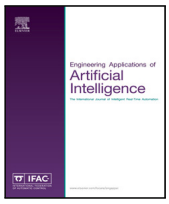
Engineering Applications of Artificial Intelligence 162 (2025) 112623

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

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



Research paper

Multi-To-Binary: A generalizable deepfake detection approach with multi-classification guidance

Fei Wang [a], Bo Wang [a], Botao Jing [a], Wei Wang [b,*], Fei Wei [c], Junxin Chen [d]

[a] School of Information and Communication Engineering, Dalian University of Technology, Dalian, 116024, China
[b] Institute of Automation, Chinese Academy of Sciences, Beijing, 100190, China
[c] Alibaba Group, Hangzhou, 310023, China
[d] School of Software, Dalian University of Technology, Dalian, 116620, China

ARTICLE INFO

Keywords:
Deepfake detection
Multi-classification guidance
Freezing mechanism
Label loss

ABSTRACT

Visual content forgery techniques, such as Deepfake, have rapidly advanced in recent years. Due to the potential misuse of these techniques for malicious purposes, there is increasing attention to the corresponding detection methods. Most existing methods focus on specific forgery patterns, making it difficult to detect forgeries with unknown or evolving patterns. In this work, we propose a novel forgery detection framework designed to extract comprehensive features utilizing multiple classification models. More specifically, our proposed framework consists of both binary-classification and multi-classification models working collaboratively, enhanced by innovative fusion and freezing mechanisms to improve accuracy and efficiency. We conducted extensive experiments to evaluate the performance of our approach. The results demonstrate that our approach outperforms state-of-the-art techniques in terms of generalization to new forgery patterns and robustness against various types of forgeries. This demonstrates promising effectiveness for real-world applications where forgeries can be diverse and sophisticated. Our code is available at https://github.com/Phoebe-cap/M2B-main.

1. Introduction

The quality of image content generation has been rapidly improving, as evidenced by recent advancements in the field (Jiang et al., 2020; Li et al., 2020a; Thies et al., 2020; Zhou et al., 2019; Vougioukas et al., 2020). Alongside this trend, numerous entertainment applications featuring image generation capabilities are entering the market, complementing conventional editing functionalities and attracting large user bases. However, there are growing concerns about the potential misuse of these applications for malicious purposes, such as image forgery. Consequently, there is an increasing demand for effective forged image detection methods, particularly for detecting forgeries in facial images.

Most of the existing approaches for facial image forgery detection utilize advanced binary classification models and are primarily based on specific forgery patterns. These methods focus on identifying and classifying particular forgery patterns, such as noise patterns (Gu et al., 2022a; Zhou et al., 2017), local textures (Li et al., 2018; Gu et al., 2022b; Chen et al., 2021; Haliassos et al., 2021; Zhao et al., 2021b; Ba et al., 2024), frequency information (Qian et al., 2020; Li et al., 2020b), and implicit identity (Dong et al., 2023; Huang et al., 2023). While these approaches have shown promising results, their performance relies heavily on prior knowledge of the specific forgery patterns. This reliance poses significant challenges when dealing with unknown or novel forgery patterns, limiting their effectiveness in real-world scenarios where forgeries continuously evolve. To overcome these challenges, it is essential to develop more generalized forgery detection methods that do not depend on specific forgery patterns. Such methods would be better equipped to handle a wider variety of forgeries, enhancing their robustness and applicability in diverse and dynamic environments where forgery techniques continually evolve.

Most of the existing methods utilize binary classifiers, while one major weakness of binary classification models is their inability to capture the differences among multiple classes, in our case, the various forgery patterns. To address this limitation and expand the classification boundaries among different forgery patterns, we propose a novel forgery detection method called *Multi-To-Binary (M2B)*. This method collaborates binary and multi-classification models through a fusion mechanism. Specifically, we adopt a two-step approach. First, we use multi-classifiers to distinguish different forgery types. Then, we use the classification results from these multi-classifiers as guidance to train the binary classifiers. This guidance is achieved through a sophisticated fusion mechanism consisting of multiple convolutional layers, which

* Corresponding author.
E-mail address: wei.wong@ia.ac.cn (W. Wang).

https://doi.org/10.1016/j.engappai.2025.112623
Received 9 August 2024; Received in revised form 31 July 2025; Accepted 2 October 2025
Available online 16 October 2025

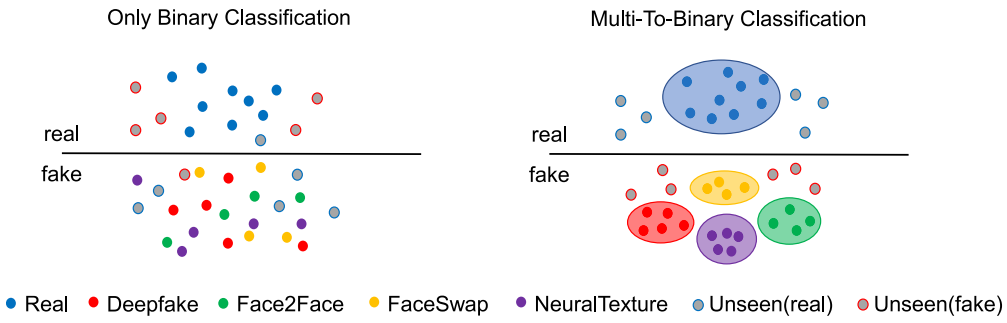


Fig. 1. Illustration of Multi-To-Binary classification strategy. Simple binary classification tends to have poor generalization performance on unseen data. Multi-To-Binary classification, on the other hand, makes the feature expression more accurate and the distinction between real and fake features more obvious, which is conducive to binary classification.

effectively capture the characteristics of multi-classification. In addition, we propose a freezing mechanism, which allows us to freeze the multi-classification model (i.e., network) in a tunable manner. This improves the performance of the binary classification model by leveraging the learned features from the multi-classification stage. Furthermore, to better synchronize the multi-classification and binary models, we introduce a label loss specifically tailored to our network structure. This loss function enables the multi-classification model to better guide the binary model.

Our approach effectively builds clear classification boundaries among the different forgery types, resulting in improved generalization ability and distinguishability between authentic and forged images, as shown in Fig. 1. More details about our proposed method will be discussed in Section 3. Comprehensive experiments successfully demonstrate the superiority of our method against the state-of-the-art solutions when the target samples are forged with unknown methods (unseen patterns).

To summarize, the main contributions of this work can be summarized as follows:

- We propose an innovative deepfake detection method that uses multi classification to guide binary classification, which addresses the concerns of detecting images forged by unknown methods, i.e. with unknown patterns.
- We propose a novel freezing mechanism that selectively freezes the multi-class network during training to emphasize binary classification as the primary task, and a fusion module that enhances the discriminability and generality of the features used for classification.

In this paper, subsequent sections are as follows: Section 2 reviews related works. Section 3 illustrates our proposed method. Section 4 details the experimental settings and explains the evaluation results and analysis. Section 5 provides our conclusion.

2. Related works

In this section, we present a concise survey of the deepfake-based visual content manipulation (forgery) and corresponding detection approaches. More specifically, we survey forgery detection in the following categories: conventional detection approaches, specific artifacts or architectures, and auxiliary tasks or synthetic data.

2.1. Deepfake manipulation

Facial image manipulation technologies can generally be categorized into four types (Mirsky and Lee, 2021): Synthesis, Editing, Replacement, and Reenactment.

Face Synthesis refers to the generation of hyper-realistic synthetic characters by learning the latent representations of facial datasets, typically using GAN networks (Goodfellow et al., 2014). It is created without a target identity as a basis, resulting in characters that do not exist in the real world.

Face Editing refers to the modification of certain attributes of a feature within a face without changing the identity information, such as eye color, hairstyle, wrinkles, skin tone, gender, and age. Applications such as FaceApp (Clark et al., 2019) allow users to alter their appearance for entertainment purposes.

