure while allowing flexibility in its internal configuration to adapt to different backbone architectures or

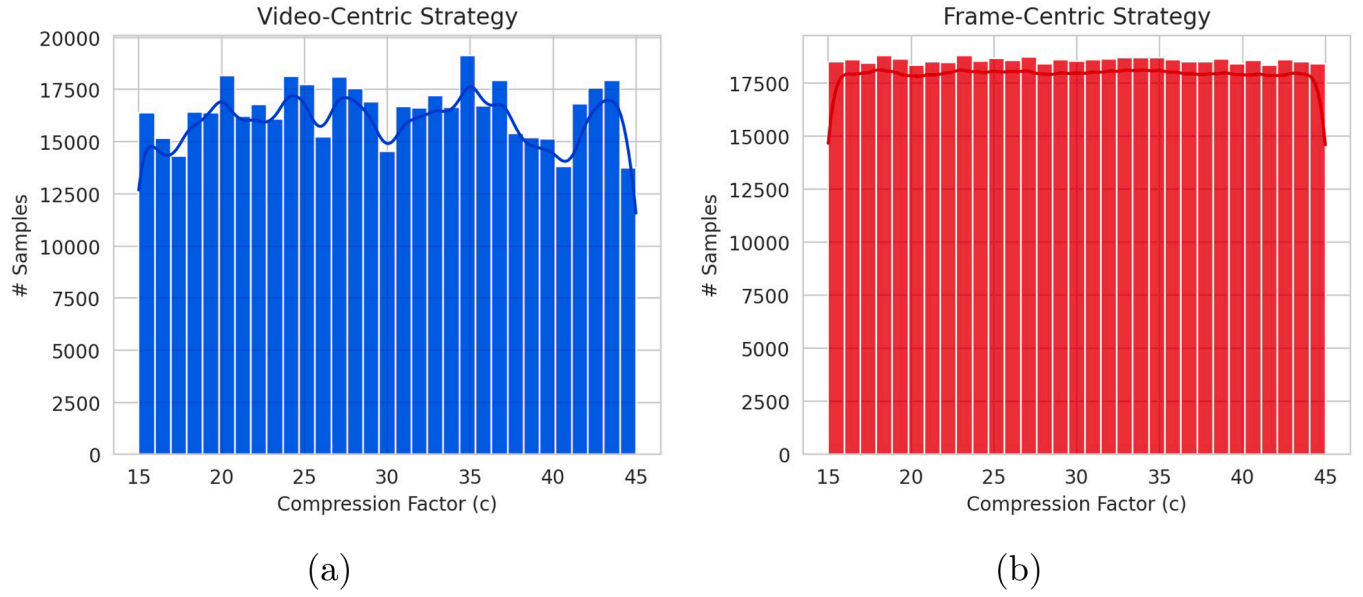


Fig. 3. Distribution of sampled frames per compression level $c \in \mathcal{I} = [15, 45]$ for the video-centric (a) and frame-centric (b) augmentation strategies with $N_q = 2$ and $N_f = 100$. Compression is performed using the H.264 codec, where c denotes the Constant Rate Factor (CRF).

computational constraints. In our configuration, we employed the following components:

- *Fully Connected (FC) layer*: the first dense layer projects the features extracted by the backbone into a lower-dimensional representation. The number of units can be adjusted based on the model capacity and dataset characteristics, and is treated as a design parameter within the architecture.
- *50% Dropout layer*: randomly set half of the neurons to zero during each training step. A dropout rate of 0.5 is commonly used as it balances retaining enough information for learning while effectively mitigating overfitting, as suggested in Srivastava et al. (2014).
- *Batch normalization*: included to stabilize the learning process and accelerate convergence;
- *ReLU activation function*: a widely adopted non-linearity function that enables efficient gradient propagation and stable training;
- *FC with a single neuron*: the classification branch uses a sigmoid activation function to output a probability score for the binary classification task (real vs. fake), while the regression branch uses a linear activation function to output a continuous value representing the compression level of the input image.

To construct a MHN from a pre-trained model, depending on the network, it may be necessary to add a layer to connect to the fully connected layers of the MHN. We used a Global Average Pooling 2D layer for this purpose. Otherwise, removing only the final layer suffices to connect to the MH module without additional layers.

The overall loss function in the MHN is a weighted sum of the losses from the classification and regression tasks:

$$\mathcal{L}_{\text{tot}} = \lambda_{\text{BCE}}\mathcal{L}_{\text{BCE}} + \lambda_{\text{SE}}\mathcal{L}_{\text{SE}} \tag{4}$$

where $\mathcal{L}_{\text{BCE}}$ and $\mathcal{L}_{\text{SE}}$ represent Binary Cross Entropy and Squared Error losses respectively whereas λ_{BCE} and λ_{SE} are the related weights. Regarding the supervision signal, we assigned a regression target of 0 to RAW frames. Although there is a physical distinction between RAW data and theoretical zero-level compression, this assignment provides a clear numerical baseline for comparison with compressed samples (CRF range $[15, 45]$). To facilitate the multi-task formulation, regression targets were globally standardized to zero mean and unit variance during pre-processing. This normalization ensures that the scale of regression gradients is consistent with those of the classification head, thereby supporting stable joint optimization. Consequently, equal weights were assigned to the classification and compression estimation tasks. Preliminary tests with alternative values on MobileNetV3Small (Howard et al., 2019) did not show clear advantages over the balanced setting and occasionally reduced optimisation stability.

The auxiliary regression head has two intended roles. First, joint optimization of classification and compression estimation encourages the shared representation to preserve sensitivity to the underlying compression level instead of treating compression as uncontrolled noise. Frames with similar manipulation patterns but different CRF values receive distinct regression targets, which promotes a partial separation between compression-related artefacts and features primarily associated with authenticity. The design does not assume a perfectly monotonic relation between regression accuracy and detection performance across all backbones and compression regimes, but it constrains the backbone to retain compression information in a structured way. Second, the predicted compression level provides an additional quantity that can be inspected alongside the deepfake score. Predictions on heavily compressed samples can be examined together with the estimated CRF in order to assess whether errors tend to concentrate in specific compression regimes. The architecture is therefore intended to facilitate a compression-aware reading of classifier outputs, while the empirical relation between compression estimation and detection performance is investigated in Section 4.4.4

3.3. Multi-branch network (MBN)

A standard solution often adopted when aiming to improve the accuracy of deepfake detection models is to increase the depth and complexity of deep learning architectures. While this approach can be effective, it typically results in deep, sequential architectures in which model capacity is increased by stacking layers along a single forward path. However, these architectures may be impractical when deepfake detection is required on-device, where computational resources are limited and the input data, such as compressed videos from messaging apps or social media, is often degraded. The Multi-Head Network helps address the classification of compressed videos without significantly increasing the number of parameters; however, it does not solve the computational burden of heavy backbone architectures.

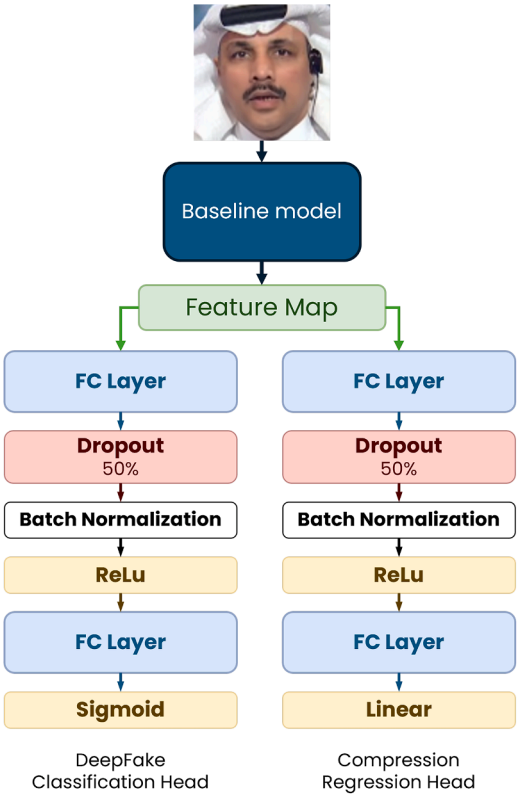


Fig. 4. Multi-Head Network (MHN) architecture. A shared feature map generated by the baseline backbone feeds two parallel heads dedicated to deepfake classification and compression-level estimation. The structure can be interpreted as a standard deepfake classifier equipped with an additional output that predicts the strength of compression from the same internal representation.

To address these challenges, we introduce the Multi-Branch Network, a modular architecture composed of B parallel branches, each implemented as an independent Multi-Head Network. At a conceptual level, the MBN can be interpreted as running several MHNs in parallel on the same input frame and letting a final fusion stage combine what each branch has learned. The design resembles an ensemble in which multiple models contribute complementary viewpoints, while the concatenation layer replaces simple score averaging and allows the final classifier to exploit joint feature patterns across branches. Instead of increasing depth within a single pathway, the MBN distributes capacity horizontally across multiple branches, enabling more flexible control over the trade-off between complexity and performance.

A key factor in the success of such ensembles is the diversity among classifiers, which allows the combined model to leverage complementary information and better capture the variability in the data (Yang, 2011). In order to promote such diversity in our MBN, we adopt the *bootstrap aggregating* (bagging) strategy, which builds upon the bootstrap resampling method introduced initially by Efron (1992) and later formalized for predictive ensemble learning (Breiman, 1996).

Bootstrap is a non-parametric resampling method used to approximate the sampling distribution of a statistic. Given a training set $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$, where N is the total number of examples, $\mathbf{x}_i$ is the input feature vector, and y_i the corresponding label, each bootstrap replicate $\mathcal{D}^{(b)}$ is formed by sampling N times with replacement from $\mathcal{D}$, for $b = 1, \ldots, B$, where B is the total number of replicates. Due to sampling with replacement, each replicate omits a fraction of the original samples: the probability that a given example does not appear in $\mathcal{D}^{(b)}$ is $(1 - 1/N)^N$, which converges to $e^{-1} \approx 0.368$ as N grows, so on average only about 63.2% of the unique examples are present in each replicate.

We exploit this mechanism in our MBN by assigning each bootstrap replicate $\mathcal{D}^{(b)}$ to a separate branch. For simplicity, we adopt a homogeneous configuration, replicating the same Multi-Head Network across all B branches, although the architecture naturally supports heterogeneous designs. The overall parameter count increases with the number and type of branches, but computational complexity can be explicitly controlled at design time. By choosing the number of branches and the backbone architecture assigned to each, it is possible to tailor the model to different resource constraints. Our goal is to demonstrate that this form of controlled diversity injection can outperform single-branch architectures with comparable or even higher parameter counts.

In terms of architectural design, the concatenation process occurs at the two penultimate layers within the MHN. As illustrated in Fig. 5, the final layers of the heads are discarded, and the FC layers pertaining to deepfake classification are concatenated on one side, while those related to compression level regression are concatenated on the other. These layers are then followed by a *Dropout* layer with a rate of 0.5, ensuring that all neurons and branches are engaged. Ultimately, an FC layer and a sigmoid activation function are utilized for the classification branch (fakeness), while an FC layer and a linear activation function are employed for the compression regression branch.

4. Experimental results

4.1. Data sets

We apply our proposed method on two state-of-the-art public datasets: FaceForensics++ (Rossler et al., 2019) and Celeb-DF (v2) (Li et al., 2020).

FaceForensics++ (FF++) is a widely-used dataset for evaluating deepfake detection systems. It comprises 1000 real and 4000 fake videos, created using four different face manipulation techniques: DeepFakes, FaceSwap, Face2Face, and Neural Textures. The videos in FF++ are sourced from YouTube and feature various individuals in various scenarios. The dataset is split into three versions based on compression levels using the H.264 codec: raw (videos without any compression), c23 (often referred to as HQ, high-quality), and c40 (often referred to as LQ, low quality). The c23 and c40 versions represent videos compressed with Constant Rate Factors (CRF) of 23 and 40, respectively. This diverse set of videos, both in terms of manipulation techniques and compression levels, provides a robust benchmark for testing the generalization and robustness of deepfake detection models across different conditions.

Celeb-DF (v2) is a challenging dataset for deepfake detection, consisting of 890 real and 5639 fake videos. The real videos feature 59 celebrities sourced from YouTube, providing a wide range of facial expressions, poses, and lighting conditions. The fake videos were generated using an improved DeepFake synthesis process that enhances visual quality and realism, making the dataset particularly valuable for evaluating the performance of detectors in real-world scenarios. Since this dataset does not provide any compressed versions, we applied the same compression levels as used in FF++ (CRF values of 23 and 40) to Celeb-DF (v2) to ensure a direct comparison between the two datasets.

