Faster Than Lies: Real-time Deepfake Detection using Binary Neural Networks

Romeo Lanzino, Federico Fontana, Anxhelo Diko, Marco Raoul Marini, Luigi Cinque
Department of Computer Science
Sapienza University of Rome
Via Salaria 113, 00198
Rome, Italy
{lanzino, fontana.f, diko, marini, cinque}@di.uniroma1.it

Abstract

Deepfake detection aims to contrast the spread of deep-generated media that undermines trust in online content. While existing methods focus on large and complex models, the need for real-time detection demands greater efficiency. With this in mind, unlike previous work, we introduce a novel deepfake detection approach on images using Binary Neural Networks (BNNs) for fast inference with minimal accuracy loss. Moreover, our method incorporates Fast Fourier Transform (FFT) and Local Binary Pattern (LBP) as additional channel features to uncover manipulation traces in frequency and texture domains. Evaluations on COCOFake, DFFD, and CIFAKE datasets demonstrate our method's state-of-the-art performance in most scenarios with a significant efficiency gain of up to a 20× reduction in FLOPs during inference. Finally, by exploring BNNs in deepfake detection to balance accuracy and efficiency, this work paves the way for future research on efficient deepfake detection.

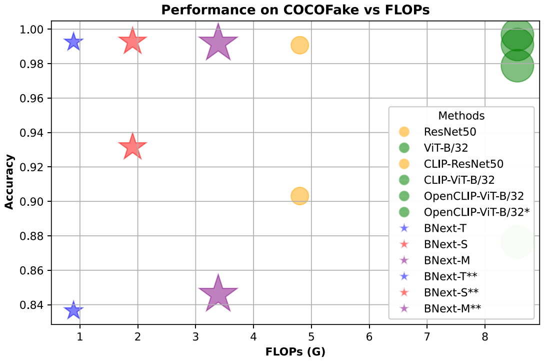

Figure 1. Depiction of the trade-off between performance (accuracy) and the computational complexity measured in FLOPs (G) on the COCOFake [2] dataset. Points with the same color indicate models that share the same architecture. The size of a point represents the number of parameters of its model. *This model shares the architecture with the one with the same name but is trained on a different dataset. **In these models, which are the ones on the bottom left part, the backbone is kept frozen.

1. Introduction

The rise of deepfakes and media manipulated with sophisticated artificial intelligence threatens to erode the foundation of trust in the digital age. From fabricated revenge pornography [5] to doctored political speeches [24], these synthetic creations have the power to deceive, defame, and destabilize. As deepfake creation tools become more accessible and the quality of these fakes rapidly improves [2, 28], the ability to distinguish authentic content from malicious manipulation has become a critical battleground for preserving truth and preventing the diffusion of fake information. This urge to identify deepfakes has brought much attention to the rapidly evolving field of deepfakes detection [1, 7, 13, 27]. As the name suggests, this field concentrates on creating intelligent algorithms that are able to depict common patterns that distinguish real from fake content [2].

Deepfake detection is an evolving task driven by the relentless advancement of generative models, especially in image generation. In such regard, recent years have seen remarkable progress in generating realistic images, thanks to Generative Adversarial Networks (GANs) [12]. This spurred the creation of deepfake detection algorithms, often based on deep neural networks [1, 7, 44], that accurately identified characteristic GAN-based artifacts. However, the emergence of diffusion models [2, 18, 43], capable of even higher-quality fakes, challenges these methods [39, 41, 45]. In tackling the challenges of identifying fake images generated by diffusion models, previous works concentrate on training large neural networks with millions of parameters to identify the inherent patterns of generated images [2]. While these detection methods show promise, their reliance on large, complex models raises concerns. Deepfakes primarily spread on social media platforms and web applications [21], where devices like mobile phones and personal computers have limited computational resources. This raises a critical question: *Can we develop efficient deep-learning-based deepfake detection methods without sacrificing accuracy?*

In the pursuit of answering our question, we shift the attention to Binary Neural Networks (BNNs). BNNs offer exceptional memory and computational savings by quantizing both weights and activations to 1-bit values [15, 42, 54, 57]. This makes them ideal for real-time deepfake detection on resource-constrained devices like phones and personal computers. Specifically, we employ the BNext [14] convolutional neural network for its proven feature extraction capabilities on RGB images. Additionally, since generative methods often leave subtle artifacts, particularly around edges and in frequency domains [13, 28, 56], we augment our RGB input with features derived from two specialized filters: the Fast Fourier Transform (FFT) magnitude, and the Local Binary Patterns (LBP). Such features further enhance the model's ability to identify generated artifacts by emphasizing micro-patterns and textures.

To assess the performance of our method's ability to identify deep-generated content and its efficiency in terms of computational resources, we conduct extensive experiments in three deepfakes detection datasets, including COCOFake [2], DFFD [10], and CIFAKE [4]. Our method competes or improves the results of existing SOTA in almost all scenarios while reducing up to 20× in computational consumption measured in FLOPs. A glimpse of the performance and computational complexity trade-off of our method compared to the current SOTA can be seen in Fig. 1.