Face Replacement involves substituting the content of a target identity with that of a source identity while preserving the information of the source identity. Face replacement includes two types: Transfer and Swap. Users can utilize this technology to exchange their identities with others. Common face replacement techniques include DeepFake (DF) and FaceSwap (FS).

Face Reenactment refers to the reproduction of facial expressions and actions from a source image onto a target image without altering the identity of the face in the target image. The reenacted elements include expressions, mouth movements, eye gaze, posture, and even body movements. Common face reenactment techniques include Face2Face (F2F) and NeuralTexture (NT).

Although face editing and synthesis are active research topics, face reenactment and replacement deepfakes are the most concerning, as they allow hackers to easily control a person's identity. Therefore, this paper focuses on reenactment and replacement deepfakes, specifically targeting the four most common types of forgery: DF, FS, F2F, and NT.

2.2. Conventional image forgery detection

Conventional image forgery detection methods mainly relied on signal processing techniques, which involved analyzing images through local noise analysis and quality assessment. These methods were primarily designed to detect common image tampering issues such as copying (Amerini et al., 2011), removal and splicing (De Carvalho et al., 2013). While the detection has been extensively studied, these conventional detectors (Farid, 2009; Rocha et al., 2011) are weak in effectively handling face forgery and there are several reasons. These methods were primarily designed to detect general image editing operations, which produce distinctive artifacts that are different from those generated by GAN-based facial forgery techniques. Also, face forgeries present unique challenges due to their relatively smaller size and inferior quality compared to natural images. With the continuous advancement of artificial intelligence technology, the generated content has become increasingly genuine, such that conventional image forgery detection methods are no longer sufficient to meet the requirements.

2.3. Specific artifacts or novel architectures

There are a number of works that focus on specific artifacts in deepfake images and use them for detection. For example, Li et al.

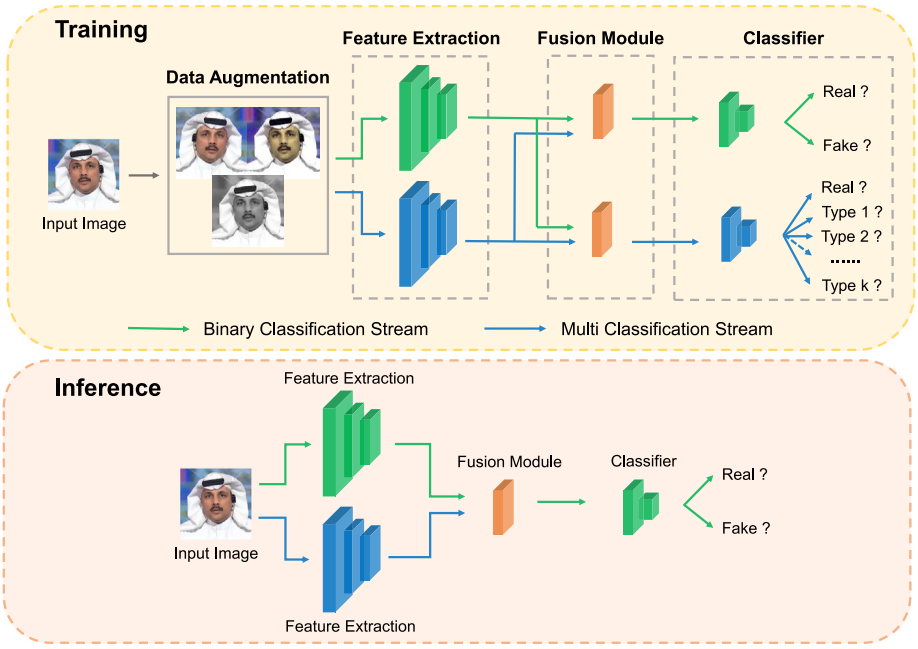


Fig. 2. The overview framework of our proposed method. During the training phase, the input images are first enhanced by data augmentation and then pass through two streams: binary classification stream and multi-classification stream. The features extracted from the both streams are fused through two same fusion modules, where the only difference is the sequence of feature inputs. Finally, the fusion characteristics pass through a binary classifier and a multi classifier, respectively. During the inference phase, we discard the multi-class classification head. Only true and false binary detection is performed.

(2018) identified the absence of blinking as a clear indication of deepfake videos. Fernandes et al. (2019) proposed a method to differentiate between genuine and fake videos based on heart rate. Similarly, Ciftci et al. (2020) found that facial signals hidden in human videos can serve as implicit cues for authenticity. Haliassos et al. (2021) focused on the irregularities in high-level semantic mouth movements for detecting fake videos. Matern et al. (2019) used inconsistencies in eye pupil color, missing details in the teeth region, and coloring errors caused by inaccurate estimation of incident lighting as artifacts to detect various forgeries. Qian et al. (2020)performed authenticity detection using frequency features. Gao et al. (2024) simultaneously detected the texture and artifacts in forged images. However, as generation techniques advance, these artifacts may have disappeared.

In terms of network design, Zhou et al. (2017) proposed a classic dual-stream network for deepfake detection. Afchar et al. (2018) introduced two compact networks to capture mesoscopic features. Zhao et al. (2021b) proposed a deepfake detection method based on multi-attention. Wang et al. (2024) utilized a knowledge distillation framework to improve the detection performance on low-quality compressed images. These methods emphasize the power of representation and computational efficiency but do not explicitly consider the generalization ability.

2.4. Auxiliary tasks or synthetic data

Some methods attempt to improve the generalization of models by leveraging additional tasks or generating representative fake samples. For example, Nguyen et al. (2019) use the multi-task learning approach to simultaneously detect manipulated images or videos, and locate the manipulated regions for each sample. Dang et al. (2020) proposed to jointly predict the binary label and the attention map of the manipulated region. Cao et al. (2022) employed an encoder–decoder reconstruction network to perform representation learning by reconstructing the difference as a guide for deepfake detection. Yang and Lim (2020) generated samples with similar distributions to those in the target domain. These methods require additional annotations or extra data samples. Le and Woo (2023) proposed a general model-internal collaborative learning framework capable of detecting deepfakes of different qualities simultaneously. Shiohara and Yamasaki (2022) utilized self-blending images to learn more general and harder-to-identify fake samples, making the classifier more generalized and robust. While these methods have shown some progress in enhancing generalization, there is still ample opportunity for further improvement. This has motivated us to delve into the exploration of new auxiliary tasks that can effectively bolster the generalization and robustness of the detector, to address the challenges posed by complex and diverse forged images.

Although the generalization of the above three types of detection methods is gradually increasing, existing methods still lack generalization and robustness, making it difficult to cope with increasingly complex forgery technology. We find that existing methods lack an in-depth study of the internal characteristics of various types of forgeries. They overlook the significant supportive role that different forgery types play in the task of deepfake detection.

In this paper, we propose a novel model training scheme. We use multitask detection of different types of forgeries as an auxiliary task, and in the latter half of training, we freeze it to focus more on improving the main task. This fine-grained learning enables the model to be used more precisely for deepfake detection tasks, resulting in better generalization and robustness.

3. Proposed method

3.1. Overview

Aiming to solve the performance degradation issue of previous methods when encountering unseen forgery patterns, we propose a novel Deepfake detection method called *Multi-To-Binary (M2B)*. This method leverages multi-classification guidance to enhance binary classification. As illustrated in Fig. 2, M2B is a dual-stream network that integrates both binary and multi-classification. During the training phase, images undergo data augmentation before being inputted into both the binary classification network and the multi-classification network separately. Both networks employ the same modified Xception architecture (Rossler et al., 2019) to extract features. These features

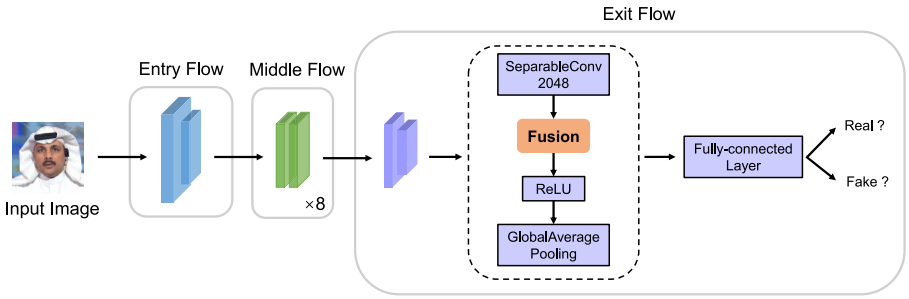


Fig. 3. The modified Xception network for classification. The entry flow and moddle flow of the Xception network are unchanged. In the exit flow, we add the fusion module after the SeparableConv 2048 layer.

are then combined through a fusion module. In the inference phase, the classifier component of the multi-classification branch is discarded. Instead, only the fused features are fed into the binary classifier to determine whether the input is real or fake. This process ensures that the binary classifier benefits from the rich feature extraction capability of the multi-classification model, enhancing its ability to generalize to unseen forgery patterns.