4.2. Experimental protocol

This section details the experimental protocol employed to evaluate our deepfake detection framework. For the FF++ dataset, we divided the dataset into training, validation, and testing sets in a ratio of 720:140:140, following the original settings in Rossler et al. (2019). Similarly, the training and test sets for the Celeb-DF (v2) dataset were partitioned according to the official guidelines.

We uniformly sampled $N_f = 100$ frames from each video in the datasets to comprehensively assess our model's performance. To guarantee reproducibility and consistency, the sampling procedure was performed offline, and the extracted frames were saved to disk. The same set of frames is used across all experiments, preventing dynamic resampling and maintaining strict train, validation, and test splits. Com-

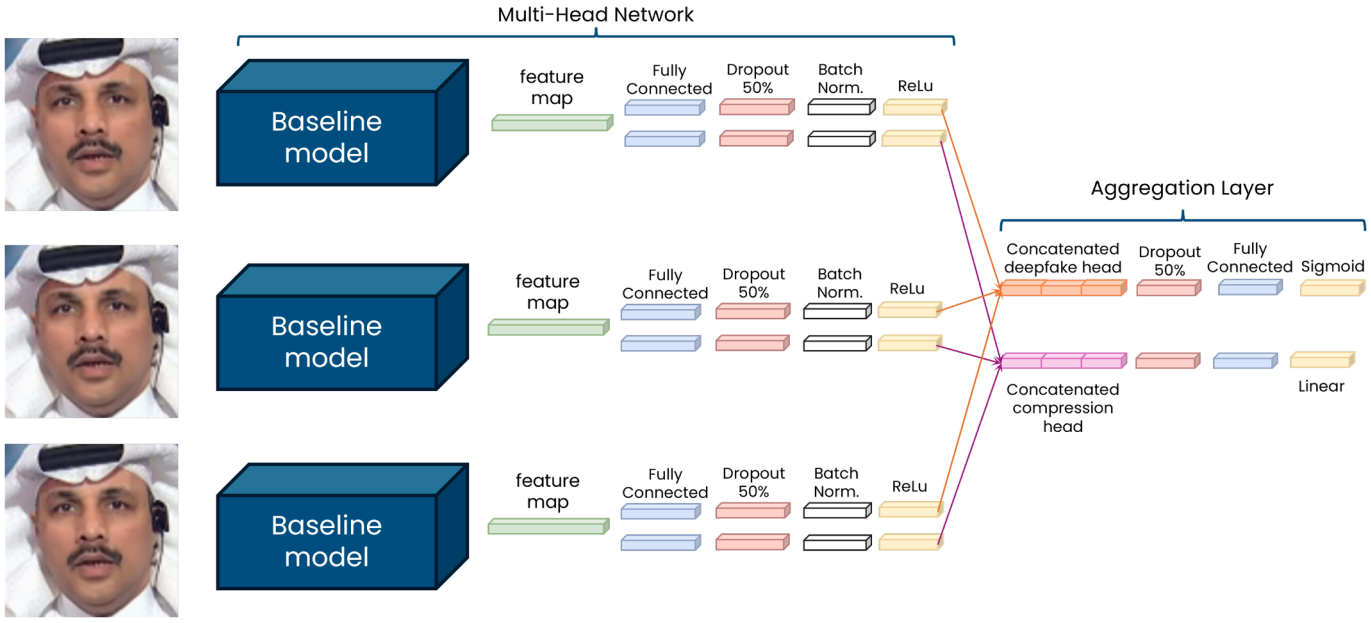


Fig. 5. Multi-Branch Network architecture. Each branch represents a replicated Multi-Head Network (MHN), independently trained to introduce diversity. The final layers of each branch are concatenated to aggregate features, providing a comprehensive representation for the subsequent classification and regression tasks. The overall structure is analogous to an ensemble of MHNs in which internal feature vectors are combined before the decision stage.

pression was applied using the H.264 codec, with compression levels sampled from the interval $\mathcal{I} = [15, 45]$, where $c_{\min} = 15$ corresponds to light compression and $c_{\max} = 45$ to strong compression. Such a range was selected to span a variety of compression levels typically encountered in real-world scenarios, where lossy compression is applied to reduce bitrate and storage requirements. Implementation relied on FFmpeg [1], with *libx264* handling bitrate control exclusively through the CRF parameter, while all remaining encoder settings, including GOP (Group of Pictures) structure, reference-frame configuration, and quantization matrices, followed the standard *libx264* defaults. The resulting configuration reflects a common H.264 deployment in which CRF determines the effective bitrate, and other compression parameters are governed by encoder defaults.

The complete set of compressed versions $\mathcal{V}$ was generated accordingly, as defined in Eq. (2). Among the two augmentation strategies introduced in Section 3.1, we adopted the frame-centric strategy, which yielded superior results in our evaluation (see Section 4.4.2 for a comparative analysis). The number of compression levels used in the augmentation process was fixed to $N_q = 2$ to ensure that each raw frame is associated with more than one compression level and to provide variability across training samples. Therefore, the total number of training frames used was $N_{tot} = N_v \cdot N_f \cdot (1 + N_q) = N_v \cdot 300$.

Subsequently, we employed the Multi-task Cascaded Convolutional Networks (MTCNN) (Yin & Liu, 2017) to detect faces in the scene. Given the presence of multiple faces in certain videos, particularly those in the FF++ dataset, it was crucial to ensure that the extracted face was the manipulated one. In some cases, errors were found where the manipulation was not applied to the target face but to background regions. For proper training, our algorithm selects only the bounding box within the area of the mask provided by the dataset, applying a conservative approach by scaling it by a factor of 2 to ensure that the entire face region is captured. Once the correct bounding box was obtained, we followed the methodology of Rossler et al. (2019), extending the bounding box by a factor of 1.3 while keeping the center fixed and maintaining a square shape. This enlargement was necessary to include slightly external facial features, such as the neck, where manipulations often occur in attacks such as lip sync. Moreover, to ensure that the network utilizes all available pixels effectively, we applied a random shift, either negative or positive, to the bounding box coordinates by a variable size up to a maximum of 10% of the bounding box side length. This additional augmentation step occurs with a probability of 75% of the frames. Finally, the cropped face was resized to 224x224x3 pixels and normalized to the range [-1, 1] for network input. The entire process is illustrated in Fig. 6.

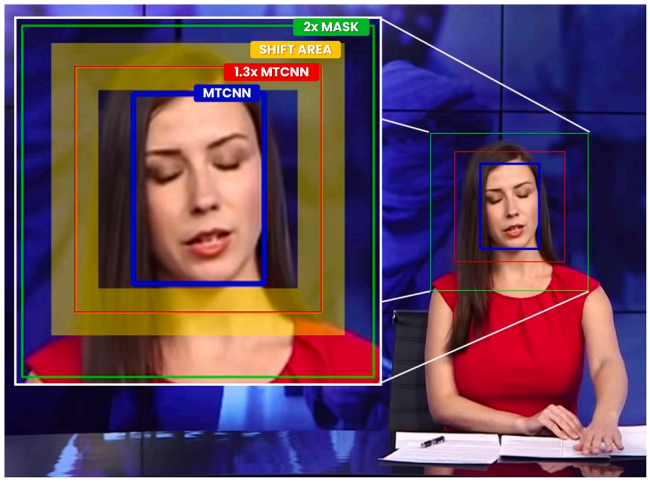


Fig. 6. Face cropping process: the green box represents the scaled mask region (x2), the blue box shows the initial MTCNN (Yin & Liu, 2017) detection, the red box is the extended bounding box (x1.3), and the yellow shaded area indicates potential shifts up to 10% of the bounding box side length.

We investigated several network architectures as baselines, selected for their balance of performance and computational efficiency:

- **MobileNetV3Small** (Howard et al., 2019): With its low parameter count (~1M), MobileNetV3Small is suitable for mobile and embedded applications, providing a lightweight yet effective option for real-time deepfake detection.

[1] https://www.ffmpeg.org-lastaccessed:5thDecember2025

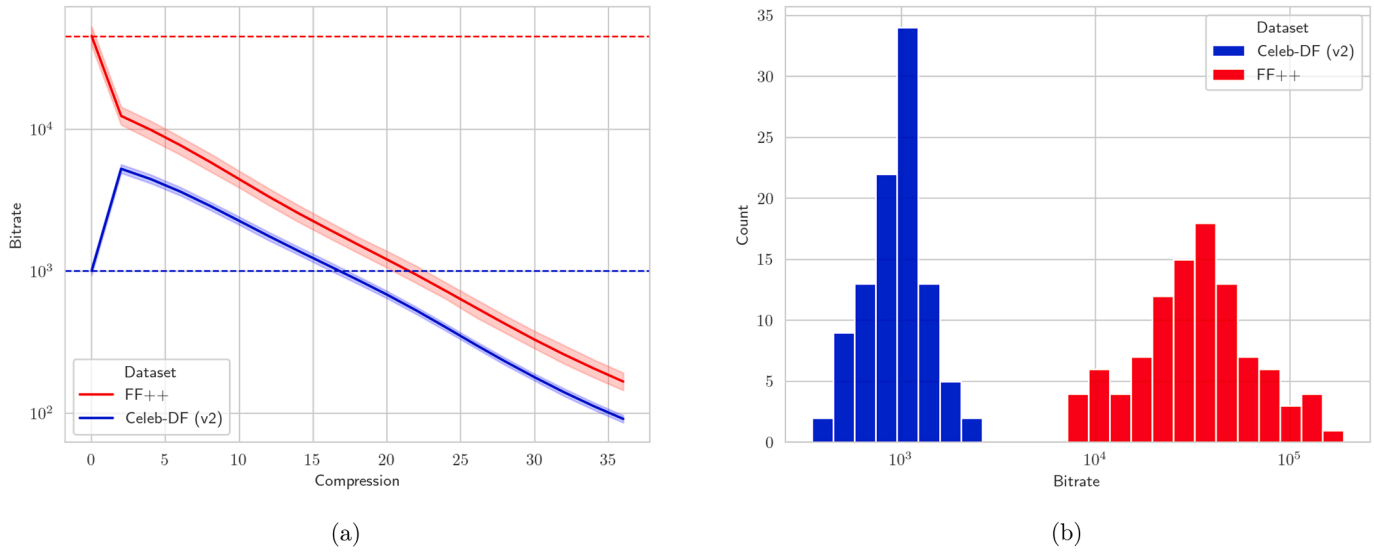


Fig. 7. (a) Bitrate as a function of CRF for 100 randomly selected real videos from the FF++ and Celeb-DF (v2) datasets. The solid lines represent the average bitrate, while the shaded areas indicate the 95% confidence interval across the selected videos. (b) Distribution of the bitrate for 100 raw videos from the Celeb-DF (v2) and FF++ datasets, highlighting the disparity in their initial quality levels. The value c = 0 denotes the bitrate of the original source videos (RAW), serving as the uncompressed baseline before any additional encoding applied in this work.

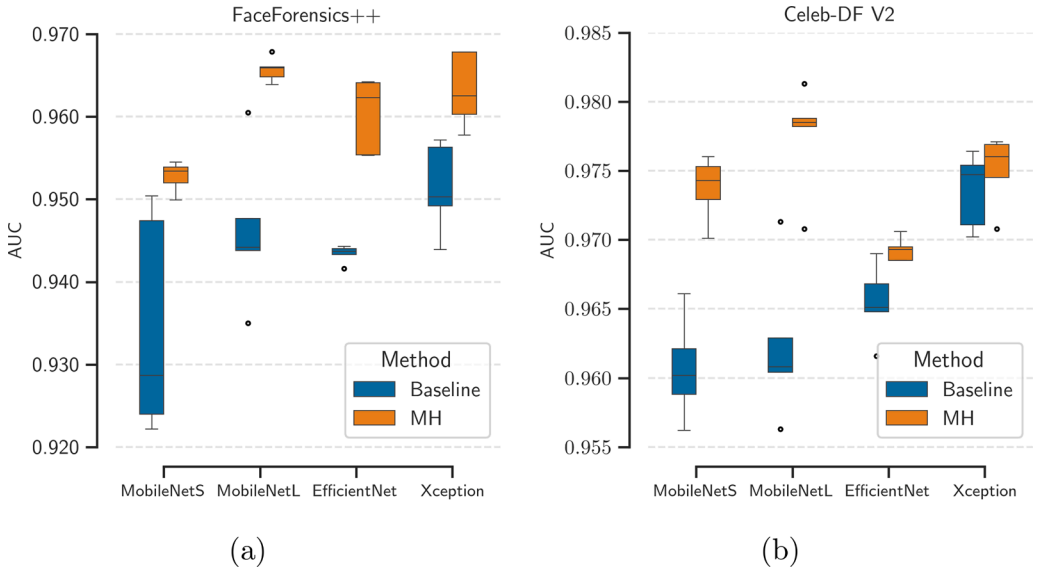


Fig. 8. Boxplots of AUC scores for baseline (Simple) and Multi-Head (MH)-enhanced models, computed over five training runs on the FF++ dataset (a) and the Celeb-DF (v2) dataset (b).

