Neural Computing and Applications (2024) 36:2749–2765
https://doi.org/10.1007/s00521-023-09196-3

ORIGINAL ARTICLE

An ensemble of CNNs with self-attention mechanism for DeepFake video detection

Karima Omar[1] · Rasha H. Sakr[1] · Mohammed F. Alrahmawy[1]

Received: 15 February 2023 / Accepted: 20 October 2023 / Published online: 23 November 2023

Abstract
The availability of large-scale facial datasets with the rapid progress of deep learning techniques, such as Generative Adversarial Networks, has enabled anyone to create realistic fake videos. These fake videos can potentially become harmful when used for fake news, hoaxes, and identity fraud. We propose a deep learning bagging ensemble classifier to detect manipulated faces in videos. The proposed bagging classifier uses the convolution and self-attention network (CoAtNet) model as a base learner. CoAtNet model is vertically stacking depthwise convolution layers and self-attention layers in such a way that generalization, capacity, and efficiency are improved. Depthwise convolution captures local features from faces extracted from video then pass these features to the attention layers to extract global information and efficiently capture long-range dependencies of spatial details. Each learner is trained on a different subset randomly taken of training data with a replacement then models' predictions are combined to classify the video either as real or fake. We also use CutMix data augmentation on the extracted faces to enhance the generalization and localization performance of the base learner model. Our experimental results show that our proposed method achieves higher efficiency compared to state-of-the-art methods with AUC values of 99.70%, 97.49%, 98.90%, and 87.62% on the different manipulation techniques of the FaceForensics++ dataset (DeepFakes (DF), Face2Face (F2F), FaceSwap (FS), and NeuralTextures (NT)), respectively, and 99.74% on the Celeb-DF dataset.

Keywords Generative Adversarial Networks · Bagging Ensemble · Self-Attention · CutMix · DeepFake Detection

1 Introduction

Images and videos containing forgery faces generated by DeepFake methods have recently sparked widespread public concern. The term "DeepFake" refers to swapping the face of one individual with the face of another one using deep learning techniques. Originally, the term "DeepFake" was named by a Reddit user in late 2017 when he built a machine learning algorithm to exchange the faces of celebrities [1]. The DeepFake technology has the potential to be harmful when utilized for malicious purposes such as revenge pornography, hoaxes, identity fraud, and spread of misinformation during e.g., political elections.

Recently, it has become increasingly easy to automatically manipulate a person's genuine face in an image or video and generate a fake one. There are many reasons leading to the growing DeepFakes, such as the availability of large-scale datasets and the evolution of deep learning techniques like Generative Adversarial Networks (GAN) [2] and Autoencoder (AE) models [3]. As a result, many applications have been released, such as FaceApp [4] and FaceSwap [5], which can be easily used by anyone to manipulate faces and create fake images and videos without a background. This heightens the risk and urgency associated with the problem.

There are four different types of facial manipulations: 1) entire face synthesis, 2) identity swap, 3) expression swap,

✉ Karima Omar
karima_asaad@mans.edu.eg

Rasha H. Sakr
rashah@mans.edu.eg

Mohammed F. Alrahmawy
mrahmawy@mans.edu.eg

[1] Department of Computer Science, Faculty of Computer and Information Sciences, Mansoura University, Mansoura 35516, Egypt

and 4) attribute manipulation. In entire face synthesis, which generates fully non-existent face images using the GAN [2] model, such as StyleGAN [6]. This manipulation could be useful for some disciplines such as generating 3D modeling and video games. On the other hand, it could be harmful to some applications, such as using a fake image as a profile photo on social media.

The second type is identity swap, which is commonly known as "face swap" or "face replacement". This manipulation includes altering an image or video to substitute one person's face with another's. It can be achieved through two main techniques: using traditional computer graphics techniques such as FaceSwap [5], or utilizing deep learning-based methods known as DeepFakes [7]. One example of this is the mobile app ZAO [8]. This type of manipulation could be useful, particularly in the film industry. However, it can also be utilized for harmful intentions such as financial fraud, producing pornographic videos featuring celebrities, and hoaxes.

The expression swap is the third type and sometimes referred to as "face Reenactment", which includes swapping the facial expression, gaze, mouth, and pose of one person to another person in an image or video without changing the identity. There is a widely circulated video of Mark Zuckerberg making statements he never actually made [9]. Popular techniques used to generate this type of manipulation utilize GAN [2] models such as Face2Face [10] and NeuralTextures [11].

The last type of face manipulation is attribute manipulation. It is also referred to as face retouching or face editing. This manipulation includes editing some attributes of the real face, by changing gender, adding color to hair or skin, adding or removing glasses, and so on. This manipulation is usually achieved using GAN [2] models such as StarGAN [12]. A popular example of a mobile application for this type of manipulation is FaceApp [4].

A lot of research efforts have been conducted to find methods that can identify manipulated faces. Traditional forgery detection methods are basically based on: 1) The fingerprints inside the camera by analyzing the intrinsic information recorded by the camera device such as filter array of color, compression, and the optical lens [13]. 2) The external fingerprint information provided by the editing software such as moving or copy-paste various components of the image [14].