3.2. Classification network structure

In order to obtain comprehensive features and effectively fuse them to improve the generalization ability of the network, we propose the modified Xception (Rossler et al., 2019) network, as shown in Fig. 3. We use the same entry layers and intermediate layers as Xception, only changing the exit layer. We input the features extracted from the SeparableConv 2048 layer of both the binary classification and multi-classification networks into the fusion module. Then, the fused features are passed through the ReLU activation layer and global average pooling layer, followed by a fully connected layer to obtain the classification results. During the training phase, the differences between the binary classification network and the multi-classification network lie in the order of the features inputted into the fusion module and the number of classes in the fully connected layer. The structure of the fusion module and the used loss functions are as follows.

Fusion Module. As shown in Fig. 4, the fusion module has two inputs: the feature map F_{binary} from the binary classification network and the feature map F_{multi} from the multi-classification network. Both feature maps have dimensions of $H \times W \times C$, where $H \times W$ represents the spatial dimensions and C represents the number of channels. First, the feature maps from the two branches are concatenated together, maintaining the spatial dimensions while doubling the number of channels to obtain a feature map with dimensions of $H \times W \times 2C$. Then, the concatenated feature map is split into two branches, inspired by the ResNet network (He et al., 2016). In one branch, the feature map passes through two consecutive 3×3 convolution layers with a stride of 1 and padding of 1, to learn the non-linear mapping of the input features. In the other branch, a 1×1 convolution layer with a stride of 1 and padding of 0 matches the output dimensions with the other branch. Each convolutional layer uses C convolution kernels (filters). Here, a skip connection (not conventional) is employed, where the input features after a 1×1 convolution layer are added to the output of the convolution layers. This type of connection allows information to skip some layers and be directly passed to the subsequent layers, avoiding information loss and degradation. Finally, the features from the two branches are added together using a simple addition operation to obtain the fused feature F_{out}. The dimension of the fused feature is still $H \times W \times C$, consistent with the dimensions of the two input features.

Loss Functions. To implement the M2B framework for detection, we designed three different loss functions: binary-classification loss, multi-classification loss, and label loss that ensures consistency between the binary and multi-class predictions at the level of real and fake labels. These losses are combined into a weighted sum to create an overall loss function for training the framework.

(1) **Binary-Classification Loss.** Inspired by the promising results reported in Luo et al. (2021), our binary-classification network adopts the AM-Softmax Loss (Wang et al., 2018) as the objective function as Eq. (1). This choice is motivated by the fact that, compared to regular cross-entropy loss, the AM-Softmax Loss leads to smaller intra-class variations and larger inter-class differences.

$$\mathcal{L}_{\text{binary}} = -\frac{1}{N}\sum_{i=1}^{N}\log\left(\frac{e^{s\left(\cos\left(\theta_{y_i}\right)-m\right)}}{e^{s\left(\cos\left(\theta_{y_i}\right)-m\right)}+\sum_{j\neq y_i}e^{s\cdot\cos\left(\theta_j\right)}}\right) \tag{1}$$

where N represents the number of batch samples, y_i denotes the true class of the ith sample, s is the scale parameter, θ_{y_i} represents the angle between the feature vector of the ith sample belonging to the true class y_i and the weight vector, θ_j represents the angle between the feature vector of the ith sample and the weight vector for the non-true class j, and m is an auxiliary parameter that controls the angular margin between classes. The feature vector represents the outputs after the fusion module, while the weight vector represents the learnable parameters of the fully connected layer in the neural network. The $\cos(\cdot)$ represents the cosine similarity function, and $e^{(\cdot)}$ represents the exponential function.

(2) **Multi-Classification Loss.** We further enhance the performance of the model by employing the Focal Loss (Lin et al., 2017) on the multi-classification task to down-weight well-learned samples as Eq. (2) and Eq. (3).

$$p_t = \begin{cases} p & \text{if } y = 1, \\ 1-p & \text{otherwise,} \end{cases} \tag{2}$$

$$\mathcal{L}_{\text{multi}} = -a_t\left(1-p_t\right)^{\gamma}\log\left(p_t\right), \tag{3}$$

where $y = 1$ indicates the fake label, p_t is the prediction corresponding to the label, a_t is the balance weight, and γ is the focus modulating factor.

(3) **Label Loss.** Let $S = \left\{\left(X_i, Y_i\right)\right\}_{i=1}^{n}$ be a training dataset generated by k (k>1) different forgery methods, which includes images $X_i \in \mathbb{R}^d$ (where d represents the number of channels) and their corresponding labels $Y_i \in \{0, 1, 2, \ldots, k\}$, where 0 represents genuine images and the others represent different types of forged images. For example, in the paper, we selected the FaceForensics++ dataset (Rossler et al., 2019) as the training set, which includes images generated by four forgery methods: Deepfake (Anon, 2020), Face2Face (Thies et al., 2016), FaceSwap (Zhang et al., 2020), and NeuralTexture (Thies et al., 2019). Therefore, $k = 4$.

When performing binary classification, we transform the labels by

$$\begin{cases} Y_j = Y_i, & \text{if } Y_i = 0, \\ Y_j = 1, & \text{otherwise,} \end{cases} \tag{4}$$

where 0 means real and 1 indicates fake. In this way, we obtain the binary classification label $Y_j \in \{0, 1\}$.

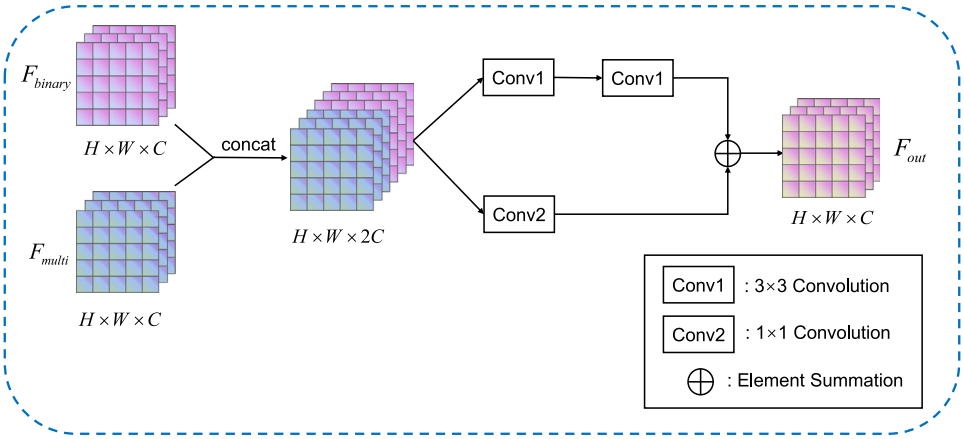


Fig. 4. Illustration of the fusion module. The fusion module utilizes a structure similar to ResBlock (He et al., 2016), where the concatenated input features are directly added to the output of the convolution layers, ensuring equal status for both features and avoiding one feature influencing the other.