- **MobileNetV3Large** (Howard et al., 2019): This variant offers increased capacity and complexity, making it more suitable for scenarios requiring higher accuracy without significant computational overhead. It has approximately 3M parameters.
- **EfficientNetV2B0** (Tan and Le, 2021): Known for its balance between accuracy and efficiency, EfficientNetV2B0, with approximately 6M parameters, provides a robust framework for deepfake detection.
- **Xception** (Chollet, 2017): Renowned for its high accuracy in image classification tasks, Xception serves as a common benchmark in deepfake detection studies, offering a comprehensive evaluation of our approach's effectiveness, albeit at the cost of significantly higher computational demands (~21M parameters).

We added our proposed Multi-Head (MH) and Multi-Branch (MB) enhancements to these baseline architectures. It is worth remarking that these modifications can be applied to any network architecture chosen as a "backbone". Our selected baselines serve as exemplars to demonstrate on a representative set of architectures the efficiency and scalability of our approach in terms of computational cost and parameter count.

4.3. Implementation details

The Multi-Head model is generated by keeping the feature extraction part of the baseline model (pre-trained with ImageNet), adding a Global Average Pooling layer, and two heads. Each head includes a series of layers as described in Section 3.2, and the parameters initialization used are the default in Keras. The fully connected layers are initialized with Glorot uniform initialization for kernels (Glorot & Bengio, 2010) and zero for biases; no seed is set in the dropout layer; batch normalization has a momentum of 0.99 and epsilon of 0.001. The Multi-Branch model training occurs in several steps: first, we train the branches separately, one

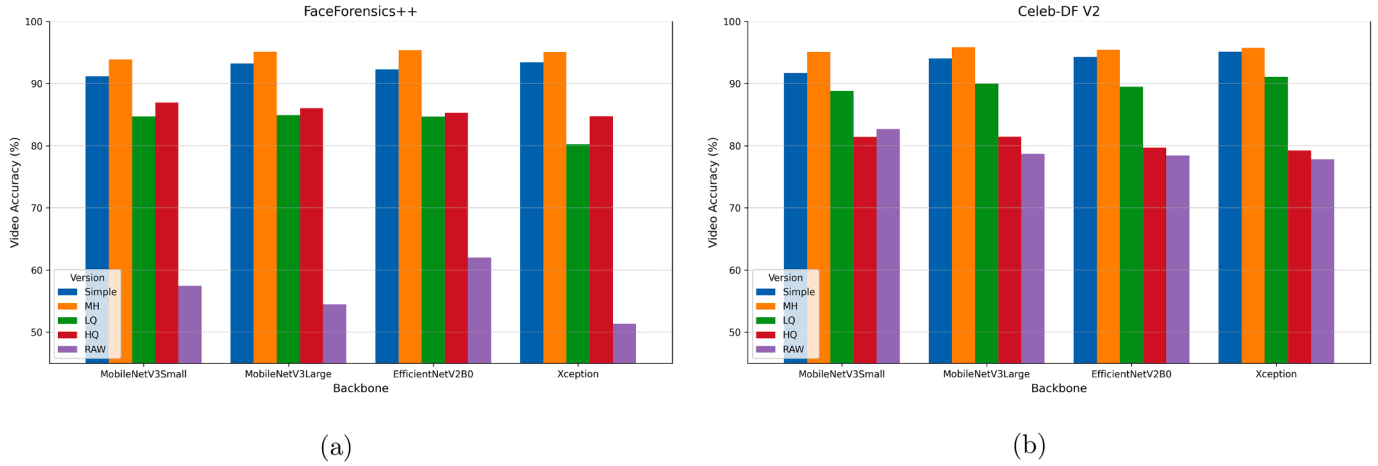


Fig. 9. Comparison of the mean video accuracy, averaged across all three compression levels (LQ, HQ, and RAW), for the FaceForensics++ (a) and Celeb-DF (v2) (b) datasets, using different model architectures and five training approaches: Simple (augmentation only), MH (Multi-Head architecture), and models trained on RAW, HQ, or LQ images.

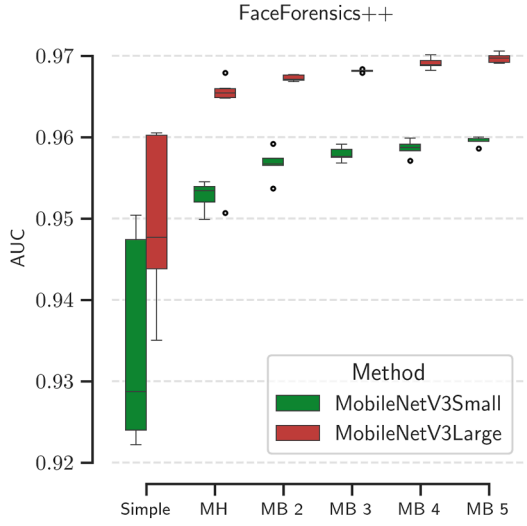


Fig. 10. Boxplots of AUC scores for baseline (Simple), Multi-Head (MH), and Multi-Branch (MB) models, evaluated on the FF++ dataset using MobileNetV3 family as backbone. Results are averaged over five independent training runs.

at a time, as if they were individual MH models, using a stratified bootstrap resampling of the training set. The stratification is applied only with respect to the binary class label (real vs. fake), while no constraint is imposed on the manipulation type distribution, which may vary across branches depending on the random sampling. Next, we concatenate the heads as described in Section 3.3 and train only the newly added fusion layers, while freezing all branch parameters (i.e., the backbone and the per-branch heads up to the penultimate dense layers). Batch Normalization layers are kept frozen by setting them as non-trainable, so that neither their affine parameters nor their running statistics are updated during fine-tuning. In the last step, we perform end-to-end fine-tuning by unfreezing the branch weights while keeping Batch Normalization layers frozen, using a very low learning rate (0.00001). In principle, identifying a strictly optimal configuration would require a grid search over several training hyperparameters. In this study, the emphasis is on comparing architectural variants under a common training protocol rather than on exhaustive hyperparameter optimisation. We therefore adopt a single set of standard hyperparameters per dataset, using *Adam* as optimizer with a learning rate of 0.001 for both datasets, selected from a small set of candidate values in preliminary experiments and then kept fixed for all architectures. We apply a batch size of 32, and we implemented a plateau-based decay schedule, monitoring the validation loss at each epoch. The learning rate is reduced by a factor of 0.1 whenever the validation loss fails to improve. To mitigate overfitting and optimize computational resources, we employed an early stopping mechanism with a patience of 3 epochs. Furthermore, we utilized model checkpointing to save only the model weights corresponding to the lowest validation loss achieved during training. All experiments were conducted on an Ubuntu 22.04.2 LTS using Keras and Tensorflow in a Python environment. The GPU used is an NVIDIA RTX A6000 48 GB.

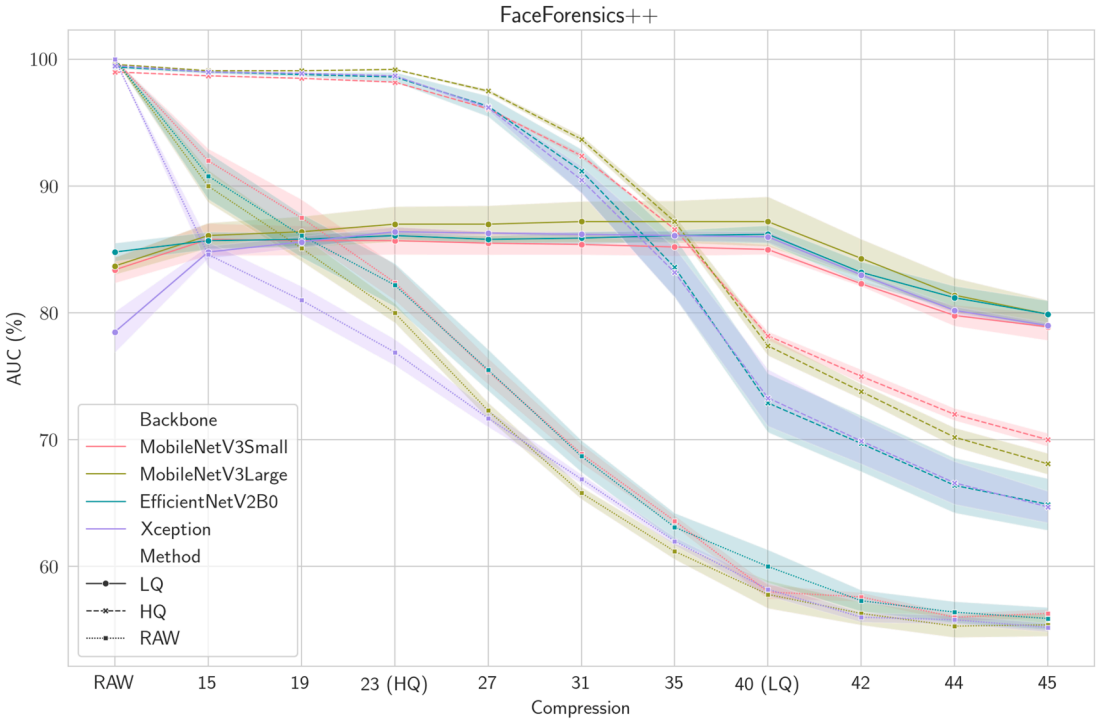
4.4. Results

A comprehensive set of experiments evaluates the proposed framework under different compression conditions and architectural configurations. Results are reported in terms of frame-level accuracy (ACC), area under the ROC curve (AUC), and video-level accuracy obtained through majority voting over frame predictions.

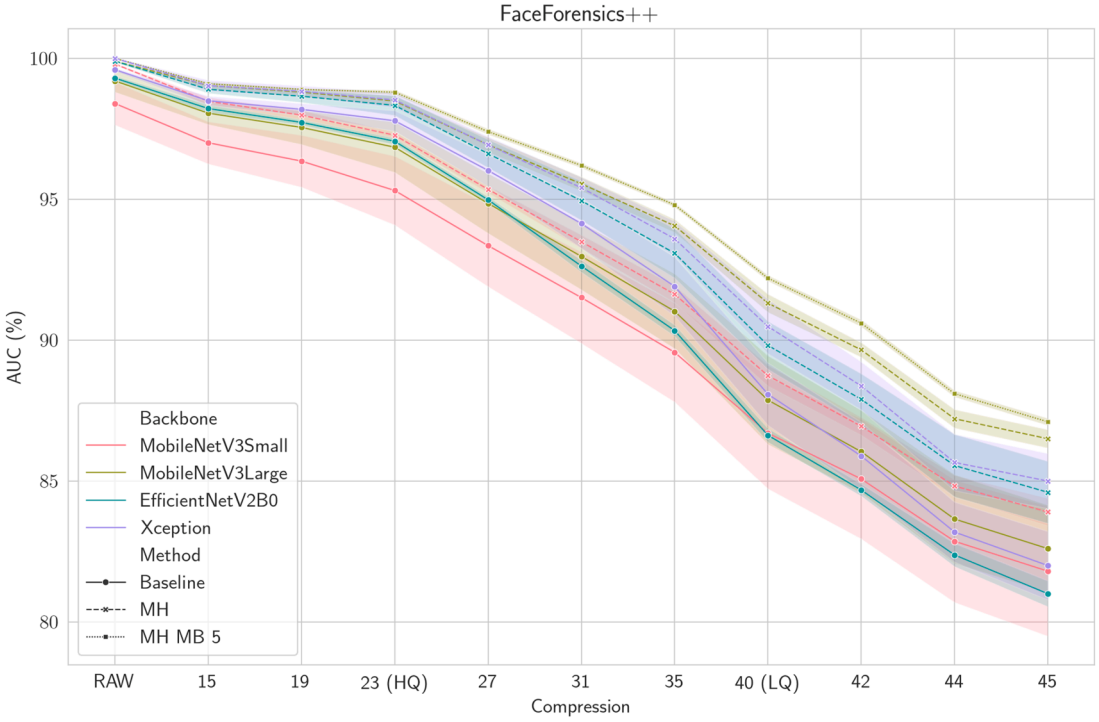
To rigorously assess robustness against stochastic optimization dynamics, all results are averaged over five independent runs. In these experiments, the training data and backbone initialization (ImageNet pre-trained) remained constant. Variability was introduced exclusively by varying the random seed, which determined the initialization of the custom MH layers, the data shuffling order, and the dropout patterns. In addition to reporting mean values and standard deviations, we further assess the statistical significance of observed differences between model variants using the Wilcoxon signed-rank test (Demšar, 2006; Wilcoxon, 1945). For each pairwise comparison, we report the p-value alongside a complementary effect size measure: the Rank-Biserial Correlation (r_{rb}) (Kerby, 2014), which ensures methodological consistency with the non-parametric hypothesis test. All statistical comparisons were performed across multiple seeds to account for randomness in the training process.