In summary, the contributions of this work are four-fold:

1. to the best of our knowledge, we propose the first-ever implementation of a BNN for deepfake detection, which enables real-time detection on low-resource devices;
2. extensive experimentation of three benchmark datasets (COCOFake, DFFD, and CIFAKE) showcasing the exceptional ability of BNNs in identifying generated images;
3. an ablation study that highlights the impact of each design choice made in building the proposed method;
4. quantitative results that underline the quality of the proposal and promote further investigation.

The remaining parts of this work are structured as follows: Section 2 delves into the existing literature, situating our work within the broader context of deepfake detection and BNN advancements; Section 3 details the methodology employed, including our use of BNNs and the rationale behind augmenting input images with FFT magnitude and LBP channels for enhanced detection capabilities; Section 4 presents the experimental setup, the datasets utilized, the metrics for evaluation, and the results obtained, alongside an ablation study to discern the impact of each augmentation of the input images; finally, Section 5 summarizes the findings and the limitations of our work on the deepfake detection task, outlining avenues for future research.

The code is available at https://github.com/fedeloper/binary_deepfake_detection.

2. Related work

2.1. Deepfakes detection

The fight against deepfakes has spurred intense research efforts, yielding a range of detection strategies. Early methods focused on low-level artifacts stemming from generative processes, pinpointing anomalies like unnatural blinking patterns [27] or inconsistencies in physiological signals patterns [7]. However, these techniques are vulnerable to increasingly sophisticated deepfake generation methods. A more robust approach lies in analyzing mesoscopic features, such as facial warping artifacts [28], inconsistencies in head pose [56], and textural anomalies [13]. Deep neural networks, specifically CNNs, have emerged as powerful detection tools [1]. Subsequent works have explored tailored architectures like XceptionNet [44] and attention mechanisms [38] to enhance artifact detection. To address the data scarcity issue and improve adaptability to new deepfake techniques, self-supervised [17] and semi-supervised methods [9] are gaining traction. However, current techniques primarily detect GAN-generated samples, limiting their effectiveness across the full range of deepfake creation methods [2]. In recent years, diffusion models introduced new milestones in deepfake generation with high-quality images resembling natural ones [43]. The detection of such models' images presents more challenges than those created with traditional GANs, often avoiding the telltale grid-like artifacts found in GAN outputs and requiring a shift in detection strategies [39, 41, 45]. Promising research focuses on analyzing the intrinsic local dimensionality of diffusion-generated images [36], which differs from natural images. Another approach investigates how diffusion models tend to overfit training data, leaving detectable traces in the form of reconstruction errors [52]. While tailored methods are emerging, it is important to note that most works rely on the use of advanced neural networks like heavy CNNs or Transformers [2] to detect deepfakes. Despite notable progress, deepfake detection faces ongoing challenges. Models often struggle to generalize to unseen deepfake generation techniques, and their performance degrades when encountering real-world distortions (e.g., video compression [13]).

2.2. Binary Neural Networks

The BNN architecture, pioneered by [8], involves binarizing weights and activations through the *sign* function, substituting the bulk of arithmetic operations in deep neural networks with bit-wise operations. To address quantization error, the XOR-Net [42] introduced a channel-wise scaling factor for reconstructing binarized weights, a technique pivotal in subsequent BNN models. As proposed in [32], ABC-Net endeavors to approximate full-precision weights with a linear combination of binary weight bases and employs multiple binary activations to diminish information loss. Inspired by the full-precision networks ResNet [16] and DenseNet [19] architectures, Bi-Real Net [34] integrates shortcuts to minimize the performance gap between 1-bit and real-valued CNN models. Concurrently, BinaryDenseNet [3] enhances BNN accuracy by increasing the number of shortcuts. Further, IR-Net [40] proposes the Libra-PB method, aimed at reducing information loss during forward propagation through the maximization of quantized parameters' information entropy and minimizing quantization error, bounded within $\{-1, +1\}$. ReActNet [35] develops a generalized version of traditional *sign* and *PReLU* functions, named *RSign* and *RPReLU*, respectively, facilitating explicit learning of distribution reshaping and shifting with minimal computational overhead. RBNN [29] examines and mitigates angular bias's effect on quantization error. SiMaN [30] reveals that removing L_2 Regularization during training maximizes the entropy. ReCU [55] introduces a rectified clamp unit to revive the so-called "dead weights", thus reducing quantization error. AdaBin [50] integrates equalization methods for weights and introduces learnable parameters for activations, employing the *Max-Out* nonlinear activation function to add a negligible count of floating-point operations. B-Next [14] proposed a hybrid approach with a basic block with binarized convolutions and INT-4 linear layers, achieving state-of-the-art performances.

3. Proposed method

The proposed method, whose architecture is shown in Figure 2, processes an RGB image to classify it as either real or generated. Initially, the method augments the input image by adding two additional channels that correspond to the FFT magnitude and the LBP [51]. Subsequently, these augmented images undergo adaptation through an Adapter to revert them to a 3-channel format, which is then fed into the backbone for feature extraction. The extracted features are then classified as real or fake.