During the training phase, we employ the same approach to transform the predicted values for multi-class classification. Assuming the predicted values for multi-class classification are $P_i \in \{0, 1, 2, \ldots, k\}$, and the predicted values for binary classification are $P_j \in \{0, 1\}$. To ensure the consistency between the multi-class and binary classification predictions in terms of genuine and forged dimensions, we first transform the multi-class predicted values into the genuine-forged dimension,

$$\begin{cases} P'_j = P_i, & \text{if } P_i = 0, \\ P'_j = 1, & \text{otherwise,} \end{cases} \tag{5}$$

Then, we calculate the label loss by taking the absolute difference between the predicted values for multi-class classification and binary classification.

$$\mathcal{L}_{\text{label}} = \left| P_j - P'_j \right| \tag{6}$$

It is computed over a batch of samples during training.

(4) **Overall Loss.** The final loss function of the training process is the weighted sum of the above loss functions.

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{binary}} + \lambda_1 \mathcal{L}_{\text{multi}} + \lambda_2 \mathcal{L}_{\text{label}} \tag{7}$$

where λ_1 and λ_2 are hyper-parameters for balancing the overall loss. In this paper, we set the values of λ_1 to 0.1 and λ_2 to 1. The experimental results regarding the selection of λ_1 and λ_2 will be presented in Section 4.

3.3. Freezing mechanism

We analyze that using multi-classification to guide the binary classification can enhance the detection performance of the binary task by increasing the distance between different subcategories. Therefore, we propose our M2B framework, where binary detection is considered as the main task and multi-classification detection is treated as the auxiliary task. However, excessive learning of the multi-classification network may cause the model to become confused between the main and auxiliary tasks, thereby reducing the detection performance of the main task. To avoid overfitting the multi-classification network, we propose a freezing mechanism.

Inspired by Wang et al. (2023), we introduced our freezing mechanism. Unlike AltFreezing in the (Wang et al., 2023), which alternates between freezing temporal and spatial convolutions, we did not adopt the alternating freezing strategy. Instead, we update the parameters of the multi-classification network in the first half of training and freeze them in the second half. The selection of the freezing moment is demonstrated in Section 4. This allows the binary classification network to fully leverage the knowledge provided by the multi-classification network in the first half of training and focus on improving its performance in the second half. This aligns with our main task of deepfake detection.

All in all, the whole training process of M2B can be summarized as Algorithm 1.

Algorithm 1 The Training Process of M2B

Require: training dataset D, training epochs e, binary-classification network N_b, labels y_b, multi-classification network N_m, labels y_m, hyper-parameters λ_1, λ_2.

for epoch $\in [0, e-1]$ **do**
  for batch $B \in D$ **do**
    Initialize gradient $\leftarrow 0$
    Predict the output of both networks
      $pred\_b = N_b(I)$, $pred\_m = N_m(I)$
    Compute the loss of both classification networks
      $loss\_b = AM\_SoftmaxLoss(pred\_b, y_b)$
      $loss\_m = FocalLoss(pred\_m, y_m)$
    Map $pred\_m$ to {0, 1} to obtain $pred\_m'$
    Compute the label loss
      $loss\_l = \left| pred\_m' - pred\_b \right|$
    Compute the total loss
      $loss = loss\_b + \lambda_1 * loss\_m + \lambda_2 * loss\_l$
    Compute gradient $loss.backward()$
    if epoch $< e$ / 2 **then**
      Update N_b, N_m
    else
      Update N_b
    end if
  end for
end for

4. Experiments

In this section, we present a series of experimental results that demonstrate the superior performance of the proposed method and validate our key designs. To better illustrate the results, we first provide a detailed explanation of the experimental setup. Subsequently, we compare the performance of our method with existing deepfake detection methods on three cross-dataset scenarios. Additionally, we conduct ablation studies on the constituent modules to demonstrate the effectiveness of each design. We also conduct experimental validation to select the optimal value of parameter λ_1 and λ_2. Lastly, we evaluate the robustness of the proposed method under more challenging perturbations.

Table 1
Specifications of benchmark datasets.

Dataset	Video scale	Manipulation algorithm
FF++ (Rossler et al., 2019)	1000 real, 4000 fake	DF, F2F, FS, NT
Celeb-DF (Li et al., 2020c)	408 real, 795 fake	Improved DF
DFD (Anon, 2021)	363 real, 3068 fake	Improved DF
DF-1.0 (Jiang et al., 2020)	11 000 fake	DF-VAE

4.1. Experimental settings

Datasets. To evaluate the generalization ability, we perform experiments on four large-scale benchmark datasets: **FaceForensics**++ (FF++) (Rossler et al., 2019), **Celeb-DF** (Li et al., 2020c), **DeepFakeDetection** (DFD) (Anon, 2021), and **DeeperForensics-1.0** (DF-1.0) (Jiang et al., 2020). Their detailed specifications are presented in Table 1. We mostly use FF++ as our training dataset due to its forgery diversity. It is a large-scale dataset containing 1000 real videos and 4000 synthesized videos, consisting of four types of face manipulations: Deepfake (DF), Face2Face (F2F), FaceSwap (FS) and NeuralTexture (NT). There are three versions of FF++ in terms of compression level, i.e., raw, lightly compressed (HQ), and heavily compressed (LQ). The heavier the compression level, the harder it is to distinguish the forgery traces. Since realistic forgeries always have a limited quality, we use the HQ versions in experiments. We evaluate our approach performance on Celeb-DF, DFD, and DF-1.0. Celeb-DF is a large-scale and challenging Deepfakes detection video database. The videos exhibit an extensive range of aspects, such as face sizes, lighting conditions, and backgrounds. DFD is a comprehensive collection of images and videos specifically curated for the task of deepfake detection. The dataset encompasses various types of manipulated content, including deepfake videos, face swaps, and other forms of facial manipulations. DF-1.0 is composed of diverse subjects, backgrounds, lighting conditions, and facial expressions to simulate real-world scenarios and challenges. Sample images from these datasets are shown in Fig. 5.

Evaluation metrics. We apply Accuracy (ACC), Area Under the Curve (AUC), Average Precision (AP) and Equal Error Rate (EER) as our evaluation metrics to compare our proposed method with prior works.

(1) Accuracy (ACC) calculates the ratio between the correctly predicted samples and all samples by

$$ACC = \frac{TP + TN}{n} \tag{8}$$

where TP and TN refer to the true positives and true negatives, respectively. n is the total number of images.

(2) Area Under the Curve (AUC) is a metric that measures the performance of a binary classifier by calculating the area under the receiver operating characteristic (ROC) curve.

(3) Average Precision (AP) computes the two-dimensional area under the PR curve (Recall is the horizontal axis and Precision is the vertical axis). We use this measure to evaluate the generalization ability of different forgery detection methods.

(4) Equal Error Rate (EER) is another important metric, especially in biometric recognition systems. It is the point where the false acceptance rate (FAR) equals the false rejection rate (FRR).

In this context, higher values of ACC, AUC, and AP indicate better detection performance, while a lower value of EER indicates better detection performance.

Implementation details. For the FF++ dataset, we extract the first 32 frames from each real video and the first 8 frames from each fake video of the four types of forgery methods, ensuring data balance. The remaining datasets consist of the first 32 frames extracted from the videos. The training and testing sets are split in a 3:1 ratio. We use the Xception (Rossler et al., 2019) pre-trained on the ImageNet dataset (Deng et al., 2009) as our backbone network. During the data preprocessing stage, we employ DLIB (Sagonas et al., 2016) for facial extraction and alignment. The aligned faces are resized to 224×224 for both training and testing. During training, we employed several random data augmentation methods (Shorten and Khoshgoftaar, 2019), including horizontal flipping with probability 0.5, color jitter (brightness=0.4, contrast=0.4, saturation=0.4, hue=0.1) with probability 0.8, and grayscale with probability 0.2. We train the framework using the Adam optimizer (Kingma and Ba, 2014) with a weight decay of 1e-8 and beta values of 0.9 and 0.999. The learning rate is set to 0.0001, and the batch size is 16. The value of hyperparameter λ_1 is set to 0.1, and λ_2 is set to 1. The code is in Python language using the PyTorch library. The operating system is Ubuntu. Experiments are conducted on four NVIDIA GeForce GTX 1080 Ti GPUs.