4.4.1. Impact of unseen compressed data on model performance

As reported in previous sections, the challenge of detecting deepfakes in compressed environments is well-documented. Existing research has demonstrated that the performance of deepfake detection models significantly degrades when dealing with lossy compression (Rossler et al., 2019). To establish a clear starting point for our work, we employed baseline models to quantify the impact of compression within our experimental setup, thereby reinforcing the severity of this issue and providing a foundation for demonstrating the improvements achieved by our proposed framework in subsequent experiments. Specifically, we trained all the selected baseline models on frames extracted from videos with



(a)



(b)

Fig. 11. AUC scores across varying compression levels on the FaceForensics++ dataset. Each line represents a different training setup: (a) models trained on a single compression level (RAW, HQ, LQ); (b) models trained using the proposed frame-centric strategy. Shaded areas denote the standard deviation computed over five training runs.

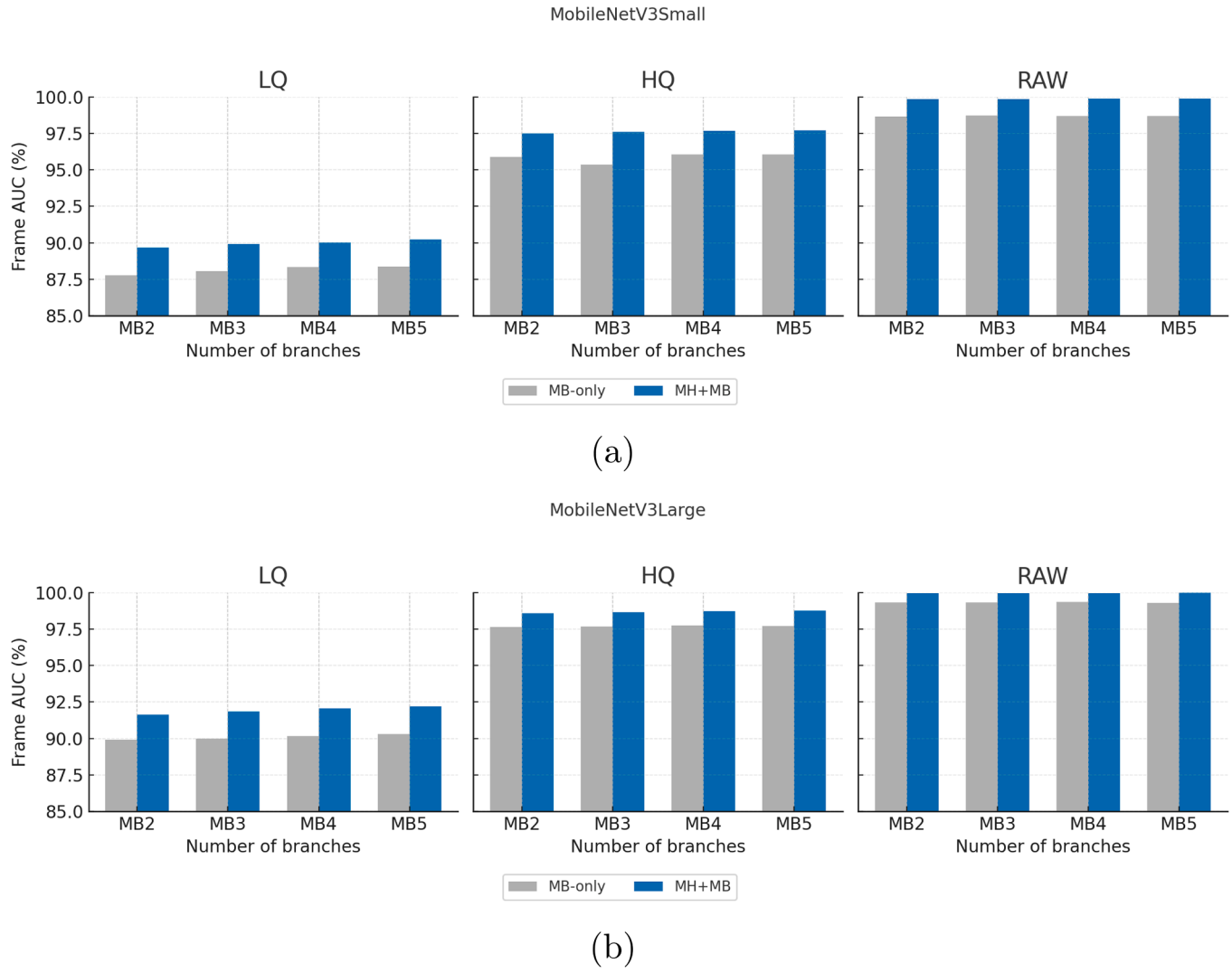


Fig. 12. Frame-level AUC on FaceForensics++ for Multi-Branch models without the compression-regression head (MB-only) and for the full MH+MB architecture. Results are shown for (a) MobileNetV3Small and (b) MobileNetV3Large backbones.

three different compression levels (RAW, HQ, and LQ), evaluating their performance across all these levels. However, for the sake of space and to provide a focused illustration of the results, we present the detailed findings for MobileNetV3Small in Table 1, though they are representative of the trends observed across the other baseline architectures considered in our analysis. The results reveal a significant drop in accuracy when the model is tested on compression levels not seen during training. In particular, models trained on RAW frames, both from the FaceForensics++ and Celeb-DF (v2) datasets, exhibited the most substantial decrease in performance when evaluated on LQ frames, as evidenced by a drop in video accuracy from 99.86% to 34.57% on the FaceForensics++ dataset, with similar trends observed on Celeb-DF (v2). Conversely, models trained on HQ frames, despite experiencing a noticeable decline in accuracy when tested on LQ frames, maintained reasonable performance on RAW frames. Training with LQ frames demonstrated the most stability across various compression levels, yet made it difficult to achieve competitive performance on higher-quality frames.

However, the results obtained on the Celeb-DF (v2) dataset present a different behaviour. Although we replicated the same compression levels (LQ, HQ, and RAW) as in FF++, the performance drop from RAW to HQ observed before is not evident in this case. Specifically, the model trained on RAW frames did not show a significant drop in accuracy when tested on HQ frames, suggesting that the original videos in the Celeb-DF (v2) dataset may have already undergone some degree of compression before we applied our artificial levels.

To corroborate this observation, we analyzed the video bitrate as a function of the CRF for both datasets, as shown in Fig. 7a. The bitrate, which measures the amount of data encoded per second of video, provides insight into the compression level applied; lower bitrates generally indicate higher compression levels. The FF++ data exhibits a typical pattern, with the video bitrate decreasing steadily as the CRF increases, reflecting the effects of increasing compression. In contrast, the starting quality of Celeb-DF (v2) aligns more closely with a CRF value between 15 and 20, which is much nearer to the HQ setting ($c = 23$) rather than to the RAW setting (labeled as $c = 0$). Consequently, the curve initially rises as CRF increases from 0, reflecting an increase in bitrate due to the minimal compression applied before aligning with the expected compression behavior in the later stages of the curve. Fig. 7b further highlights the lower starting quality of Celeb-DF (v2) with a clear distinction in the raw video bitrate distributions between the two datasets.

4.4.2. *Data augmentation strategies comparison*

Compression-aware augmentation is a key component of our framework, and two sampling strategies were defined to determine how compression levels are assigned during training: the *video-centric* (vc) and

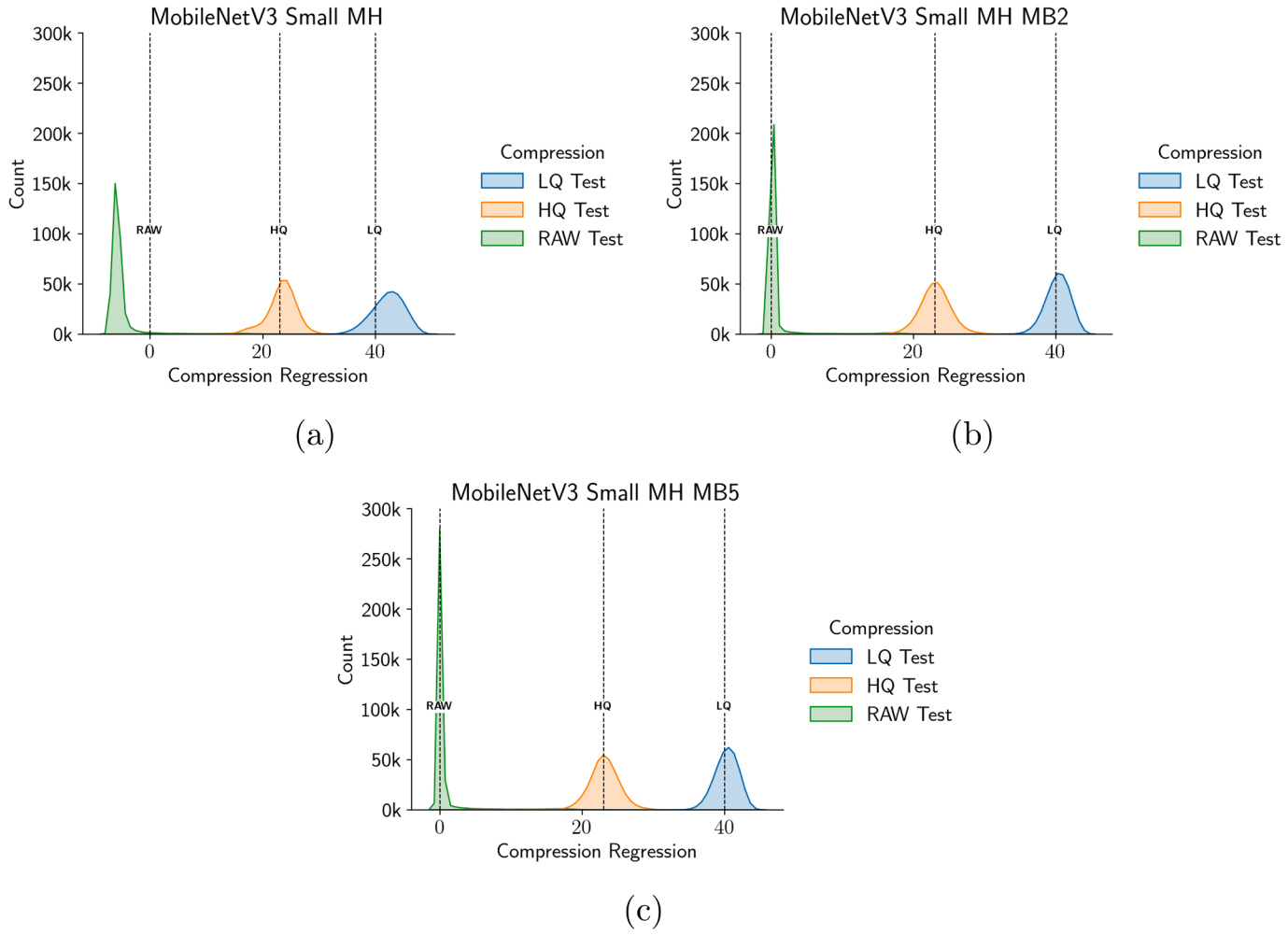


Fig. 13. Compression regression results for MobileNetV3 Small across different model configurations. (a) Single branch (MH), (b) Two branches (MB2), and (c) Six branches (MB5).

Table 1
Performance of the MobileNetV3Small model trained on different compression levels (LQ, HQ, and RAW) and evaluated on FaceForensics++ and Celeb-DF (v2) datasets. Best results are highlighted in bold.