However, another set of face forgery detection methods focuses on recognizing the artifacts that exist in the generated DeepFake images. These artifacts can be categorized into spatial and temporal artifacts. The spatial artifacts are a result of blending the generated content with the original image or anomalies in context compared to the rest of the image. Additionally, some models such as GANs [2] leave a fingerprint in the generated image, which can be traced by the detection system. The temporal artifacts can be detected by analyzing consecutive frames of the fake video to discover anomalies in behavior, inconsistencies, or lack of coherence. Moreover, some forgery detection methods are based on training a strong classifier based on traditional machine learning or deep learning techniques and letting the classifier decide which features to analyze.

In this paper, a deep learning bagging ensemble classifier is proposed to detect manipulated faces in videos. Our method uses the convolution and self-attention network (CoAtNet) model as a base learner, which consists of vertically stacked depthwise convolution and self-attention layers. The depthwise convolution captures local features from faces extracted from videos, and the attention layers extract global information and efficiently capture long-range dependencies of spatial details. We train each learner on a different subset of the training data and then combine the models' predictions to form a robust classifier. Additionally, to improve the efficiency of the ensemble model, we propose the use of CutMix data augmentation on extracted faces. The proposed method is evaluated on two different datasets, Celeb-DF and FaceForensics++. The main contributions of this paper can be summarized as:

- A deep learning-based bagging ensemble classifier is proposed to detect manipulated faces in videos.
- The ensemble classifier is based on the CoAtNet model which is a combination of depthwise convolution and self-attention layers, to improve generalization, capacity, and efficiency.
- The CutMix data augmentation is proposed to enhance the localization and generalization performance of the CoAtNet model.
- Our proposed method is evaluated on two different datasets namely, Celeb-DF and FaceForensics++ and it achieved higher efficiency compared to state-of-the-art methods.

The remaining sections of the paper are structured as follows: Sect. 2 provides an overview of related work in the field of DeepFake detection. Section 3 describes our proposed method, which includes details about the CoAtNet model and the bagging ensemble classifier with CutMix data augmentation. We present in Sect. 4 our experimental results and discussion on two datasets, namely FaceForensics++ and Celeb-DF. Lastly, we provide a concluding section and offer insights into potential future work.

2 Related work

Recently, the rapid developments in machine learning and deep learning algorithms enable researchers to develop and evaluate many methods to detect DeepFakes. In this section, we cover the most relevant approaches to detect manipulated images and videos.

Traditional machine learning methods have been used in many research works for detecting face forgery, such as Support Vector Machine (SVM) [15, 16]. For example, in [16], the authors introduced a method for detecting fake images by comparing landmark locations extracted from the original and fake images. They found that central landmarks of fake and original images are always very closest, so, they decided to use only facial landmarks to estimate the error or difference between the two images. The error values are then used as a feature vector to train an SVM classifier to detect fake images. However, this method may be sensitive to variations in the position and orientation of the facial landmarks, as small differences in landmark positions can lead to significant changes in the feature vector. In [17], the authors tested a steganography-based method [18] on the FaceForensics++ dataset to detect face forgery. The detector is built using a rich model of features obtained through various approaches and a Fish linear discriminate-based classifier. However, the results of this method were not as good as those achieved by deep learning-based methods.

On the contrary, various methods have utilized deep learning techniques, such as convolutional neural network (CNN) and capsule networks, to detect faked faces. For instance, in [19], the authors proposed a compact neural network, MesoNet, to capture mesoscopic properties of images for detecting facial video forgeries. Two network architectures, Meso-4 and MesoInception-4, are introduced, Meso-4 has a sequence of four layers of successive convolutions and pooling, followed by a dense network with one hidden layer, while MesoInception-4 used a variant of the inception module instead of the first two convolutional layers as in Meso-4. However, the proposed networks are only tested for detecting DF and F2F facial video manipulations. In [20], Rahmouni et al. proposed a method that used CNN to differentiate between real and computer-generated graphics images. The method involves splitting the input images into smaller patches and feeding them to CNN to extract feature maps. Statistical features are then computed based on the extracted feature maps, and these statistical features are utilized for distinguishing between computer-generated and genuine photographic images. However, the method may not generalize well to detect different types of DeepFake manipulations. In [21], they proposed a method that utilizes biological signals present in videos as a means of identifying authenticity. The method involves several transformations of these signals and the use of a CNN to enhance the classifier's ability to detect synthetic content. In [22], they combined a capsule network with VGG-19 for detecting fake faces, including those using printed images and replayed videos, as well as those created using deep learning. This network has demonstrated the capability to perform comparably to conventional CNN but with a much smaller number of parameters. In [23], the authors proposed a CNN method that leverages both the spatial information of the image and the phase spectrum to detect up-sampling artifacts caused by face forgery. However, this method may not be as effective if the forgery wasn't created using generative models. In [24], the authors introduced a supervised contrastive learning approach for classifying images as either fake or real. Their method involves applying two different augmentation techniques, selected randomly, to the input image to generate a pair of distinct perspectives of an identical image, which is then passed through their framework, which consists of two branches. In each branch, they use XceptionNet as an encoder to extract features, a projector to represent these features as a vector, and a predictor to generate a new vector based on this projection vector and the feature maps of the other branch, allowing the model to learn consistency. In [25], an attention technique was utilized with CNNs to identify key regions in the image and highlight them to improve the features used for classifying manipulated faces. In [26], a graph neural network (GNN) is proposed to detect real and fake faces. This method involves extracting faces from videos and dividing them into patches, where each patch represents a node in a graph. The edges are constructed using K-Nearest Neighbor (KNN), and an aggregation function is used to update edge weights through a number of iterations. The GNN is integrated into a pyramid ResNet architecture to process multiscale image attributes. However, the performance of the KNN used to build the edges can be influenced by the choice of the number of neighbors and the distance metric used, which may have an impact on the accuracy of the resulting graph.