4.2. Generalization to unseen datasets

Given the rapid advancements in forgery generation techniques, it is crucial to develop a detector that can accurately classify samples from novel and previously unseen manipulation methods. To simulate this scenario, we aim to train a detector using a set of known manipulated samples and evaluate its performance on unfamiliar and unknown manipulation techniques.

To evaluate cross-dataset generalization, we trained on FF++ (all four methods) and tested on Celeb-DF, DFD, and DF-1.0. We present the results of our method and other methods: (1) **Xception** model proposed by Rossler et al. (2019), (2) **F3Net** (Qian et al., 2020): a face forgery detection by mining frequency-aware clues, (3) **ETD** (Ba et al., 2024): a deepfake detection method that extracts multiple non-overlapping local representations, (4) **SBI** (Shiohara and Yamasaki, 2022): a self-blended method using real images only during training, (5) **MADD** (Zhao et al., 2021b): a multi-attention deepfake detector, (6) **QAD** (Le and Woo, 2023): a deepfake detector with intra-model collaborative learning, (7) **RECCE** (Cao et al., 2022): a deepfake detection method with end-to-end reconstruction-classification learning.

Table 2 presents the ACC (%), AUC (%) AP (%) and EER (%) results when training on the FF++ dataset and testing on Celeb-DF, DFD, and DF-1.0 datasets. ↑ indicates that a higher value signifies better performance, while ↓ indicates that a lower value signifies better performance. The "Avg." column represents the average values across all datasets. † and * denote results obtained from training with publicly available code and pre-trained weights, respectively. The best results are highlighted in bold. M2B has surpassed other methods in most scenarios. On the Celeb-DF dataset, we achieved the best AP of 87.65%. On the DFD dataset, we achieved the best EER of 26.71%. On the DF-1.0 dataset and on average, we achieve the best performance across four metrics. Specifically, on the DF-1.0 dataset, we achieved the best ACC of 72.17% and the best AP of 88.83%. On average, we achieved the best performance with an average AUC of 75.53%, surpassing the second-best result by 6.05%. Furthermore, we obtained an average AP of 90.94%, surpassing the second-best result by a significant margin of 3.73%. These results demonstrate that our method consistently outperforms other approaches, providing promise for its performance on future improved forgeries. These findings provide additional strong evidence that exploring the distribution of different forgery techniques is crucial for generalization.

4.3. Generalization to unseen manipulations

For a general face forgery detector, it typically does not know what manipulations have been performed on the tested data. It is important to have strong generalization capabilities for unseen manipulations.

Fig. 5. Sample images from four datasets we use.

Table 2
Cross-dataset generalization. ACC (%), AUC (%) AP (%) and EER (%) results on Celeb-DF, DFD, and DF-1.0 when trained on FF++. The best results are highlighted in bold. * and † indicate results were obtained from methods' pre-trained weights and published code, respectively.

Method		Xception† (Rossler et al., 2019)	F3Net† (Qian et al., 2020)	ETD† (Ba et al., 2024)	SBI* (Shiohara and Yamasaki, 2022)	MADD* (Zhao et al., 2021b)	QAD* (Le and Woo, 2023)	RECCE* (Cao et al., 2022)	M2B (Ours)
Celeb-DF	ACC ↑	79.54	77.69	80.66	63.77	**82.68**	81.93	80.12	81.76
	AUC ↑	56.69	45.31	47.38	49.93	58.28	**60.95**	55.09	59.59
	AP ↑	85.11	81.05	81.52	82.76	84.16	86.72	84.67	**87.65**
	EER ↓	45.15	54.01	51.99	50.10	43.85	**42.41**	46.14	43.18
DFD	ACC ↑	66.81	**88.36**	85.18	72.04	86.56	85.10	87.98	85.79
	AUC ↑	65.24	79.16	76.69	53.52	71.43	67.05	**79.90**	78.98
	AP ↑	94.20	96.79	95.51	91.39	94.70	93.76	**97.12**	96.34
	EER ↓	38.76	27.57	28.84	48.20	33.22	35.61	28.02	**26.71**
DF-1.0	ACC ↑	54.27	65.75	67.36	50.51	66.69	55.56	59.67	**72.17**
	AUC ↑	62.38	83.96	78.25	51.35	71.20	57.75	63.14	**88.03**
	AP ↑	59.82	83.79	79.40	51.37	65.59	56.63	63.31	**88.83**
	EER ↓	40.51	23.08	27.21	48.93	34.88	44.43	40.51	**20.28**
Avg.	ACC ↑	66.87	77.27	77.73	62.11	78.64	74.20	75.92	**79.91**
	AUC ↑	61.44	69.48	67.44	51.60	66.97	61.92	66.04	**75.53**
	AP ↑	79.71	87.21	85.48	75.17	81.48	79.04	81.70	**90.94**
	EER ↓	41.47	34.89	36.01	49.08	37.32	40.82	38.22	**30.06**

Following previous work (Wang et al., 2023), we conduct leave-one-out experiments on FF++ dataset. There are four types of manipulation methods in FF++ dataset (DF, F2F, FS, NT). We select three of these forgery subsets as the training set, while the remaining subset is used to evaluate the model's generalization ability.

Table 3 presents the leave-one-out experimental results of our proposed method in comparison with several baseline approaches. Our method achieves the highest performance on both DF and NT datasets, while F3Net obtains the best results on the F2F dataset. Although our method exhibits slightly inferior performance on certain metrics for FS, it still attains the highest overall average accuracy, with an ACC of 81.67%, an AUC of 70.05%, an AP of 42.61%, and an EER of 35.18%. These results demonstrate that our method generalizes well across different types of manipulations.

Table 3
Cross-manipulation generalization. We report the ACC (%), AUC (%) AP (%) and EER (%) results on the FF++ dataset, which consists of four manipulation methods (DF, F2F, FS, NT). The experiments obey the leave-one-out rule as (Wang et al., 2023). The three subsets of fake videos are used as the training data, the other one serves as the testing data.

Method	Metrics	Train on remaining three				
		DF	F2F	FS	NT	Avg.
Xception (Rossler et al., 2019)	ACC ↑	80.95	66.91	51.84	61.71	65.35
	AUC ↑	51.71	54.41	44.65	56.38	51.79
	AP ↑	19.40	20.07	16.91	23.07	19.86
	EER ↓	49.16	46.34	**53.80**	45.28	48.65
F3Net (Qian et al., 2020)	ACC ↑	83.02	**82.26**	75.54	80.62	80.36
	AUC ↑	78.84	**78.89**	**44.66**	73.72	69.03
	AP ↑	51.87	**50.10**	13.78	46.53	40.57
	EER ↓	28.30	**28.13**	55.86	32.91	36.30
ETD (Ba et al., 2024)	ACC ↑	75.74	79.08	74.35	80.26	77.36
	AUC ↑	70.72	71.95	41.45	71.43	63.89
	AP ↑	32.14	38.22	15.93	39.98	31.57
	EER ↓	35.25	34.38	55.06	33.82	39.63
M2B (Ours)	ACC ↑	**87.37**	80.73	**76.68**	**81.90**	**81.67**
	AUC ↑	**89.99**	67.51	44.17	**78.51**	**70.05**
	AP ↑	**72.78**	32.56	**17.42**	**47.69**	**42.61**
	EER ↓	**18.65**	38.22	54.75	**29.11**	**35.18**

However, it should be noted that the average AP is only 42.61%. We attribute this relatively low value to data imbalance in the training datasets. While AUC reflects the classifier's performance across various thresholds and is relatively robust to imbalanced data, AP is highly sensitive to such imbalance, as the precision and recall for minority classes are more easily affected. This suggests that our method, like many others, is susceptible to the effects of data imbalance. Therefore, ensuring a balanced dataset is particularly important for improving average precision in future research.