3.1. Augmented features

The model takes an RGB image in $\mathbb{R}^{3\times h\times w}$ as input, where h and w are its height and width, respectively. This study enriches this image with two channels representing its FFT magnitude and LBP. These augmentations are specifically selected to underscore the subtle yet significant micro-patterns that deepfakes often disrupt, leveraging the intuition that specific texture and edge information can be pivotal in distinguishing between genuine and generated imagery.

We assume that deepfakes could introduce distortions in the frequency domain which are typically not present in genuine images. Thus, we exploited the FFT magnitude channel to highlight these anomalies. This channel is obtained by applying the FFT to the image to extract its magnitude spectrum.

The LBP is a texture descriptor that encapsulates the local spatial structure of an image. It is introduced in the pipeline to capture the unique textures of facial features. Those are areas where deepfakes typically struggle to maintain accuracy. LBP enriches the model's input with robust texture pattern features by comparing each pixel with its neighbors and encoding this comparison into a new image.

3.2. Adapter

As the backbone is compatible with 3-channel images and the augmented image has more than those channels, we introduced a particular layer just before the backbone, namely the Adapter, to manage the new features. It is a convolution layer that takes as input an image with 5 channels (red, green, blue, FFT magnitude, and LBP) and squeezes them into an image of the same height and width but with just 3 channels, according to the shape of the input accepted by the backbone.

3.3. Binary backbone

The concept of a BNN was pioneered by [8]. Their innovative approach entailed using the sign function to binarize weights and activations, effectively replacing the majority of arithmetic computations in deep neural networks with bit-wise operations, obtaining a theoretical speedup of 58× in inference speed and 32× times less memory needed. Before the explanation of our proposed BNN for this work, we first describe what are the differences between a full-precision and a binary CNN.

We describe a CNN by representing its layer-specific real-valued weights as $\mathcal{W}_r$ and the inputs as $\mathcal{A}_r$. Consequently, the output $\mathcal{Y}$ of a convolution is formulated as follows:

$$\mathcal{Y} = \mathcal{A}_r \otimes \mathcal{W}_r, \tag{1}$$

with $\otimes$ symbolizing standard convolution operations. BNNs seeks to convert every weight in $\mathcal{W}_r$ and every activation $\mathcal{A}_r$ into their binary counterparts $\mathcal{W}_b$ and $\mathcal{A}_b$, whose values are quantized in $\{-1, +1\}$ using the *sign* function. For a real-valued input x_r, this function is defined as fol-

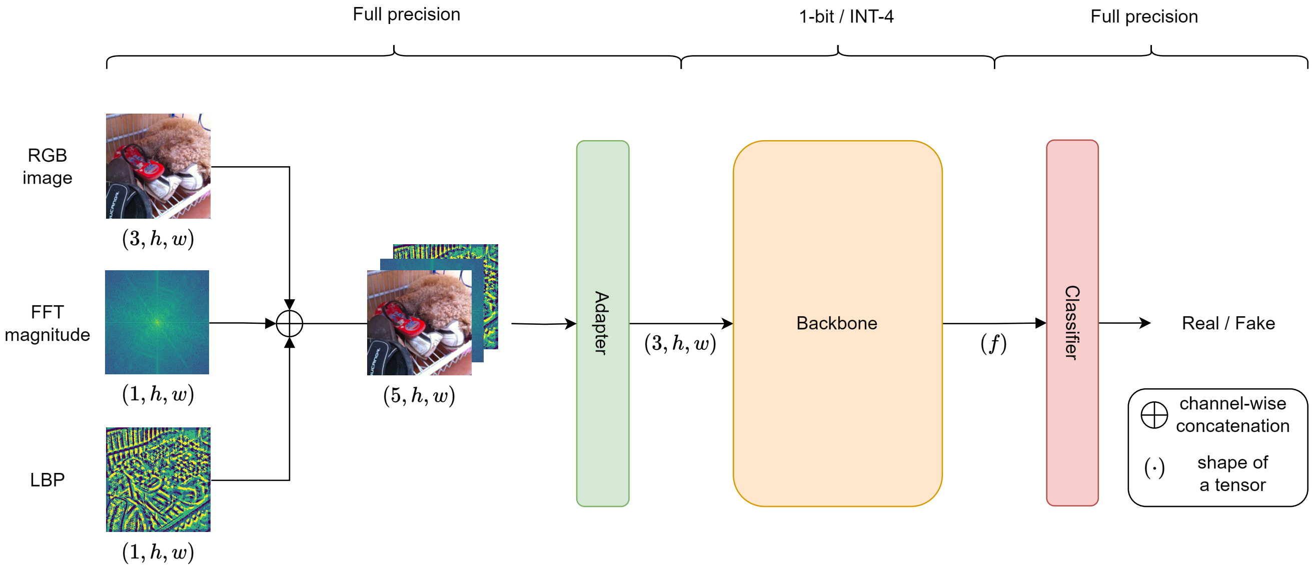

Figure 2. The architecture of the proposed model.