	FaceForensics++ train			Celeb-DF (v2) train		
	LQ	HQ	RAW	LQ	HQ	RAW
	Test LQ					
FRAME ACC (%)	**82.68 ± 1.29**	63.90 ± 1.66	37.28 ± 1.56	**83.08 ± 0.35**	63.23 ± 1.94	53.81 ± 3.23
AUC (%)	**84.97 ± 0.37**	78.22 ± 0.33	57.95 ± 0.54	**91.00 ± 0.77**	84.86 ± 0.57	79.65 ± 1.00
VIDEO ACC (%)	**86.11 ± 0.41**	66.76 ± 3.24	34.57 ± 1.26	**86.42 ± 1.46**	62.48 ± 2.67	51.16 ± 4.38
	Test HQ					
FRAME ACC (%)	82.33 ± 1.09	**94.33 ± 0.04**	39.22 ± 1.39	87.94 ± 0.75	**95.60 ± 0.68**	95.15 ± 0.50
AUC (%)	85.71 ± 1.09	**98.25 ± 0.04**	82.36 ± 1.39	95.29 ± 0.32	**98.91 ± 0.18**	98.85 ± 0.25
VIDEO ACC (%)	85.01 ± 2.72	**96.85 ± 0.12**	37.92 ± 1.20	90.99 ± 0.95	98.07 ± 0.42	**98.13 ± 0.55**
	Test RAW					
FRAME ACC (%)	80.34 ± 3.18	96.32 ± 0.55	**99.50 ± 0.04**	88.22 ± 0.83	96.18 ± 0.67	**96.64 ± 0.60**
AUC (%)	83.41 ± 3.18	99.04 ± 0.55	**99.98 ± 0.04**	95.50 ± 0.32	99.11 ± 0.15	**99.39 ± 0.13**
VIDEO ACC (%)	83.09 ± 3.24	97.33 ± 0.65	**99.86 ± 0.00**	91.18 ± 0.96	98.52 ± 0.24	**98.65 ± 0.54**

the *frame-centric* (fc) approaches. To assess the impact of these strategies, we conducted a comparative evaluation across the four backbone architectures employed throughout the paper. We trained two versions of each model on the FF++ dataset using identical settings, differing only in the sampling strategy used to generate the augmented training set. The results are summarized in Table 2. A consistent trend emerges in favor of the frame-centric approach concerning AUC, indicating a more effective encoding of compression variability during training. Despite minor variations in frame and video-level accuracy, the improvement is observed across all architectures, suggesting that exposure to a broader and more uniform compression spectrum results in more stable performance across decision thresholds. Based on this empirical evidence, the

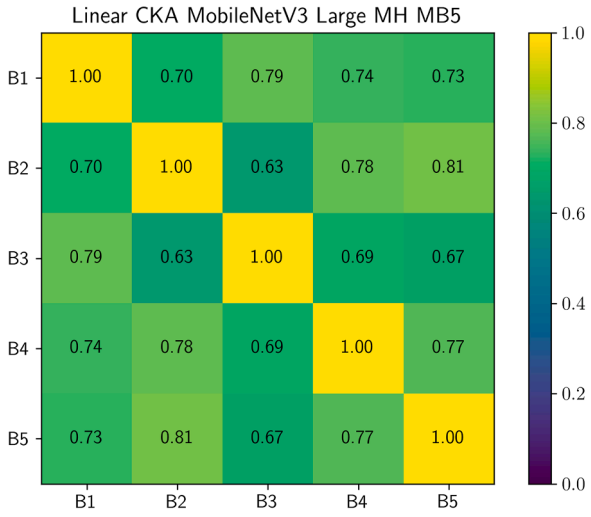


Fig. 14. Linear CKA similarity matrices for the penultimate dense-layer representations of the classification head in the five branches of the MobileNetV3-Small (a) and MobileNetV3-Large (b) Multi-Branch Networks, computed on the FaceForensics++ validation split.

Table 2
Comparison between video-centric (vc) and frame-centric (fc) augmentation strategies on FaceForensics++ dataset. Best results are highlighted in bold.

MODEL	FRAME ACC.	AUC	VIDEO ACC.
MobileNetV3Small (vc)	87.19	92.61	89.91
MobileNetV3Small (fc)	**88.35**	**93.12**	**91.15**
MobileNetV3Large (vc)	**90.74**	94.48	**93.43**
MobileNetV3Large (fc)	90.48	**94.76**	93.06
EfficientNetV2B0 (vc)	88.49	93.88	**92.11**
EfficientNetV2B0 (fc)	**89.02**	**94.30**	91.95
Xception (vc)	89.21	94.08	91.87
Xception (fc)	**91.07**	**95.20**	**93.98**

frame-centric configuration is adopted as the standard setting in all subsequent experiments presented in the paper.

4.4.3. Evaluation of the multi-head network architecture

Building on the analysis of baseline model limitations in Section 4.4.1, we now assess the impact of our proposed augmentation technique and the Multi-Head Network architecture on enhancing the robustness and accuracy of deepfake detection models. The results of our experiments are presented in Tables 3–5.

Firstly, compared to the results previously discussed (see Table 1), where models were trained on isolated compression levels, the models trained with our compression-aware augmentation, labeled as *Simple* in the Tables, demonstrate superior adaptability across quality settings. In contrast to models specialized on LQ, HQ, or RAW videos, which tend to overfit to their respective domains and perform poorly on other compression levels, the *Simple* models benefit from exposure to a broader compression spectrum, resulting in higher accuracy and more balanced performance. Notably, this robustness is consistently observed across both the FaceForensics++ and Celeb-DF (v2) datasets, although the accuracy metrics for Celeb-DF (v2) are slightly skewed due to the dataset's inherent lower quality. As discussed in the previous section, the training labels were based on FF++ compression levels, but Celeb-DF (v2) videos probably started with a lower quality, which may have affected the results. Despite this, we maintained uniform labeling across datasets to ensure uniformity in our evaluation.

Secondly, these results highlight that our augmentation technique alone is not sufficient. A specialized architecture like the Multi-Head Network is required to fully leverage the injected compression information. The addition of the MHN provides a significant boost in performance compared to the *Simple* model that only uses augmentation. Furthermore, the MHN helps stabilize the training process, as indicated by the lower standard deviations across multiple training sessions, providing more consistent results (Fig. 8). Class-wise F1-scores follow the same trend: MH architectures reduce the tendency to misclassify genuine videos, increasing the F1-score for the real class without sacrificing detection performance on manipulated content (see Tables 4 and 6). To provide a clearer view of this improvement, Fig. 9 illustrates the mean video accuracy across the three test sets for each configuration. It is easy to see that MHN-enhanced models consistently outperform both *Simple* models and single-compression-trained baselines, confirming the effectiveness of combining compression-aware augmentation with a dedicated architectural design.

A representative case is MobileNetV3Small with MHN, which not only outperforms the *Simple* configurations of its larger baseline counterpart (MobileNetV3Large) but also approaches the performance of more complex and parameter-heavy models such as EfficientNet and Xception. Such findings suggest that smaller models can challenge the conventional approach of increasing model size to achieve better performance when strategically enhanced. The advantage of MHN is also evident across compression settings. As discussed in Section 4.4.1, models trained exclusively on RAW suffer substantial degradation when evaluated on compressed data, while those trained on HQ or LQ generalize more effectively, especially on Celeb-DF (v2), where native compression artifacts may act as a form of implicit regularization.

To assess the robustness and statistical significance of the observed improvements, we performed a paired comparison between each baseline model and its MHN-enhanced counterpart using the Wilcoxon signed-rank test. All pairwise comparisons across both datasets yielded statistically significant AUC differences ($p = .0312$), i.e., below the commonly adopted threshold $\alpha = 0.05$ used to reject the null hypothesis of no difference (Benjamin et al., 2018). The value $p = .0312$ corresponds to the minimum attainable two-sided p-value for $N = 5$ paired runs, which reflects that the direction of the difference is consistent across runs. To ensure methodological consistency with this non-parametric framework, we quantify the strength of the paired differences using the Rank-Biserial Correlation (r_{rb}), which ranges in $[-1, 1]$ and measures the degree to which one condition systematically yields larger values than the other. In our comparisons, $r_{rb} = 1.0$ throughout, indicating that the MHN-enhanced models outperform the corresponding baselines in every run; therefore, the effect is fully directional across repetitions, independently of its absolute magnitude. The next section investigates whether architectural gains can be further amplified by exploiting structured parallelism, as opposed to increasing model depth or width.

4.4.4. Evaluation of the multi-branch network architecture

The analysis of the Multi-Branch Network (MBN) builds on the concept of efficient parameter utilization and investigates whether the architecture can serve as a more effective alternative to increasing model depth by exploiting the parallelism of multiple branches. For the sake of space, we have limited our analysis to the FaceForensics++ dataset and selected the MobileNetV3 family due to its balance between performance and efficiency. As shown in Table 8, the adoption of additional branches leads to progressive improvements across all compression conditions, particularly in terms of AUC (Fig. 10). The mean metrics reported in Table 9 confirm that the gains observed on AUC are mirrored at the video level and across both classes. Frame and video accuracy increase when moving from the Simple configuration to MH and then to MH+MB, while the F1-score for real videos steadily improves and the F1-score for fake videos remains high and stable. Also in this case, we conducted a statistical analysis comparing each MBN variant to the baseline Simple model. All configurations achieve statistically significant differences (Wilcoxon signed-rank test, $p = .0312$), with max-

Table 3
Performance comparison between standard models and their enhanced versions with our Multi-Head Network (MHN) across different compression levels (LQ, HQ, and RAW) on the FaceForensics++ dataset. Best results are highlighted in bold.

	MobileNetV3Small Simple	MobileNetV3Small MH	MobileNetV3Large Simple	MobileNetV3Large MH
Million Params	0.94	0.98	3.00	3.12
	LQ test set			
FRAME ACC (%)	80.50 ± 2.38	**83.02 ± 0.66**	81.89 ± 2.95	**84.44 ± 0.85**
AUC (%)	86.70 ± 1.97	**88.74 ± 0.36**	87.87 ± 1.57	**90.68 ± 1.38**
VIDEO ACC (%)	84.50 ± 2.80	**87.68 ± 1.16**	87.08 ± 2.44	**90.03 ± 1.06**
	HQ test set			
FRAME ACC (%)	90.27 ± 1.99	**92.63 ± 0.30**	92.48 ± 1.17	**94.39 ± 0.93**
AUC (%)	95.31 ± 1.22	**97.27 ± 0.12**	96.85 ± 0.89	**98.33 ± 0.48**
VIDEO ACC (%)	92.98 ± 2.03	**95.44 ± 0.59**	95.05 ± 1.07	**96.36 ± 0.79**
	RAW test set			
FRAME ACC (%)	94.74 ± 1.43	**98.25 ± 0.13**	96.27 ± 1.03	**98.90 ± 0.21**
AUC (%)	98.35 ± 0.76	**99.81 ± 0.03**	99.15 ± 0.39	**99.91 ± 0.04**
VIDEO ACC (%)	96.07 ± 1.11	**98.62 ± 0.41**	97.19 ± 0.93	**99.11 ± 0.19**
	EfficientNetV2B0 Simple	EfficientNetV2B0 MH	Xception Simple	Xception MH
Million Params	5.92	6.08	20.86	21.12
	LQ test set			
FRAME ACC (%)	76.98 ± 2.35	**83.50 ± 1.97**	80.82 ± 2.89	**84.23 ± 2.14**
AUC (%)	86.62 ± **0.24**	**89.81** ± 0.85	88.08 ± 1.11	**90.48 ± 0.97**
VIDEO ACC (%)	81.55 ± 2.96	**89.91 ± 1.66**	86.71 ± 3.28	**89.11 ± 1.96**
	HQ test set			
FRAME ACC (%)	91.41 ± 0.91	**94.79 ± 0.71**	93.13 ± 1.95	**94.66 ± 0.70**
AUC (%)	97.06 ± **0.12**	**98.33** ± 0.35	97.79 ± 0.37	**98.52 ± 0.27**
VIDEO ACC (%)	94.38 ± 1.20	**97.02 ± 0.51**	95.64 ± 1.55	**96.45 ± 0.58**
	RAW test set			
FRAME ACC (%)	96.81 ± **0.21**	**98.99** ± 0.24	97.22 ± **0.24**	**99.45** ± 0.31
AUC (%)	99.33 ± 0.08	**99.93** ± 0.03	99.55 ± 0.06	**99.97 ± 0.02**
VIDEO ACC (%)	97.62 ± 0.31	**99.23** ± 0.17	97.91 ± **0.11**	**99.71** ± 0.31

Table 4
Mean performance comparison between standard models and their Multi-Head (MH) variants on the FaceForensics++ dataset. Results are averaged over five runs and reported in terms of frame accuracy, AUC, video accuracy, and class-wise F1-scores for real and fake samples.