4.4. Ablation study

Effect of Different Components. To confirm the effectiveness of our model, we evaluated how the freezing mechanism (FE), label loss (LL), and fusion module (FU) affect the generalization of the model. We trained the model on the FF++ dataset and tested the performance on Celeb-DF, DFD, and DF-1.0 datasets. Specifically, we develop the following variants: (a) the baseline model which follows the classic image classification pipeline, *i.e.*, Xception (Rossler et al., 2019), (b) the baseline model equipped with the proposed FE, (c) the baseline model equipped with the proposed LL, (d) the proposed method without FU, and (e) our proposed method with FE, LL, and FU. As shown in Table 4, the baseline model performed poorly on all three datasets, exhibiting the worst results. Comparing variants (a) and (b), we can observe that the proposed freezing mechanism yields an average AP gain of 2.22%. We argue that the freezing mechanism can assist the model in focusing more on the primary classification task and improving the model's generalization capability. Furthermore, from variants (a) and (c), we can also observe improvements in the average AP metric when incorporating the label loss. We extensively integrated the binary classification model and the multi-classification model, while also enhancing the consistency between the two models. Merging variants (b) and (c) further improved the cross-dataset evaluation capability, enabling better performance across different datasets. The optimal performance is achieved when combining all the proposed components with an average AP of 90.94%, which represents a 3.26% improvement over the baseline model. Overall, these experimental results validate the effectiveness of different components in our model.

Influence of the value of λ_1 and λ_2. We then study the influence of the value of λ_1 and λ_2 on generalization. We trained the model on the FF++ dataset and tested the performance on Celeb-DF, DFD, and DF-1.0 datasets. When selecting the best choice according to the experiments for λ_1, we randomly set λ_2 to 1. The results, as shown in Table 5, suggest the existence of a performance peak for different choices of λ_1. The table reveals that when λ_1 is either too large ($\lambda_1 = 10$) or too small ($\lambda_1 = 0.01$), the model's performance is limited. Our experiments indicate that the optimal choice for λ_1 is 0.1, with an average AP of 90.94%. Then we set λ_1 to 0.1 and select the best choice according to the experiments for λ_2. As shown in Table 6, when $\lambda_2 = 1$ there exists a performance peak. In conclusion, we select λ_1 as 0.1 and λ_2 as 1 to achieve the best performance.

Influence of the freezing moment. In order to select the appropriate freezing moment, we freeze the parameters of the multi-classification network at 1/4, 1/2, and 3/4 of the training epochs, respectively. The experimental results are shown in Table 7. The experimental results show that the average cross-dataset AP (%) is highest at 1/2. We analyze the possible reasons to be that at 1/2, the binary classification network can fully learn the knowledge from the multi-classification network while preventing overfitting of the multi-classification network. Therefore, we choose to freeze the parameters of the multi-classification network at the midpoint of the training epochs.

Impact of the number of forgery data categories. We investigated the impact of the number of forgery data categories. In previous experiments, we used the FF++ dataset with $k = 4$ forgery data categories. Building upon this, we employed the I2G method mentioned in Zhao et al. (2021a), which uses real data from the FF++ dataset to generate forgery data. We incorporated the generated forgery data as a new category into the training set, resulting in $k = 5$ forgery data categories. It is worth noting that this new category of forgery data differs from the existing forgery methods, ensuring fairness in cross-dataset experiments. The ACC (%) results before and after incorporating the I2G data are shown in Table 8. We can observe that the performance of the method significantly improves after incorporating the I2G data. The accuracy on the DF-1.0 dataset even increased by 18.37%. This indicates that as the number of forgery data categories increases, the two-classification benefits more from the guidance of multi-classification.

Influence of the choice of Binary-Classification Loss. We conduct experiments on different choices of Binary-Classification Loss. We test both Cross Entropy Loss and AM-Softmax Loss. As shown in Table 9, when using Cross Entropy Loss, the average ACC on Celeb-DF, DFD, and DF-1.0 is 79.12%, with an average AP of 89.87%. In contrast, when using AM-Softmax Loss, the average ACC across the three datasets is 79.91%, with an average AP of 90.94%, both of which are higher than the former. Therefore, we choose AM-Softmax Loss as the Binary-Classification Loss.

Table 4
Experimental results for the effect of different components of our model. Each model was trained on FF++ and tested on Celeb-DF, DFD, and DF-1.0. "Avg." represents the average AP for cross-datasets. Our model shows a significant improvement in cross-dataset evaluation.

ID	FE	LL	FU	Testing AP (%)			
				Celeb-DF	DFD	DF-1.0	Avg.
(*a*)	×	×	×	82.05	95.98	85.02	87.68
(*b*)	✓	×	×	84.49	96.01	**89.19**	89.90
(*c*)	×	✓	×	81.64	97.02	89.09	89.25
(*d*)	✓	✓	×	85.31	**97.14**	88.46	90.30
(*e*)	✓	✓	✓	**87.65**	96.34	88.83	**90.94**

Table 5
Influence of different value of λ_1. The other hyper-parameter λ_2 is set to be 1 in this setting.

Training dataset	Method	Testing AP (%)			
		Celeb-DF	DFD	DF-1.0	Avg.
FF++	$\lambda_1 = 0.01$	83.36	95.90	**90.12**	89.79
	$\lambda_1 = 0.1$	**87.65**	96.34	88.83	**90.94**
	$\lambda_1 = 1$	85.98	**97.61**	88.70	90.76
	$\lambda_1 = 10$	81.12	97.19	87.70	88.67

Table 6
Influence of different value of λ_2. The other hyper-parameter λ_1 is set to be 0.1 in this setting.

Training dataset	Method	Testing AP (%)			
		Celeb-DF	DFD	DF-1.0	Avg.
FF++	$\lambda_2 = 0.1$	85.20	97.28	87.83	90.10
	$\lambda_2 = 1$	**87.65**	96.34	**88.83**	**90.94**
	$\lambda_2 = 10$	85.24	**97.48**	87.89	90.20

Table 7
Influence of the freezing moment. We froze the parameter updates of the multi-classification network at 1/4, 1/2 and 3/4 of the total training epochs, respectively.

Freezing moment	Testing AP (%)			
	Celeb-DF	DFD	DF-1.0	Avg.
1/4	85.58	95.92	88.06	89.85
1/2	**87.65**	96.34	88.83	**90.94**
3/4	83.52	**97.41**	**91.31**	90.75

Table 8
Impact of the number of forgery data categories. On the basis of FF++, we add the I2G forgery category. It can be seen that the model works better with the addition of forged categories.

Training dataset	Testing dataset		
	Celeb-DF	DFD	DF-1.0
FF++	81.76	85.79	72.17
FF++ & I2G	**82.80**	**88.61**	**90.54**

Table 9
Influence of the choice of Binary-Classification Loss.

Binary-classification loss	Testing dataset							
	Celeb-DF		DFD		DF-1.0		Avg.	
	ACC	AP	ACC	AP	ACC	AP	ACC	AP
Cross Entropy Loss	**82.14**	83.78	**85.98**	**96.82**	69.23	**89.02**	79.12	89.87
AM-Softmax Loss	81.76	**87.65**	85.79	96.34	**72.17**	88.83	**79.91**	**90.94**

4.5. Robustness to unseen perturbations

Given the ubiquitous image manipulation operations on social media, it is crucial for deployed deepfake detectors to be resistant to common perturbations. We evaluate the robustness of the detector by training it on the FF++ dataset and testing it on various unseen corrupted DFD samples. We consider five severity levels of the operations mentioned in Jiang et al. (2020): saturation changes, contrast changes, block-wise distortions, and JPEG compression. Fig. 6 illustrates examples of each type of corruption at five different severity levels.

In Table 10, we provide the average ACC score changes (compared to clean data) on DFD dataset when images are exposed to various corruptions, averaged over all severity levels. "Avg." denotes the mean across all corruptions (and all severity levels). M2B has ACC changes within 5.62% for every perturbation, with an average change of 1.13%.