lows:

$$sign(x_r) = \begin{cases} +1, & \text{if } x_r \geq 0, \\ -1, & \text{otherwise.} \end{cases} \tag{2}$$

To address the significant quantization error inherent in deep neural networks binarization, XNOR-Net [8] introduces dual scaling factors for both weights $\mathcal{W}_b$ and activations $\mathcal{A}_b$. Following the methodology outlined by [8], this document simplifies the representation of these scaling factors to a single parameter α. Hence, the output of a binary convolution is expressed as follows:

$$\mathcal{Y} = \mathcal{A}_r \otimes \mathcal{W}_r \approx (\mathcal{A}_b \circledast \mathcal{W}_b) \odot \alpha, \tag{3}$$

where $\circledast$ denotes bit-wise operations including XNOR and POPCOUNT, and $\odot$ stands for element-wise multiplication.

The proposed method uses BNext [14] as a backbone, which is pre-trained on the ImageNet [26] dataset. In BNext, each convolution operation is binary, and other operations are quantized to INT-4, resulting in much more efficient networks in terms of operations. The backbone uses a binary convolution module with full precision skip-connection and a branch with precision INT-4 to facilitate information propagation and alleviate possible bottlenecks.

The f features extracted by the backbone, represented as a tensor in $\{-1, +1\}^f$, are thus given as input to a full-precision linear layer that outputs the logits for real/fake classification.

4. Experiments

4.1. Datasets

We leverage three distinct datasets, each offering unique challenges and characteristics, to effectively evaluate our deepfake detection method. The datasets are the COCOFake [2], the Diverse Fake Face Dataset (DFFD) [10], and the CIFAKE [4].

The COCOFake dataset builds upon the COCO dataset [31] and augments it with images using the Stable Diffusion [43] text-to-image model. In particular, for each real image in COCO, which is accompanied by 5 captions, 5 corresponding synthetic images are created. This approach maintains the integrity of the original COCO dataset's division into training, validation, and test sets. The dataset comprises more than 650K training images, 30K validation images, and 30K test images.

The DFFD is an extensive collection of images aimed at enhancing the detection and localization of facial manipulations. It covers four primary types of facial manipulations: identity swaps, expression swaps, attribute manipulations, and entirely synthesized faces. Real face samples are sourced from the FFHQ [23], CelebA [33] datasets (which offer a wide range of demographic and quality diversity), and additional real images from the FaceForensics++ [44] dataset. For fake samples, PGGAN [22] and StyleGAN [23] are employed to create manipulated images in line with the 4 manipulation categories, summing up to 58K real and 240K fake images. Half of the samples are used in the training set, while 5% for validation and 45% for test sets, respectively, ensuring that manipulations derived from the same source image remain within the same set.

The CIFAKE dataset is structured to parallel the CIFAR-10 [25] dataset, featuring a balanced composition of real and synthetic 32×32 pixels images across ten classes. Specifically, it includes 60K real images directly taken from CIFAR-10 and an equal number of synthetic images generated using the Stable Diffusion [43] model, summing up to 120K images. The dataset is split into training and test sets, with 50K images designated for the first and 10K reserved for the latter.

4.2. Metrics

Several metrics have been employed when comparing our method with the state of the art. These include metrics related to classification performance (accuracy, Area Under the Curve (AUC)), and computational requirements (floating point operations per second (FLOPs). These metrics represent the standard in the deepfake detection [2, 4, 10] and Binary Neural Networks [14, 29, 50] fields.

Accuracy is defined as the ratio of correctly predicted observations, namely True Positives (TP) and True Negatives (TN), to the total observations, which includes False Positives (FP) and False Negatives (FN). It is a measure of the model's overall correctness across all classes and is particularly useful for balanced datasets.

The AUC measures the ability of a model to discriminate between positive and negative classes. The AUC is the area under the Receiver Operating Characteristic (ROC) curve, which plots the TP rate against the FP rate at various thresholds.

The count of FLOPs is a measure of the computational complexity of a model, indicating the total number of floating point precision required for a single forward pass. This metric is crucial for understanding neural networks' computational demand and efficiency, especially when deploying models in resource-constrained environments.

4.3. Experimental setup and training details

Following the standard methodology used in related literature regarding the preprocessing of images, an initial resizing step was undertaken to standardize the longest dimension of each image to 252 pixels. Subsequently, a central crop measuring 224×224 pixels was extracted from images within the validation and test sets. In contrast, crops from the training set were randomly obtained to introduce some variability. Data augmentation techniques were employed to augment the training dataset's diversity further. These techniques included random flips (both horizontal and vertical), rotations (either $90°$ or $270°$), and color jittering within a range of $[0\%, 20\%]$. All images are then normalized to have mean $[0.485, 0.456, 0.406]$ and standard deviation $[0.229, 0.224, 0.225]$. This particular choice of image size and color normalization has been done to maintain continuity with that of the data used during the pre-training of the backbone.