MODELS	FRAME ACC	AUC	VIDEO ACC	F1 real	F1 fake
MobileNetV3Small - Simple	88.50	93.45	91.18	74.49	92.54
MobileNetV3Small - MH	91.30	95.27	93.92	80.52	94.40
MobileNetV3Large - Simple	90.67	94.94	93.23	78.65	94.00
MobileNetV3Large - MH	92.65	96.31	95.17	83.61	95.26
EfficientNetV2B0 - Simple	89.39	94.56	92.29	77.98	92.97
EfficientNetV2B0 - MH	92.50	96.00	95.42	83.33	95.15
Xception - Simple	90.39	95.14	93.42	79.64	93.68
Xception - MH	92.78	96.33	95.09	83.99	95.32

imum effect sizes across the board (Rank-Biserial Correlation (r_{rb}) equal to 1.0).

Since each additional branch introduces extra parameters and computations, these gains need to be interpreted in relation to the associated cost. To quantify the computational footprint, Table 7 reports the number of parameters and MFLOPs per frame for the same configurations. The Multi-Head extension leaves the footprint almost unchanged with respect to the corresponding baseline (for instance, MobileNetV3Small moves from 0.94M parameters and 109.80 MFLOPs to 0.98M parameters and 109.87 MFLOPs), whereas the cost of Multi-Branch variants grows approximately linearly with the number of branches. Configurations such as MB2 and MB3 already capture most of the observed accuracy gains while remaining substantially lighter than heavier backbones like EfficientNetV2B0 or Xception, which makes them natural candidates for high-throughput or latency-sensitive scenarios. Deeper ensembles such as MB4 and MB5 are instead more suitable for offline analysis or high-resource deployments. The ensemble effect introduced by bootstrap resampling also contributes to lower variance across training runs, reinforcing the view of MBN as a modular mechanism to tune the accuracy-complexity trade-off according to the requirements of the target application. As a matter of fact, from an application perspective, this trade-off aligns with several deployment scenarios. Lightweight MHN and shallow MBN configurations such as MB2 or MB3 can support client-side modules in social-media or messaging platforms, or be integrated into mobile forensic toolkits, where strict limits on model size and MFLOPs per frame constrain both memory footprint and inference latency. Deeper MBN variants remain more appropriate for back-end forensic analysis or large-scale offline screening of video archives, including security or surveillance footage, where higher computational budgets are available.

Table 5
Performance comparison between standard models and their enhanced versions with our Multi-Head Network (MHN) across different compression levels (LQ, HQ, and RAW) on the Celeb-DF (v2) dataset.

	MobileNetV3Small Simple	MobileNetV3Small MH	MobileNetV3Large Simple	MobileNetV3Large MH
Million Params	0.94	0.98	3.00	3.12
	LQ test set			
FRAME ACC (%)	82.55 ± 4.38	**86.64 ± 0.66**	83.62 ± 3.39	**87.48 ± 1.66**
AUC (%)	91.70 ± 0.81	**93.77 ± 0.53**	92.25 ± 1.30	**94.51 ± 0.92**
VIDEO ACC (%)	86.14 ± 5.27	**90.46 ± 0.23**	89.04 ± 3.61	**91.82 ± 1.03**
	HQ test set			
FRAME ACC (%)	91.79 ± 1.12	**94.92 ± 0.40**	94.23 ± 0.99	**96.16 ± 0.68**
AUC (%)	98.09 ± 0.23	**98.94 ± 0.15**	98.62 ± **0.26**	**99.11** ± 0.30
VIDEO ACC (%)	94.02 ± 1.71	**96.80 ± 0.60**	96.64 ± 0.58	**97.91 ± 0.37**
	RAW test set			
FRAME ACC (%)	92.70 ± 0.91	**96.31 ± 0.33**	94.89 ± 0.96	**97.16 ± 0.80**
AUC (%)	98.41 ± 0.24	**99.40 ± 0.08**	98.93 ± 0.20	**99.63 ± 0.09**
VIDEO ACC (%)	94.90 ± 1.34	**98.03 ± 0.28**	97.30 ± 0.91	**98.49 ± 0.86**
	EfficientNetV2B0 Simple	EfficientNetV2B0 MH	Xception Simple	Xception MH
Million Params	5.92	6.08	20.86	21.12
	LQ test set			
FRAME ACC (%)	**85.03** ± 1.03	84.78 ± **0.58**	85.37 ± 1.66	**86.44 ± 0.83**
AUC (%)	91.97 ± 0.42	**92.12 ± 0.16**	93.30 ± 0.55	**93.68 ± 0.39**
VIDEO ACC (%)	89.50 ± 0.50	**90.15 ± 0.44**	90.69 ± 2.06	**91.16 ± 0.83**
	HQ test set			
FRAME ACC (%)	94.13 ± **0.30**	**95.44** ± 0.34	95.22 ± 0.62	**95.81 ± 0.43**
AUC (%)	98.70 ± 0.18	**99.16 ± 0.09**	**99.30 ± 0.14**	99.24 ± 0.25
VIDEO ACC (%)	96.53 ± 0.36	**97.95 ± 0.34**	97.11 ± 0.66	**97.80 ± 0.40**
	RAW test set			
FRAME ACC (%)	94.75 ± **0.30**	**96.36** ± 0.41	96.00 ± **0.46**	**96.90** ± 0.47
AUC (%)	98.96 ± 0.17	**99.52 ± 0.04**	99.47 ± **0.10**	**99.60** ± 0.12
VIDEO ACC (%)	96.95 ± 0.33	**98.22 ± 0.31**	97.72 ± 0.56	**98.38 ± 0.38**

Table 6
Mean performance comparison between standard models and their Multi-Head (MH) variants on the Celeb-DF (v2) dataset. Results are averaged over five runs and reported in terms of frame accuracy, AUC, video accuracy, and class-wise F1-scores for real and fake samples.

MODELS	FRAME ACC	AUC	VIDEO ACC	F1 real	F1 fake
MobileNetV3Small - Simple	89.01	96.07	91.69	84.09	91.48
MobileNetV3Small - MH	92.62	97.37	95.10	88.62	94.54
MobileNetV3Large - Simple	90.91	96.60	94.33	87.41	92.86
MobileNetV3Large - MH	93.60	97.75	96.07	90.33	95.20
EfficientNetV2B0 - Simple	91.30	96.55	94.32	87.09	93.43
EfficientNetV2B0 - MH	92.19	96.93	95.44	88.65	94.04
Xception - Simple	92.20	97.36	95.17	88.74	94.01
Xception - MH	93.05	97.51	95.78	89.69	94.74

4.4.5. Robustness and diversity analysis

Branch ensembling and compression-aware supervision represent the two key mechanisms explored in the proposed architecture. In order to clarify their respective impact, we trained Multi-Branch models without the auxiliary regression head for MobileNetV3Small and MobileNetV3Large, on FaceForensics++. In these MB-only variants, each branch outputs only a classification score, whereas branch diversity still derives from bootstrap resampling. Fig. 12 reports the mean frame-level AUC over LQ, HQ, and RAW for MB-only and MH+MB configurations as a function of the number of branches. Performance improves with the number of branches, which confirms that ensembling multiple branches is beneficial even in the absence of explicit compression-awareness. However, MB-only curves remain consistently below the corresponding MH+MB curves at comparable parameter counts. The gap is more evident on LQ and HQ, where MH+MB improves AUC by approximately 1-2 percentage points over MB-only, while differences on RAW are smaller but still positive.

The auxiliary regression task provides complementary evidence about the impact of compression-aware supervision. Fig. 13 reports the distributions of predicted CRF values for RAW, HQ, and LQ test clips on FaceForensics++ for the MobileNetV3Small backbone under MH, MB2, and MB6 configurations. The MH baseline already separates the three compression regimes, yet the predicted distributions remain relatively spread and slightly biased with respect to the ground-truth CRF values. The MB2 configuration produces narrow peaks centred around the target compression levels (approximately 0, 23, and 40), while the distributions for MB5 are almost indistinguishable from those of MB2. The improvement in compression estimation from MH to MB2 aligns

Table 7
Number of parameters and computational cost (per frame) for baseline, MH, and MH + MB configurations. MFLOPs denote the approximate number of floating-point operations per frame.

MODELS	Million Params	MFLOPs
MobileNetV3Small - Simple	0.94	109.80
MobileNetV3Small - MH	0.98	109.87
MobileNetV3Small - MH MB2	1.95	219.74
MobileNetV3Small - MH MB3	2.93	329.61
MobileNetV3Small - MH MB4	3.91	439.48
MobileNetV3Small - MH MB5	4.88	549.35
MobileNetV3Large - Simple	3.00	428.17
MobileNetV3Large - MH	3.12	428.41
MobileNetV3Large - MH MB2	6.24	856.83
MobileNetV3Large - MH MB3	9.36	1285.24
MobileNetV3Large - MH MB4	12.48	1713.65
MobileNetV3Large - MH MB5	15.60	2142.07
EfficientNetV2B0 - Simple	5.92	1433.63
EfficientNetV2B0 - MH	6.08	1433.95
Xception - Simple	20.86	16710.72
Xception - MH	21.12	16711.24

with the AUC gains observed on compressed data, whereas the marginal differences between MB2 and MB5 mirror the saturation in detection performance.

An additional analysis examines feature-level similarity between branches by using centered kernel alignment (CKA) (Kornblith et al., 2019). CKA measures how much structure is shared between two sets of activations and returns a normalised similarity score between 0 and 1; linear CKA, as used here, operates on activation matrices through dot products and Frobenius norms and remains invariant to isotropic rescaling and orthogonal transformations of the features. The analysis considers the activations of the final dense layer for the five branches of both the MobileNetV3-Small and MobileNetV3-Large MBNs, computed on a subset of LQ and HQ clips from the FaceForensics + + dataset. Fig. 14 reports the corresponding linear CKA matrices. Off-diagonal scores typically fall in the range 0.78–0.87 for both backbones, which indicates that branches learn strongly related, yet not identical, representations. The moderate but consistent AUC gains observed when moving from a single MHN to MB2 and MB3 at fixed backbone family and compression regime align with this picture of shared compression-aware features combined with controlled branch diversity.

As a complementary robustness check, the analysis is extended to a cross-dataset setting between FaceForensics + + and Celeb-DF (v2). Table 10 reports the frame-level AUC for MH baselines and their Multi-Branch extensions (MH MB2–MB5), trained on FF + + and evaluated on Celeb-DF (v2) without any form of fine-tuning or domain adaptation. A marked performance drop emerges with respect to the intra-dataset results discussed in previous Sections, which reflects the substantial distribution mismatch between the two datasets in terms of manipulation pipelines, source content, and compression chains.

Within the MobileNetV3 families, the introduction of additional branches does not fundamentally alter this picture. For MobileNetV3Small, MH and MB variants cluster around 59% AUC, with differences below one percentage point. For MobileNetV3Large, deeper ensembles provide only a modest gain over the single-branch MH baseline (57.7%). EfficientNetV2B0 MH attains the highest cross-dataset AUC (61.5%), yet the margin over the MobileNet variants remains limited. Results suggest that compression-aware supervision and branch ensembling improve robustness to compression variations within a given dataset, but do not by themselves compensate for broader dataset shifts. A more domain-agnostic treatment of compression, possibly combined with multi-dataset training or explicit domain adaptation, lies outside the scope of the present work and is left for future research.

4.4.6. Comparison with state-of-the-art methods

To evaluate the competitiveness of our approach, we selected the MobileNetV3-Large with five branches (MH MB5), identified as the best performer in the previous section, for comparison against state-of-the-art (SOTA) deepfake detection methods. The evaluation was conducted on the FaceForensics + + dataset under both LQ and HQ compression conditions (Table 11).

Absolute state-of-the-art performance is not the primary target of the study. The comparison in Table 11 shows that a lightweight architecture such as MobileNetV3, when enhanced with the proposed Multi-Head and Multi-Branch framework, attains accuracy that is consistently close to that of more complex and computationally demanding models such as Xception and EfficientNet-B4. On LQ and HQ, the MH MB5 configuration reaches an AUC of 92.18% and 98.76%, respectively, which lies within the range reported by recent top-performing detectors, despite the substantially lower parameter count and per-frame MFLOPs reported in Table 7. Some heavier baselines retain a slight advantage in specific settings, yet the gains are marginal when compared to the increase in model size and computational cost. The main value of the proposed framework lies in this trade-off: competitive, near-SOTA detection performance under compression is achieved with a compact architecture that remains explicitly scalable through the number of branches, so that capacity can be tuned to the available resources and latency constraints.