Table 10
Average robustness to unseen corruptions. ACC (%) changes compared to clean data on DFD when images are exposed to various corruptions, averaged over all severity levels. "Avg." denotes the mean across all corruptions (and all severity levels).

Method	Clean	Saturation	Contrast	Block	JPEG	Avg.
Xception Rossler et al. (2019)	66.81	-0.16	-0.38	+0.00	-8.59	-2.28
F3Net Qian et al. (2020)	88.36	-2.95	+2.14	**+0.95**	-5.36	-1.31
ETD Ba et al. (2024)	85.18	-1.08	+0.94	+0.00	-25.28	-6.36
SBI Shiohara and Yamasaki (2022)	72.04	-2.93	-33.73	-1.01	**-0.33**	-9.50
MADDZhao et al. (2021b)	86.56	-3.53	-0.48	-20.89	-2.43	-6.83
QAD Le and Woo (2023)	85.10	-9.82	**+4.67**	+0.00	-12.32	-4.37
RECCE Cao et al. (2022)	87.98	-0.22	+2.03	-11.78	-12.88	-5.71
M2B (Ours)	85.79	**-0.06**	+1.16	+0.00	-5.61	**-1.13**

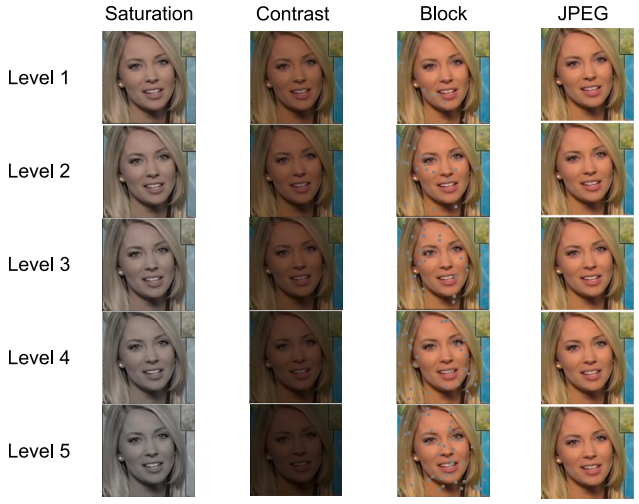


Fig. 6. Perturbations at all severity levels. This set of corruptions was introduced in Jiang et al. (2020). It consists of changes in saturation, contrast, block-wise and JPEG compression distortions. Visualization of the four perturbation types we used in our robustness experiments at all five severity levels.

Some methods show significant fluctuations for certain perturbations, for example, SBI decreases by 33.73% for contrast changes, MADD and RECCE decrease by 20.89% and 11.78% respectively for block-wise distortions. It is evident that our method exhibits significantly higher robustness to most perturbations compared to other methods. In Fig. 7, we demonstrate the effect of increasing the severity of each perturbation. Since different methods have varying initial detection accuracy for clean data, we focus on the steadiness of the curves rather than the numerical values. In the average results, a variance plot of different methods is placed. According to the steadiness of the curves, our method remains relatively stable under various perturbations, indicating its excellent robustness.

5. Conclusion

In conclusion, we have presented a novel deepfake detection method named Multi-To-Binary (M2B), which introduces the concept of multi-class features guidance to enhance the learning of fine-grained characteristics. Our approach addresses two critical objectives in face forgery detection: achieving exceptional generalization to unseen types of forgeries and enhancing robustness against various common perturbations. These objectives are crucial in combating the proliferation of fake images in today's digital landscape. By leveraging the guidance provided by multi-class features, our method exhibits remarkable generalization capabilities. It effectively adapts to previously unseen types of forgeries, enabling accurate detection even in scenarios where the manipulation patterns are unfamiliar. This ability to detect unknown patterns is a significant advancement over traditional methods that primarily focus on known forgery patterns. Moreover, our method demonstrates enhanced robustness against various common perturbations. It can effectively detect deepfakes even when subjected to typical distortions. This resilience is crucial in real-world scenarios where manipulated images may undergo unintentional alterations or deliberate efforts to deceive detection algorithms. Our method has certain limitations as the training data must include multiple forgery categories. However, considering the diversity of existing forgery techniques, collecting data with various types of manipulations can be relatively feasible.

We firmly believe that our work represents an important step forward in the fight against fake images. By leveraging the guidance of multi-class features and achieving both exceptional generalization and robustness, our method offers a reliable and effective solution for deepfake detection. We anticipate that our proposed approach will contribute to the development of more advanced and reliable techniques to combat the ever-evolving challenges posed by digitally manipulated content.

CRediT authorship contribution statement

Fei Wang: Writing – original draft, Methodology. **Bo Wang:** Writing – review & editing, Data curation. **Botao Jing:** Writing – original draft, Visualization. **Wei Wang:** Writing – review & editing, Validation. **Fei Wei:** Writing – review & editing. **Junxin Chen:** Writing – review & editing.

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.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (No. 62076052, No. 62106037), and the Application Fundamental Research Project of Liaoning Province, China (2022JH2/101300262).

Data availability

The authors do not have permission to share data.

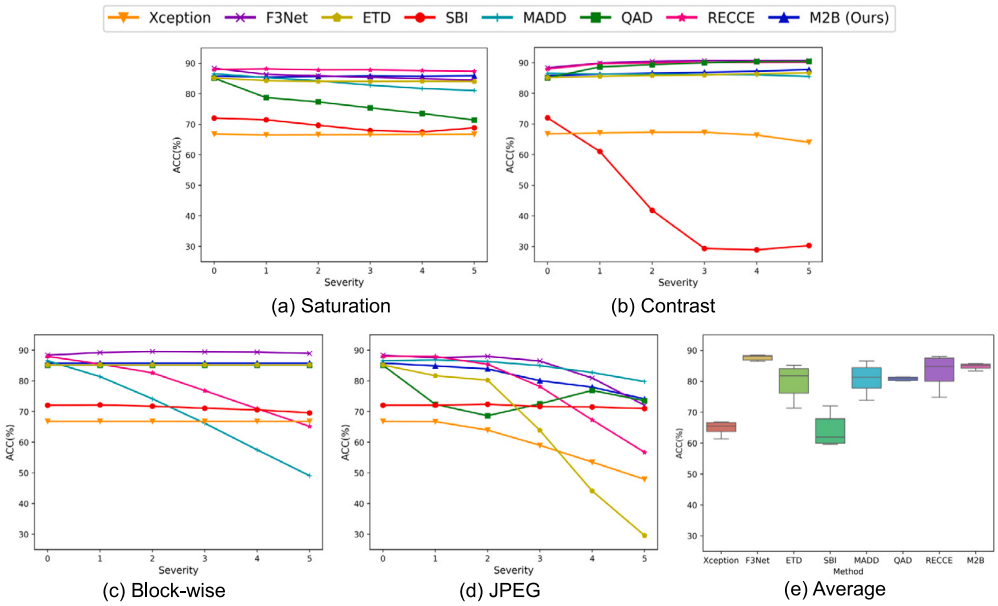


Fig. 7. Robustness to various unseen corruptions. ACC scores as a function of the severity level for various corruptions. "Average" denotes the mean across all corruptions at each severity level. M2B is more robust than previous methods in dealing with most corruptions.

References

Afchar, D., Nozick, V., Yamagishi, J., Echizen, I., 2018. Mesonet: a compact facial video forgery detection network. In: 2018 IEEE International Workshop on Information Forensics and Security. WIFS, IEEE, pp. 1–7.

Amerini, I., Ballan, L., Caldelli, R., Del Bimbo, A., Serra, G., 2011. A sift-based forensic method for copy–move attack detection and transformation recovery. IEEE Trans. Inf. Forensics Secur. 6 (3), 1099–1110.

Anon, 2020. Deepfakes. https://github.com/deepfakes/faceswap, (Accessed 02 September 2020).

Anon, 2021. Contributing data to deepfake detection research. https://ai.googleblog.com/2019/09/contributing-data-to-deepfake-detection.html, (Accessed 13 November 2021).