The model optimization was achieved using the AdamW [37] optimizer in combination with the binary cross-entropy loss. It was configured with an initial learning rate of 10^{-4}, with the first and second-moment estimates (β_1 and β_2) set to 0.9 and 0.999, respectively, accompanied by a weight decay parameter of 10^{-2}. Additionally, a learning rate scheduler was implemented to methodically reduce the learning rate to 10^{-5} by the conclusion of the fifth epoch, optimizing the training process over time. The batch size was set to 128. We experimented with two configurations of the models to evaluate their performance under different computational constraints: a version where the backbone is kept frozen to assess the model's behavior in conserving computational resources during training a trainable backbone to maximize the accuracy. Based on the involved dataset, a maximum epoch limit was established throughout the experimental phase. For COCOFake and DFFD datasets (which details are provided in Section 4.1) the epoch limit was set at 5, a decision driven by the observation that the model convergence was typically achieved well before this threshold. Conversely, for the CIFAKE dataset, which is notably smaller in size, the epoch limit was extended to 20 to accommodate the dataset's unique characteristics and ensure adequate model training.

The code was implemented in Python, leveraging the PyTorch framework for Deep Learning. Computational tasks were performed on an NVIDIA RTX 2080Ti GPU with 12GB of VRAM.

4.4. Results

We report the results of our benchmark on the COCOFake dataset in Table 1. Our results were compared with the method proposed in [2]. To maintain a fair comparison, we set our model with ResNet-50 [16] and ViT-B/32 [11], both pre-trained on ImageNet as the models we used. In the case of our models with a frozen backbone, we surpass the result of ResNet-50 by 2.84 accuracy points with the table's second-best model, BNext-S. When we also train the backbone, the margin of outperformance over ResNet-50 expanded to 8.97 accuracy points. This proves that our model can perform better than full-precision models initialized on the same dataset while having substantially lower FLOPs. Furthermore, our approach remains competitive when comparing our method with models pre-trained on substantially larger datasets. With its 99.28% accuracy, our best model trails the best-performing model, OpenCLIP-ViT-B/32 trained on LAION-2B, by just 0.4 points. Notably, the latter model was trained on a dataset comprising 2 billion images, in contrast to the 1.2 million images of the ImageNet dataset in which our models were pre-trained; this fact highlights the efficacy of our method despite its re-

Method	Model	Pre-training dataset	Accuracy	AUC	Parameters (M)	FLOPs (G)
[2]	ResNet50	ImageNet	90.31	-	25.6	4.8
	ViT-B/32	ImageNet	87.64	-	88.3	8.56
	CLIP-ResNet50	OpenAI WIT	99.07	-	25.6	4.8
	CLIP-ViT-B/32	OpenAI WIT	99.11	-	88.3	8.56
	OpenCLIP-ViT-B/32	LAION-400M	97.88	-	88.3	8.56
	OpenCLIP-ViT-B/32	LAION-2B	**99.68**	-	88.3	8.56
Ours	BNext-T with frozen backbone	ImageNet	83.65	81.98	29.8	**0.89**
	BNext-S with frozen backbone	ImageNet	93.15	95.19	67.1	1.91
	BNext-M with frozen backbone	ImageNet	84.59	82.11	133	3.39
	BNext-T	ImageNet	99.25	99.86	29.8	**0.89**
	BNext-S	ImageNet	99.28	99.89	67.1	1.91
	BNext-M	ImageNet	99.18	**99.91**	133	3.39

Table 1. Results on the COCOFake validation set. The models from [2] were trained on different datasets, including ImageNet [26], OpenAI WIT [49], LAION-400M [46] and LAION-2B [47]. **Bold** and underlined values respectively indicate the first and second-best results within their column.

Method	Model	Accuracy	AUC	Parameters (M)	FLOPs (G)
[10]	Xception	-	99.64	40.0	18.0
	VGG16	-	99.67	138.4	15.5
Ours	BNext-T with frozen backbone	89.56	87.65	29.8	**0.89**
	BNext-S with frozen backbone	89.69	88.58	67.1	1.91
	BNext-M with frozen backbone	89.61	86.64	133	3.39
	BNext-T	98.95	**99.94**	29.8	**0.89**
	BNext-S	**99.01**	**99.94**	67.1	1.91
	BNext-M	98.75	99.92	133	3.39

Table 2. Results on the DFFD test set. **Bold** and underlined values respectively indicate the first and second-best results within their column.

duced precision.

The outcomes on the DFFD dataset, detailed in Table 2, are compared against the performances of Xception [6] and VGG16 [48] as outlined in [10]. As noticed, our models with a trainable backbone consistently perform better than the two compared models, achieving an improvement of 0.27 AUC points while having up to 17.1 fewer GFLOPs.

Regarding the results on the CIFAKE dataset, shown in Table 3, we compare our method with the models proposed in [4]. To keep a fair comparison, we consider the results of BNext with a trainable backbone given the same training conditions. However, [4] does not specify the exact version of VGG and DenseNet [20], making it impossible to directly compare the number of parameters and FLOPs. To aid the reader, we have included metrics for the smallest versions of these models. Despite this, BNext demonstrates competitive performance while requiring fewer GFLOPs.