Whilst most existing literature relies on models trained exclusively on the two discrete compression levels (C23 and C40 introduced by FF + +), without worrying about what happens in the presence of different compressions, our approach transcends this limitation. Although evaluated on the same benchmarks for fair comparison, our methodology stems from a more general framework capable of modeling a continuous spectrum of video compression artifacts through dynamic MPEG augmentation. As shown in Fig. 11, ignoring the compression distribution can lead to significant drops in performance. Specifically, a network trained on heavily compressed data (LQ) appears more robust across different compression levels, although it achieves lower overall accuracy. In contrast, training with lightly compressed data (HQ) results in higher performance on similar data but causes the model to collapse when tested on videos with stronger compression. Thus, our performance on C23/C40 does not reflect explicit optimization for these levels, but rather intrinsic robustness achieved by exposing the model to a broader, more realistic range of compression conditions.

While most existing approaches evaluate detection models only at the two fixed compression settings provided by FaceForensics + + (LQ and HQ), the proposed framework is designed to operate across the whole compression spectrum. Standard LQ/HQ results are still reported for comparison, but compression is modelled as a continuous variable rather than as a pair of discrete operating points. Fig. 11 summarises how AUC evolves as the CRF increases on FaceForensics + +. Panel (a) shows that disregarding the underlying compression distribution leads to marked failure regimes: models trained exclusively on RAW or HQ data retain high performance close to their training conditions, yet exhibit abrupt drops once the CRF enters the high-compression range, whereas models specialised on LQ compression degrade more smoothly but with a lower peak AUC. Panel (b) reports the behaviour of models trained with the proposed frame-centric strategy, including MH and MH + MB variants, using a different vertical scale to better highlight relative differences. In this case, the curves decline more gradually, and the onset of severe degradation is shifted towards more extreme compression levels, indicating that the combination of dynamic MPEG-based augmentation and the modular design delays the compression-induced failure modes.

Finally, it is worth emphasizing that the proposed framework is inherently backbone-agnostic. Its components can be readily integrated into a wide range of architectures and deployment scenarios, offering a general-purpose enhancement strategy for deepfake detection pipelines. Configurations with a small number of branches provide lightweight options for latency-constrained settings, whereas deeper ensembles are

Table 8
Performance comparison between standard models and their enhanced versions with our Multi-Head (MHN) and Multi-Branch Network (MBN) across different compression levels (LQ, HQ, and RAW) on the FaceForensics++ dataset.

	MobileNetV3 Small					
	Simple	MH	MH MB2	MH MB3	MH MB4	MH MB5
Million Params	0.94	0.98	1.95	2.93	3.91	4.88
	LQ test set					
FRAME ACC (%)	80.50 ± 2.38	83.02 ± 0.66	84.74 ± **0.22**	85.00 ± 0.28	85.31 ± 0.39	**85.47** ± 0.35
AUC (%)	86.70 ± 1.97	88.74 ± 0.36	89.66 ± 0.33	89.90 ± 0.13	90.02 ± 0.12	**90.24** ± **0.08**
VIDEO ACC (%)	84.50 ± 2.80	87.68 ± 1.16	88.85 ± 0.65	89.37 ± 0.70	**89.40** ± **0.22**	89.17 ± 0.27
	HQ test set					
FRAME ACC (%)	90.27 ± 1.99	92.63 ± 0.30	93.51 ± **0.13**	93.70 ± 0.15	93.80 ± 0.17	**93.84** ± 0.14
AUC (%)	95.31 ± 1.22	97.27 ± **0.12**	97.51 ± 0.20	97.61 ± 0.19	97.68 ± 0.21	**97.72** ± **0.12**
VIDEO ACC (%)	92.98 ± 2.03	95.44 ± 0.59	96.22 ± 0.25	**96.30** ± 0.17	96.22 ± **0.15**	96.28 ± 0.20
	RAW test set					
FRAME ACC (%)	94.74 ± 1.43	98.25 ± 0.13	98.43 ± **0.12**	98.54 ± 0.19	98.62 ± 0.19	**98.65** ± 0.19
AUC (%)	98.35 ± 0.76	99.81 ± 0.03	99.84 ± 0.05	99.86 ± 0.04	99.88 ± **0.02**	**99.89** ± 0.03
VIDEO ACC (%)	96.07 ± 1.11	98.62 ± 0.41	98.91 ± **0.21**	98.85 ± 0.33	99.05 ± 0.33	**99.08** ± 0.39
	MobileNetV3 Large					
	Simple	MH	MH MB2	MH MB3	MH MB4	MH MB5
Million Params	3.00	3.12	6.24	9.36	12.48	15.60
	LQ test set					
FRAME ACC (%)	81.89 ± 2.95	84.44 ± 0.85	86.74 ± 0.54	86.92 ± 0.49	87.19 ± **0.19**	**87.47** ± 0.20
AUC (%)	87.87 ± 1.57	90.68 ± 1.38	91.64 ± 0.11	91.85 ± **0.09**	92.04 ± **0.09**	**92.18** ± 0.10
VIDEO ACC (%)	87.08 ± 2.44	90.03 ± 1.06	91.14 ± 0.64	91.23 ± 0.78	91.26 ± 0.66	**91.52** ± **0.59**
	HQ test set					
FRAME ACC (%)	92.48 ± 1.17	94.39 ± 0.93	95.47 ± **0.09**	95.48 ± 0.13	95.56 ± 0.12	**95.65** ± 0.10
AUC (%)	96.85 ± 0.89	98.33 ± 0.48	98.57 ± 0.12	98.64 ± 0.09	98.72 ± 0.11	**98.76** ± **0.08**
VIDEO ACC (%)	95.05 ± 1.07	96.36 ± 0.79	96.87 ± 0.17	96.85 ± 0.13	**96.96** ± **0.10**	96.88 ± 0.11
	RAW test set					
FRAME ACC (%)	96.27 ± 1.03	98.90 ± 0.21	99.15 ± 0.11	99.25 ± 0.11	99.27 ± 0.09	**99.36** ± **0.07**
AUC (%)	99.15 ± 0.39	99.91 ± 0.04	99.96 ± **0.01**	99.96 ± **0.01**	99.96 ± **0.01**	**99.97** ± **0.01**
VIDEO ACC (%)	97.19 ± 0.93	99.10 ± 0.19	99.40 ± 0.17	99.43 ± 0.13	99.40 ± 0.17	**99.57** ± **0.09**

Table 9
Mean performance comparison between MobileNetV3Small and MobileNetV3Large with and without the Multi-Head (MH) and Multi-Branch Network (MBN) extensions on the FaceForensics++ dataset. Results are averaged over five runs and reported in terms of frame accuracy, AUC, video accuracy, and class-wise F1-scores for real and fake samples.

	Models	Test Set (FaceForensics++ Dataset)				
		FRAME ACC	AUC	VIDEO ACC	F1 real	F1 fake
MNetV3 Small	Simple	88.35	93.12	91.15	73.46	92.48
	MH	91.02	95.36	93.86	80.58	94.13
	MH MB2	92.22	95.66	94.46	81.66	95.07
	MH MB3	92.69	95.91	94.99	82.24	95.39
	MH MB4	92.64	95.95	94.92	82.34	95.35
	MH MB5	**92.70**	**96.01**	**95.00**	**82.37**	**95.40**
MNetV3 Large	Simple	90.48	94.76	93.06	78.09	93.87
	MH	92.99	96.45	95.51	84.27	95.49
	MH MB2	93.78	96.68	95.89	85.24	96.06
	MH MB3	93.96	96.77	96.02	85.51	96.18
	MH MB4	94.03	96.80	96.07	85.53	96.24
	MH MB5	**94.18**	**96.93**	**96.08**	**85.83**	**96.34**

Table 10
Cross-dataset evaluation from FaceForensics++ (source) to Celeb-DF (v2) (target) for Multi-Head (MH) and Multi-Branch (MH MB) configurations, in terms of mean AUC over five runs.

Models (FF++ → Celeb-DF (v2))	AUC
MobileNetV3Small - MH	59.39
MobileNetV3Small MH MB2	59.52
MobileNetV3Small MH MB3	59.41
MobileNetV3Small MH MB4	59.18
MobileNetV3Small MH MB5	59.06
MobileNetV3Large - MH	57.69
MobileNetV3Large MH MB2	57.63
MobileNetV3Large MH MB3	57.84
MobileNetV3Large MH MB4	58.18
MobileNetV3Large MH MB5	58.32
EfficientNetV2B0 - MH	61.52
Xception - MH	57.78

Table 11
Performance comparison of various state-of-the-art deepfake detection approaches on the FaceForensics++ dataset, evaluated on LQ and HQ test sets. The table reports the area under the ROC curve (AUC %). Each approach is listed with its respective backbone architecture in parentheses.

Method	AUC (%)	
	LQ	HQ
Xception (Rossler et al., 2019)	89.30	96.30
F^3-Net (Qian et al., 2020) (Xception)	93.30	98.10
SPSL (Liu et al., 2021) (Xception)	82.82	95.32
FDFL (Li et al., 2021) (Xception)	92.40	99.30
MAT (Zhao et al., 2021) (EfficientNet-B4)	90.40	99.29
SIA (Sun et al., 2022) (EfficientNet-B4)	93.45	99.35
RECCE (Cao et al., 2022) (Xception)	95.02	99.32
BOF (Miao et al., 2021)(Vision Transformer Dosovitskiy et al., 2020)	91.61	99.36
LRL (Chen et al., 2021) (Custom)	95.21	99.46
MRL (Yang et al., 2023) (MC3-18 (Tran et al., 2018))	96.18	98.27
SRDF (Li et al., 2023) (EfficientNet-B4)	94.09	99.69
HiFE (Gao et al., 2024) (Xception)	79.30	97.90
3D-ST (Chen et al., 2024) (Custom)	97.98	98.82
FAMM (Liao et al., 2023) (Custom)	96.98	97.98
Our (MobileNetV3Large - MH MB5)	92.18	98.76

more suitable when higher capacity is acceptable. This adaptability makes it a promising foundation for future research aimed at maintaining high detection performance under realistic, resource-constrained operating conditions.

5. Conclusions

In this paper, we have introduced a modular approach to deepfake detection that explicitly addresses the degradation effects introduced by video compression, a major challenge in real-world applications. The method integrates a Multi-Head Network and a Multi-Branch Network architecture with a compression-aware data augmentation strategy, designed to improve performance across a range of compression levels. Experimental results on the FaceForensics++ and Celeb-DF (v2) datasets, which remain the most widely used benchmarks for video deepfake detection and provide controlled H.264-based compression regimes and manipulation families, confirm the effectiveness of the proposed framework. The Multi-Head Network achieves consistent gains in detection accuracy, particularly under low-quality conditions, while requiring only a lightweight extension of standard backbones. The Multi-Branch Network further shows that performance can be improved by adding parallel branches and is explicitly designed to expose an accuracy-complexity trade-off: configurations with a small number of branches already capture most of the observed gains and form natural candidates for latency-sensitive applications, whereas deeper ensembles are more appropriate for offline analysis or high-resource deployments.

Cross-dataset experiments highlight that the proposed strategy improves robustness across compression levels within each dataset, yet performance still degrades markedly under severe dataset shift. The current evaluation relies on FF++ and Celeb-DF (v2), whose acquisition and compression pipelines offer relatively controlled resolutions and illumination conditions. Native social-media or messaging content often exhibits mixed and only partially documented compression chains, more aggressive down-sampling, motion blur, and uncontrolled lighting, and emerging diffusion-based generators and alternative codecs such as HEVC or AV1 further enlarge the space of conditions. A systematic multi-codec and multi-platform evaluation of such data, together with scenario-specific assessments under adverse imaging conditions, remains an important direction for future work.

We remark that the proposed components remain compatible with a wide range of network designs and can be integrated into existing architectures with minimal structural modifications, so that the overall framework represents a practical candidate for deployment in scenarios with constrained computational resources and diverse compression settings, while providing a unified solution to the problem of detecting compressed deepfakes.

Finally, although the experimental analysis is restricted to facial video deepfakes, the proposed compression-aware formulation remains conceptually agnostic to the type of media. Similar strategies could be investigated in image forensics, where JPEG-aligned sampling and quality estimation may stabilise tampering detection under aggressive re-encoding, and in audio authenticity verification, where codec-aware augmentation and bitrate regression could improve robustness to platform-specific compression chains. A broader exploration of compression-aware learning across visual and audio modalities represents a promising direction for future research in multimedia forensics.

Data availability

Data will be made available on request.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

This work is supported by the European Union - NextGenerationEU within the PRIN 2022 PNRR - BullyBuster 2 - the ongoing fight against bullying and cyberbullying with the help of artificial intelligence for the human wellbeing (CUP: F53C2200074007, Proj. Code: P2022K39K8) and within the SERICS (PE00000014) under the Italian Ministry of University (MUR) and Research National Recovery and Resilience Plan.