Ba, Z., Liu, Q., Liu, Z., Wu, S., Lin, F., Lu, L., Ren, K., 2024. Exposing the deception: Uncovering more forgery clues for deepfake detection. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 38, (2), pp. 719–728.

Cao, J., Ma, C., Yao, T., Chen, S., Ding, S., Yang, X., 2022. End-to-end reconstruction-classification learning for face forgery detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4113–4122.

Chen, S., Yao, T., Chen, Y., Ding, S., Li, J., Ji, R., 2021. Local relation learning for face forgery detection. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 35, (2), pp. 1081–1088.

Ciftci, U.A., Demir, I., Yin, L., 2020. Fakecatcher: Detection of synthetic portrait videos using biological signals. IEEE Trans. Pattern Anal. Mach. Intell..

Clark, A., Donahue, J., Simonyan, K., 2019. Adversarial video generation on complex datasets. arXiv preprint arXiv:1907.06571.

Dang, H., Liu, F., Stehouwer, J., Liu, X., Jain, A.K., 2020. On the detection of digital face manipulation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5781–5790.

De Carvalho, T.J., Riess, C., Angelopoulou, E., Pedrini, H., de Rezende Rocha, A., 2013. Exposing digital image forgeries by illumination color classification. IEEE Trans. Inf. Forensics Secur. 8 (7), 1182–1194.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., Fei-Fei, L., 2009. Imagenet: A large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition. Ieee, pp. 248–255.

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

Farid, H., 2009. Image forgery detection. IEEE Signal Process. Mag. 26 (2), 16–25.

Fernandes, S., Raj, S., Ortiz, E., Vintila, I., Salter, M., Urosevic, G., Jha, S., 2019. Predicting heart rate variations of deepfake videos using neural ode. In: Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops.

Gao, J., Micheletto, M., Orrù, G., Concas, S., Feng, X., Marcialis, G.L., Roli, F., 2024. Texture and artifact decomposition for improving generalization in deep-learning-based deepfake detection. Eng. Appl. Artif. Intell. 133, 108450.

Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y., 2014. Generative adversarial nets. Adv. Neural Inf. Process. Syst. 27.

Gu, Q., Chen, S., Yao, T., Chen, Y., Ding, S., Yi, R., 2022a. Exploiting fine-grained face forgery clues via progressive enhancement learning. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 36, (1), pp. 735–743.

Gu, Z., Chen, Y., Yao, T., Ding, S., Li, J., Ma, L., 2022b. Delving into the local: Dynamic inconsistency learning for deepfake video detection. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 36, (1), pp. 744–752.

Haliassos, A., Vougioukas, K., Petridis, S., Pantic, M., 2021. Lips don't lie: A generalisable and robust approach to face forgery detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5039–5049.

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 770–778.

Huang, B., Wang, Z., Yang, J., Ai, J., Zou, Q., Wang, Q., Ye, D., 2023. Implicit identity driven deepfake face swapping detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4490–4499.

Jiang, L., Li, R., Wu, W., Qian, C., Loy, C.C., 2020. Deeperforensics-1.0: A large-scale dataset for real-world face forgery detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2889–2898.

Kingma, D., Ba, J., 2014. Adam: A method for stochastic optimization. Comput. Sci..

Le, B.M., Woo, S.S., 2023. Quality-agnostic deepfake detection with intra-model collaborative learning. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 22378–22389.

Li, L., Bao, J., Yang, H., Chen, D., Wen, F., 2020a. Advancing high fidelity identity swapping for forgery detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5074–5083.

Li, L., Bao, J., Zhang, T., Yang, H., Chen, D., Wen, F., Guo, B., 2020b. Face x-ray for more general face forgery detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5001–5010.

Li, Y., Chang, M.-C., Lyu, S., 2018. In ictu oculi: Exposing ai created fake videos by detecting eye blinking. In: 2018 IEEE International Workshop on Information Forensics and Security. WIFS, IEEE, pp. 1–7.

Li, Y., Yang, X., Sun, P., Qi, H., Lyu, S., 2020c. 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.

Lin, T.-Y., Goyal, P., Girshick, R., He, K., Dollár, P., 2017. Focal loss for dense object detection. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2980–2988.

Luo, Y., Zhang, Y., Yan, J., Liu, W., 2021. Generalizing face forgery detection with high-frequency features. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 16317–16326.

Matern, F., Riess, C., Stamminger, M., 2019. Exploiting visual artifacts to expose deepfakes and face manipulations. In: 2019 IEEE Winter Applications of Computer Vision Workshops. WACVW, IEEE, pp. 83–92.

Mirsky, Y., Lee, W., 2021. The creation and detection of deepfakes: A survey. ACM Comput. Surv. 54 (1), 1–41.

Nguyen, H.H., Fang, F., Yamagishi, J., Echizen, I., 2019. Multi-task learning for detecting and segmenting manipulated facial images and videos. In: 2019 IEEE 10th International Conference on Biometrics Theory, Applications and Systems. BTAS, IEEE, pp. 1–8.

Qian, Y., Yin, G., Sheng, L., Chen, Z., Shao, J., 2020. Thinking in frequency: Face forgery detection by mining frequency-aware clues. In: European Conference on Computer Vision. Springer, pp. 86–103.

Rocha, A., Scheirer, W., Boult, T., Goldenstein, S., 2011. Vision of the unseen: Current trends and challenges in digital image and video forensics. ACM Comput. Surv. 43 (4), 1–42.

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. pp. 1–11.

Sagonas, C., Antonakos, E., Tzimiropoulos, G., Zafeiriou, S., Pantic, M., 2016. 300 faces in-the-wild challenge: Database and results. Image Vis. Comput. 47, 3–18.

Shiohara, K., Yamasaki, T., 2022. Detecting deepfakes with self-blended images. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 18720–18729.

Shorten, C., Khoshgoftaar, T.M., 2019. A survey on image data augmentation for deep learning. J. Big Data 6 (1), 1–48.

Thies, J., Elgharib, M., Tewari, A., Theobalt, C., Nießner, M., 2020. Neural voice puppetry: Audio-driven facial reenactment. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVI 16. Springer, pp. 716–731.

Thies, J., Zollhöfer, M., Nießner, M., 2019. Deferred neural rendering: Image synthesis using neural textures. Acm Trans. Graph. (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. pp. 2387–2395.

Vougioukas, K., Petridis, S., Pantic, M., 2020. Realistic speech-driven facial animation with gans. Int. J. Comput. Vis. 128 (5), 1398–1413.

Wang, Z., Bao, J., Zhou, W., Wang, W., Li, H., 2023. Altfreezing for more general video face forgery detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4129–4138.

Wang, F., Cheng, J., Liu, W., Liu, H., 2018. Additive margin softmax for face verification. IEEE Signal Process. Lett. 25 (7), 926–930.

Wang, B., Wu, X., Wang, F., Zhang, Y., Wei, F., Song, Z., 2024. Spatial-frequency feature fusion based deepfake detection through knowledge distillation. Eng. Appl. Artif. Intell. 133, 108341.

Yang, C., Lim, S.-N., 2020. One-shot domain adaptation for face generation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5921–5930.

Zhang, J., Zeng, X., Wang, M., Pan, Y., Liu, L., Liu, Y., Ding, Y., Fan, C., 2020. Freenet: Multi-identity face reenactment. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5326–5335.

Zhao, T., Xu, X., Xu, M., Ding, H., Xiong, Y., Xia, W., 2021a. Learning self-consistency for deepfake detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 15023–15033.

Zhao, H., Zhou, W., Chen, D., Wei, T., Zhang, W., Yu, N., 2021b. Multi-attentional deepfake detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2185–2194.

Zhou, P., Han, X., Morariu, V.I., Davis, L.S., 2017. Two-stream neural networks for tampered face detection. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops. CVPRW, IEEE, pp. 1831–1839.

Zhou, H., Liu, Y., Liu, Z., Luo, P., Wang, X., 2019. Talking face generation by adversarially disentangled audio-visual representation. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 33, (01), pp. 9299–9306.