4.5. Ablation study

We conducted an ablation study to ascertain the optimal amalgamation of features incorporated into the input. The results are delineated in Table 4. Specifically, this study juxtaposed a baseline model against various configurations incorporating supplementary channels, using accuracy as a metric since the differences in the number of FLOPs are negligible. The baseline model processes an RGB image as input, directly channeling it into a pre-trained BNext-T backbone without integrating an Adapter. Conversely, the alternative models evaluated entail baseline variations, each retrained with the inclusion of one or more of the additional channels, each paired with a congruent Adapter. These new channels are the FFT magnitude and LBP described in Section 3, plus the one obtained by applying a Sobel filter to the image. The latter is employed to accentuate the edge information of the image and is obtained by applying a pair of 3×3 convolution kernels, one estimating the gradient horizontally and the other vertically, to approximate the gradient magnitude of the image at each point. By emphasizing

Method	Model	Accuracy	AUC	Parameters (M)	FLOPs (G)
[53]	ResNet-50	95.00	99.00	25.6	4.8
	VGG	96.00	99.00	133	7.63
	DenseNet	**98.00**	99.00	7.9	5.6
Ours	BNext-T with frozen backbone	83.89	91.70	29.8	**0.89**
	BNext-S with frozen backbone	80.71	89.25	67.1	1.91
	BNext-M with frozen backbone	82.77	90.73	133	3.39
	BNext-T	97.29	**99.65**	29.8	**0.89**
	BNext-S	96.96	99.55	67.1	1.91
	BNext-M	97.35	99.62	133	3.39

Table 3. Results on the CIFAKE test set. All the models were pre-trained on the ImageNet [26] dataset. **Bold** and underlined values respectively indicate the first and second-best results within their column.

Ablation	Variations	Accuracy (%)
Baseline	-	90.35
Features added to the learned ones	Magnitude	82.36
	FFT	88.18
	LBP	88.42
	Magnitude and FFT	81.20
	Magnitude and LBP	81.67
	FFT and LBP	**91.60**
	Magnitude, FFT and LBP	81.56

Table 4. Ablation study on the COCOFake Dataset.

edges and contours, we thought the Sobel filter would aid in highlighting discrepancies in the boundary regions often overlooked by deepfakes, focusing on the premise that genuine images possess naturally smooth transitions, which manipulated images struggle to replicate accurately.

The ablation study highlights that FFT magnitude and LBP channels, when combined, markedly improve model performance in detecting manipulated images, surpassing the baseline and other variations. The increase amounts to 1.25% over the second best-performing configuration, which is the baseline. This synergy stems from their complementary analytical approaches: FFT magnitude exposes anomalies in the frequency domain indicative of digital manipulation. At the same time, LBP captures nuanced local texture patterns disrupted by such manipulations. Alone, each channel is seen to have a partial view: FFT magnitude might miss subtle textural alterations, and LBP could overlook frequency-based distortions. Together, they cover spectral and spatial discrepancies, enhancing detection capabilities. The Sobel filter's marginal impact suggests that edge information alone is insufficient for deepfake detection, underscoring the importance of integrating features that address global and local image characteristics for optimal performance.

5. Conclusion

In this study, we investigate the performance of more computationally efficient neural networks, particularly BNNs, in the context of deepfake detection tasks. Our findings reveal that the proposed BNN-based method, which requires up to 5 times fewer FLOPs compared to a ResNet-50 model and nearly 10 times fewer FLOPs than a ViT-B/32 model, is capable of matching the performance of their full-precision counterparts with minimal loss in classification accuracy. These results suggest a promising direction for enhancing the efficiency of deepfake detection methodologies.

One notable limitation of our study is the emphasis on theoretical FLOP reductions, as the real-world application of BNNs necessitates a specialized framework or accelerator to fully realize the benefits of reduced precision. Furthermore, our evaluation was confined to a network pre-trained on the ImageNet dataset, whereas other investigations have leveraged larger datasets for pre-training, thereby achieving enhanced transfer-learning capabilities.

For future research, there is potential for practical implementation of our proposed method on specialized hardware or within specific computational frameworks to actualize the theoretical efficiency gains. Additionally, exploring alternative pre-training datasets could further augment the transfer-learning efficacy of the network, potentially leading to more robust and efficient deepfake detection systems.

6. Acknoledgements

The research leading to these results has received funding from Project “Ecosistema dell’innovazione - Rome Technopole” financed by EU in NextGenerationEU plan through MUR Decree n. 1051 23.06.2022 - CUP H33C22000420001.