Supplementary material

Supplementary material associated with this article can be found, in the online version, at 10.1016/j.eswa.2026.131761.

References

Abbas, F., & Taeihagh, A. (2024). Unmasking deepfakes: A systematic review of deepfake detection and generation techniques using artificial intelligence. *Expert Systems with Applications*, *252*, 124260.

Benjamin, D. J., Berger, J. O., Johannesson, M., Nosek, B. A., Wagenmakers, E.-J., Berk, R., Bollen, K. A., Brembs, B., Brown, L., Camerer, C., et al. (2018). Redefine statistical significance. *Nature Human Behaviour*, *2*(1), 6–10.

Blauth, T. F., Gstrein, O. J., & Zwitter, A. (2022). Artificial intelligence crime: An overview of malicious use and abuse of AI. *IEEE Access*, *10*, 77110–77122.

Bondi, L., Cannas, E. D., Bestagini, P., & Tubaro, S. (2020). Training strategies and data augmentations in CNN-based deepfake video detection. In *2020 IEEE international workshop on information forensics and security (WIFS)* (pp. 1–6). IEEE.

Breiman, L. (1996). Bagging predictors. *Machine Learning*, *24*, 123–140.

Cao, J., Ma, C., Yao, T., Chen, S., Ding, S., & Yang, X. (2022). End-to-end reconstruction-classification learning for face forgery detection. In *2022 IEEE/CVF conference on computer vision and pattern recognition (CVPR)* (pp. 4103–4112). https://doi.org/10.1109/CVPR52688.2022.00408

Chen, G.-L., & Hsu, C.-C. (2023). Jointly defending deepfake manipulation and adversarial attack using decoy mechanism. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *45*(8), 9922–9931. https://doi.org/10.1109/TPAMI.2023.3253390

Chen, S., Yao, T., Chen, Y., Ding, S., Li, J., & Ji, R. (2021). Local relation learning for face forgery detection. *Proceedings of the AAAI Conference on Artificial Intelligence*, *35*, 1081–1088.

Chen, Z., Liao, X., Wu, X., & Chen, Y. (2024). Compressed Deepfake Video Detection Based on 3D Spatiotemporal Trajectories. In *Proc. APSIPA Annual Summit and Conference (APSIPA ASC)*, 1–8

Chhabra, S., Thakral, K., Mittal, S., Vatsa, M., & Singh, R. (2023). Low quality deepfake detection via unseen artifacts. *IEEE Transactions on Artificial Intelligence* (pp. 1–13). https://doi.org/10.1109/TAI.2023.3299894

Chollet, F. (2017). Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1251–1258).

Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research : JMLR*, *7*, 1–30.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. arXiv:2010.11929.

Efron, B. (1992). Bootstrap methods: another look at the jackknife, In *Breakthroughs in statistics: Methodology and distribution*. Springer, 569–593

Gao, J., Xia, Z., Marcialis, G. L., Dang, C., Dai, J., & Feng, X. (2024). Deepfake detection based on high-frequency enhancement network for highly compressed content. *Expert Systems with Applications*, *249*, 123732.

Glorot, X., & Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics* the Thirteenth International Conference on Artificial Intelligence and Statistics (pp. 249–256).

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). Generative adversarial nets. *Advances in Neural Information Processing Systems*, *27*, 2672–2680.

Guarnera, L., Giudice, O., Guarnera, F., Ortis, A., Puglisi, G., Paratore, A., Bui, L. M. Q., Fontani, M., Coccomini, D. A., Caldelli, R., Falchi, F., Gennaro, C., Messina, N., Amato, G., Perelli, G., Concas, S., Cuccu, C., Orrù, G., Marcialis, G. L., & Battiato, S. (2022). The face deepfake detection challenge. *Journal of Imaging*, *8*(10), 263.

Heidari, A., Jafari Navimipour, N., Dag, H., & MUnal (2024). Deepfake detection using deep learning methods: A systematic and comprehensive review, *Wiley Interdisciplinary Reviews*. 14 (2), e1520.

Howard, A., Sandler, M., Chu, G., Chen, L.-C., Chen, B., Tan, M., Wang, W., Zhu, Y., Pang, R., Vasudevan, V., et al. (2019). Searching for mobilenetv3. In *Proceedings of the IEEE/CVF international conference on computer vision* the IEEE/CVF international conference on computer vision (pp. 1314–1324).

Humidan, A.-S., Abdullah, L. N., & Halin, A. A. (2022). Detection of compressed deepfake video drawbacks and technical developments. In *2022 5th international conference on signal processing and information security (ICSPIS)* (pp. 11–16). https://doi.org/10.1109/ICSPIS57063.2022.10002433

Kerby, D. S. (2014). *The simple difference formula: An approach to teaching nonparametric correlation*. Comprehensive Psychology, 3. https://doi.org/10.2466/11.IT.3.1

Khormali, A., & Yuan, J.-S. (2022). DFDT: An end-to-end deepfake detection framework using vision transformer. *Applied Sciences*, *12*(6), 2953.

Kohli, A., & Gupta, A. (2022). Light-weight 3D CNN for deepfakes, faceswap and face2face facial forgery detection. *Multimedia Tools and Applications*, *81*(22), 31391–31403.

Kornblith, S., Norouzi, M., Lee, H., & Hinton, G. (2019). Similarity of neural network representations revisited. In *International conference on machine learning*. PMLR, 3519–3529.

Li, C., Zheng, Z., Bin, Y., Wang, G., Yang, Y., Li, X., & Shen, H. T. (2023). Pixel bleach network for detecting face forgery under compression. (pp. 1–13). https://doi.org/10.1109/TMM.2023.3301242

Li, J., Xie, H., Li, J., Wang, Z., & Zhang, Y. (2021). Frequency-aware discriminative feature learning supervised by single-center loss for face forgery detection. In *2021 IEEE/CVF conference on computer vision and pattern recognition (CVPR)* (pp. 6454–6463). https://doi.org/10.1109/CVPR46437.2021.00639

Li, L., Bao, J., Yang, H., Chen, D., & Wen, F., (2019). FaceShifter: Towards high fidelity and occlusion aware face swapping. Technical Report arXiv:1912.13457.

Li, Y., Bian, S., Wang, C., Polat, K., Alhudhaif, A., & Alenezi, F. (2023). Exposing low-quality deepfake videos of social network service using spatial restored detection framework. *Expert Systems with Applications*, *231*, 120646.

Li, Y., Yang, X., Sun, P., Qi, H., & Lyu, S. (2020). Celeb-DF: A large-scale challenging dataset for deepfake forensics. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 3207–3216).

Liao, X., Wang, Y., Wang, T., Hu, J., & Wu, X. (2023). FAMM: Facial muscle motions for detecting compressed deepfake videos over social networks. In *IEEE transactions on circuits and systems for video technology* (pp. 1–1). https://doi.org/10.1109/TCSVT.2023.3278310

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

Ma, S., Zhang, X., Jia, C., Zhao, Z., Wang, S., & Wang, S. (2019). Image and video compression with neural networks: A review. *IEEE Transactions on Circuits and Systems for Video Technology*, *30*(6), 1683–1698.

Maiano, L., Amerini, I., Ricciardi Celsi, L., & Anagnostopoulos, A., (2021). Identification of social-media platform of videos through the use of shared features. *Journal of Imaging*, *7*(8), 140.

Miao, C., Chu, Q., Li, W., Gong, T., Zhuang, W., & Yu, N. (2021). Towards generalizable and robust face manipulation detection via bag-of-local-feature. arXiv:2103.07915.

Mitra, A., Mohanty, S. P., Corcoran, P., & Kougianos, E. (2020). A novel machine learning based method for deepfake video detection in social media. In *2020 IEEE international symposium on smart electronic systems (ISES) (Formerly INIS)* (pp. 91–96). https://doi.org/10.1109/iSES50453.2020.00031

Mubarak, R., Alsboui, T., Alshaikh, O., Inuwa-Dute, I., Khan, S., & Parkinson, S. (2023). A survey on the detection and impacts of deepfakes in visual, audio, and textual formats. IEEE Access.

Qian, Y., Yin, G., Sheng, L., Chen, Z., & Shao, J. (2020). Thinking in frequency: Face forgery detection by mining frequency-aware clues. In A. Vedaldi, H. Bischof, T. Brox, & J.-M. Frahm (Eds.), *Computer vision - eccv 2020* Cham (pp. 86–103). Springer International Publishing.

Rancourt-Raymond, A. D., & Smaili, N. (2023). The unethical use of deepfakes. *Journal of Financial Crime*, *30*(4), 1066–1077.

Rossler, A., Cozzolino, D., Verdoliva, L., Riess, C., Thies, J., Nießner, M., & (2019). FaceForensics++: Learning to detect manipulated facial images. In *Proceedings of the ieee/cvf international conference on computer vision* the IEEE/CVF international conference on computer vision (pp. 1–11).

Sikora, T. (1997). The mpeg-4 video standard verification model. *IEEE Transactions on Circuits and Systems for Video Technology*, *7*(1), 19–31.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, *15*(1), 1929–1958.

Sun, K., Liu, H., Yao, T., Sun, X., Chen, S., Ding, S., & Ji, R. (2022). An information theoretic approach for attention-driven face forgery detection. Springer.

Tan, M., & Le, Q., (2021). EfficientNetv: Smaller models and faster training In *International conference on machine learning*, PMLR, *2*, 10096–10106.

Tarekegn, A. N., Ullah, M., & Cheikh, F. A. (2024). Deep learning for multi-label learning: A comprehensive survey. arXiv:2401.16549.

Thies, J., Zollhöfer, M., & Nießner, M. (2019). Deferred neural rendering: Image synthesis using neural textures. *ACM Transactions on Graphics (TOG)*, *38*(4), 1–12.

Thies, J., Zollhofer, M., Stamminger, M., Theobalt, C., & Nießner, M. (2016). Face2face: Real-time face capture and reenactment of rgb videos, In *Proceedings of the IEEE conference on computer vision and pattern recognition* , 2387–2395

Tran, D., Wang, H., Torresani, L., Ray, J., Lecun, Y., & Paluri, M. (2018). A closer look at spatiotemporal convolutions for action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 6450–6459).

Wang, S.-Y., Wang, O., Zhang, R., Owens, A., & Efros, A. A. (2020). CNN-generated images are surprisingly easy to spot…for now. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 8695–8704).

Wiegand, T., Sullivan, G. J., Bjontegaard, G., & Luthra, A. (2003). Overview of the h.264/AVC video coding standard. *IEEE Transactions on Circuits and Systems for Video Technology*, *13*(7), 560–576.

Wilcoxon, F. (1945). Individual comparisons by ranking methods. *Biometrics Bulletin*, *1*(6), 80–83.

Yang, L. (2011). Classifiers selection for ensemble learning based on accuracy and diversity. *Procedia Engineering*, *15*, 4266–4270.

Yang, Z., Liang, J., Xu, Y., Zhang, X.-Y., & He, R. (2023). Masked relation learning for deepfake detection. *IEEE Transactions on Information Forensics and Security*, *18*, 1696–1708. https://doi.org/10.1109/TIFS.2023.3249566

Yin, X., & Liu, X. (2017). Multi-task convolutional neural network for pose-invariant face recognition. *IEEE Transactions on Image Processing*, *27*(2), 964–975.

Zhang, J., Ni, J., & Xie, H. (2021). Deepfake videos detection using self-supervised decoupling network. In *2021 IEEE international conference on multimedia and expo (ICME)* (pp. 1–6). https://doi.org/10.1109/ICME51207.2021.9428368

Zhang, Y., & Yang, Q. (2021). A survey on multi-task learning. *IEEE Transactions on Knowledge and Data Engineering*, *34*(12), 5586–5609.

Zhao, H., Zhou, W., Chen, D., Wei, T., Zhang, W., & Yu, N. (2021). Multi-attentional deepfake detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 2185–2194).

Zhao, Y., Liu, B., Ding, M., Liu, B., Zhu, T., & Yu, X. (2023). Proactive deepfake defence via identity watermarking. In *2023 IEEE/CVF winter conference on applications of computer vision (WACV)* (pp. 4591–4600). https://doi.org/10.1109/WACV56688.2023.00458

Zhou, X., Wang, Y., & Wu, P. (2020). Detecting deepfake videos via frame serialization learning. In *2020 IEEE 3rd international conference of safe production and informatization (IICSPI)* (pp. 391–395). https://doi.org/10.1109/IICSPI51290.2020.9332419