References

[1] Darius Afchar, Vincent Nozick, Junichi Yamagishi, and Isao Echizen. Mesonet: a compact facial video forgery detection network. In *International Workshop on Information Forensics and Security*, pages 1–7. IEEE, 2018. 1, 2

[2] Roberto Amoroso, Davide Morelli, Marcella Cornia, Lorenzo Baraldi, Alberto Del Bimbo, and Rita Cucchiara. Parents and children: Distinguishing multimodal deepfakes from natural images. *arXiv preprint arXiv:2304.00500*, 2023. 1, 2, 4, 5, 6

[3] Joseph Bethge, Haojin Yang, Marvin Bornstein, and Christoph Meinel. Back to simplicity: How to train accurate bnns from scratch? *arXiv preprint arXiv:1906.08637*, 2019. 3

[4] Jordan J. Bird and Ahmad Lotfi. Cifake: Image classification and explainable identification of ai-generated synthetic images. *IEEE Access*, 12:15642–15650, 2024. 2, 4, 5, 6

[5] Bobby Chesney and Danielle Citron. Deep fakes: A looming challenge for privacy, democracy, and national security. *Calif. L. Rev.*, 107:1753, 2019. 1

[6] Francois Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 6

[7] Umur Aybars Ciftci, Ilke Demir, and Lijun Yin. Fakecatcher: Detection of synthetic portrait videos using biological signals. *IEEE transactions on Pattern Analysis and Machine Intelligence*, 2020. 1, 2

[8] Matthieu Courbariaux, Itay Hubara, Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio. Binarized neural networks: Training deep neural networks with weights and activations constrained to+ 1 or-1. *arXiv preprint arXiv:1602.02830*, 2016. 3, 4

[9] Davide Cozzolino, Justus Thies, Andreas Rössler, Christian Riess, Matthias Nießner, and Luisa Verdoliva. Forensictransfer: Weakly-supervised domain adaptation for forgery detection. *Conference on Computer Vision and Pattern Recognition*, 2018. 2

[10] Hao Dang, Feng Liu, Joel Stehouwer, Xiaoming Liu, and Anil K. Jain. On the detection of digital face manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2, 4, 5, 6

[11] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2021. 5

[12] Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. *Deep learning*. MIT Press, 2016. 1

[13] Luca Guarnera, Oliver Giudice, and Sebastiano Battiato. Deepfake detection by analyzing convolutional traces. In *Conference on Computer Vision and Pattern Recognition Workshops*, pages 666–667, 2020. 1, 2

[14] Nianhui Guo, Joseph Bethge, Christoph Meinel, and Haojin Yang. Join the high accuracy club on imagenet with a binary neural network ticket. *arXiv preprint arXiv:2211.12933*, 2022. 2, 3, 4, 5

[15] Kai Han, Yunhe Wang, Yixing Xu, Chunjing Xu, Enhua Wu, and Chang Xu. Training binary neural networks through learning with noisy supervision. In *International Conference on Machine Learning*, pages 4017–4026. PMLR, 2020. 2

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 3, 5

[17] Chih-Chung Hsu, Yi-Xiu Zhuang, and Chia-Yen Lee. Deep fake image detection based on pairwise learning. *Applied Science.*, 10(1):370, 2020. 2

[18] Wenqing Hu, Chris Junchi Li, Lei Li, and Jian-Guo Liu. On the diffusion approximation of nonconvex stochastic gradient descent. *arXiv preprint arXiv:1705.07562*, 2017. 1

[19] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017. 3

[20] Forrest Iandola, Matt Moskewicz, Sergey Karayev, Ross Girshick, Trevor Darrell, and Kurt Keutzer. Densenet: Implementing efficient convnet descriptor pyramids, 2014. 6

[21] Ignas Kalpokas and Julija Kalpokiene. *Deepfakes: A Realistic Assessment of Potentials, Risks, and Policy Regulation*. Springer, 2022. 2

[22] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation, 2018. 4

[23] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 4

[24] Jan Kietzmann, Linda W Lee, Ian P McCarthy, and Tim C Kietzmann. Deepfakes: Trick or treat? *Business Horizons*, 63(2):135–146, 2020. 1

[25] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 5

[26] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012. 4, 6, 7

[27] Yuezun Li, Ming-Ching Chang, and Siwei Lyu. In ictu oculi: Exposing ai created fake videos by detecting eye blinking. In *International Workshop on Information Forensics and Security (WIFS)*, pages 1–7. IEEE, 2018. 1, 2

[28] Yuezun Li, Xin Yang, Pu Sun, Honggang Qi, and Siwei Lyu. Celeb-df: A large-scale challenging dataset for deepfake forensics. In *Conference on Computer Vision and Pattern Recognition*, pages 3207–3216, 2020. 1, 2

[29] Mingbao Lin, Rongrong Ji, Zihan Xu, Baochang Zhang, Yan Wang, Yongjian Wu, Feiyue Huang, and Chia-Wen Lin. Rotated binary neural network. *Advances in Neural Information Processing Systems*, 33, 2020. 3, 5

[30] Mingbao Lin, Rongrong Ji, Zihan Xu, Baochang Zhang, Fei Chao, Mingliang Xu, Chia-Wen Lin, and Ling Shao. Siman: Sign-to-magnitude network binarization. *arXiv preprint arXiv:2102.07981*, 2021. 3

[31] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision – ECCV 2014*, pages 740–755, Cham, 2014. Springer International Publishing. 4

[32] Xiaofan Lin, Cong Zhao, and Wei Pan. Towards accurate binary convolutional neural network. *arXiv preprint arXiv:1711.11294*, 2017. 3

[33] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2015. 4

[34] Zechun Liu, Baoyuan Wu, Wenhan Luo, Xin Yang, Wei Liu, and Kwang-Ting Cheng. Bi-real net: Enhancing the performance of 1-bit cnns with improved representational capability and advanced training algorithm. In *Proceedings of the European conference on computer vision (ECCV)*, pages 722–737, 2018. 3

[35] Zechun Liu, Zhiqiang Shen, Marios Savvides, and Kwang-Ting Cheng. Reactnet: Towards precise binary neural network with generalized activation functions. In *European Conference on Computer Vision*, pages 143–159. Springer, 2020. 3

[36] Peter Lorenz, Ricard L Durall, and Janis Keuper. Detecting images generated by deep diffusion models using their local intrinsic dimensionality. In *International Conference on Computer Vision*, pages 448–459, 2023. 2

[37] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2019. 5

[38] Huy H Nguyen, Fuming Fang, Junichi Yamagishi, and Isao Echizen. Multi-task learning for detecting and segmenting manipulated facial images and videos. In *International Conference on Biometrics Theory, Applications and Systems*, pages 1–8. IEEE, 2019. 2

[39] Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741*, 2021. 1, 2

[40] Haotong Qin, Ruihao Gong, Xianglong Liu, Mingzhu Shen, Ziran Wei, Fengwei Yu, and Jingkuan Song. Forward and backward information retention for accurate binary neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2250–2259, 2020. 3

[41] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 1 (2):3, 2022. 1, 2

[42] Mohammad Rastegari, Vicente Ordonez, Joseph Redmon, and Ali Farhadi. Xnor-net: Imagenet classification using binary convolutional neural networks. In *European conference on computer vision*, pages 525–542. Springer, 2016. 2, 3

[43] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10684–10695, 2022. 1, 2, 4, 5

[44] Andreas Rossler, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies, and Matthias Niessner. Faceforensics++: Learning to detect manipulated facial images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019. 1, 2, 4

[45] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in Neural Information Processing Systems*, 35:36479–36494, 2022. 1, 2

[46] Christoph Schuhmann, Richard Vencu, Romain Beaumont, Robert Kaczmarczyk, Clayton Mullis, Aarush Katta, Theo Coombes, Jenia Jitsev, and Aran Komatsuzaki. Laion-400m: Open dataset of clip-filtered 400 million image-text pairs, 2021. 6

[47] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, Patrick Schramowski, Srivatsa Kundurthy, Katherine Crowson, Ludwig Schmidt, Robert Kaczmarczyk, and Jenia Jitsev. Laion-5b: An open large-scale dataset for training next generation image-text models, 2022. 6

[48] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014. 6

[49] Krishna Srinivasan, Karthik Raman, Jiecao Chen, Michael Bendersky, and Marc Najork. Wit: Wikipedia-based image text dataset for multimodal multilingual machine learning. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 2021. 6

[50] Zhijun Tu, Xinghao Chen, Pengju Ren, and Yunhe Wang. Adabin: Improving binary neural networks with adaptive binary sets, 2022. 3, 5

[51] Li Wang and Dong-Chen He. Texture classification using texture spectrum. *Pattern Recognition*, 23(8):905–910, 1990. 3

[52] Sheng-Yu Wang, Oliver Wang, Richard Zhang, Andrew Owens, and Alexei A Efros. Cnn-generated images are surprisingly easy to spot... for now. In *Conference on Computer Vision and Pattern Recognition*, pages 8695–8704, 2020. 2

[53] Yuyang Wang, Yizhi Hao, and Amando Xu Cong. Harnessing machine learning for discerning ai-generated synthetic images. *arXiv preprint arXiv:2401.07358*, 2024. 7

[54] Yixing Xu, Kai Han, Chang Xu, Yehui Tang, Chunjing Xu, and Yunhe Wang. Learning frequency domain approximation for binary neural networks. In *NeurIPS*, 2021. 2

[55] Zihan Xu, Mingbao Lin, Jianzhuang Liu, Jie Chen, Ling Shao, Yue Gao, Yonghong Tian, and Rongrong Ji. Recu: Reviving the dead weights in binary neural networks. *arXiv preprint arXiv:2103.12369*, 2021. 3

[56] Xin Yang, Yuezun Li, and Siwei Lyu. Exposing deep fakes using inconsistent head poses. In *International Conference on Acoustics, Speech and Signal Processing*, pages 8261–8265. IEEE, 2019. 2

[57] Zhaohui Yang, Yunhe Wang, Kai Han, Chunjing Xu, Chao Xu, Dacheng Tao, and Chang Xu. Searching for low-bit weights in quantized neural networks. *arXiv preprint arXiv:2009.08695*, 2020. 2