Additionally, some methods exploit the temporal information of video frames. For example, the authors in [27] proposed employing different spatio-temporal convolutional network architectures such as Recurrent CNN (RCN), ResNet-3D (R3D), and Inflated 3D CNN (I3D) to exploit temporal information for detecting DeepFakes. In [28], the authors proposed a two-stream method for detecting DeepFakes, which involves an analysis of the compressed video's temporal-level and frame-level characteristics. They recognize that fake videos may lack temporal consistency, so they apply a temporal-level stream to extract temporal correlation features to deal with

this problem. However, these methods require large amounts of training data to learn effective representations of spatio-temporal features.

Moreover, some other detection methods use an ensemble-based learning architecture. For instance, in [29], the authors proposed an ensemble of CNN architectures (XceptionNet, XceptionNet + Attention, and EfficientNet+ Attention) by applying a supervised attention network on the original model. This network includes attention-based data augmentation and decision explanation to enhance the model's accuracy and improve generalization. They trained and evaluated their model using the DFDC dataset. In [30], the authors proposed a technique based on ensemble learning called DeepfakeStack. Their architecture consists of two levels: level 0 involves a series of deep learning models called base-learners, and level 1 is a CNN classifier called meta-learner, which is trained on the output predictions of level 0. They trained and evaluated their technique on the FaceForensics++ dataset [17]. The authors in [31] introduced DefakeHop, a method that utilizes the principle of successive subspace learning to automatically extract features from various parts of face images. These features are obtained through the Saab transform and then further processed by a feature distillation module. The final decision is obtained by integrating soft decisions from all facial regions and selected frames.

Currently, Vision Transformer (ViT) models have been proven to be effective in image classification. For instance, Heo et al. [32, 33] introduced a ViT-based model that concatenates patch embedding of input sequences with EfficientNet to extract local features. The output of this concatenated layer is then passed to ViT with a distillation method that learns global information for DeepFake detection. Similarly, in [34], the authors proposed a model consisting of a stack of convolutional blocks to extract learnable features from video frames, followed by a ViT that takes the extracted feature as input and classifies it as either fake or real. However, the examination of ViT-based models and their variations for DeepFake detection is still limited.

3 The proposed method

In recent years, there has been a growing interest in incorporating the concept of attention into computer vision, as various works have demonstrated the effectiveness of this approach [35–38]. One of the most recent development models is ViT [39]. Despite its promising results, the performance of ViT still falls behind that of CNNs [40]. This is likely due to the fact that ViT lacks the image-specific inductive bias that CNNs possess [41, 42]. CNNs are typically able to generalize and converge faster than ViT, while the latter has more model capacity.

CoAtNet model [43] combines both architectures' strengths in such a way that this combination improves generalization, capacity, and efficiency. The CoAtNet model has two main components: mobile inverted bottleneck convolution (MBConv) block [44] and self-attention with relative bias [45]. The specifics of these two components are addressed in subsequent subsections.

In this paper, we propose an ensemble of CoAtNet models with CutMix augmentation to effectively detect DeepFake videos as shown in Fig. 1. The CoAtNet combines both convolutional layers and self-attention layers with a relative bias to capture features from the input images. The convolutional layers capture local features from the images, while the self-attention layers extract global information and efficiently capture the long-range dependencies of spatial details.

3.1 MBConv block

MBConv block was proposed in [44]. The MBConv block has been shown to be effective at improving the performance of deep learning models while keeping them lightweight and efficient. The structure of the MBConv block consists of three layers. First, an expansion layer applies a (1×1) convolutional layer to the input to increase the number of channels. Typically (1×1) convolutional layers are used to combine the features across channels and control the number of channels in the output. Then, a depthwise convolution (3×3) is performed on the output of the expansion layer. Depthwise convolution is a type of convolution that differs from standard convolution. In standard convolution, a set of filters with learnable weights is applied to the entire input feature map. Each filter slides across the entire input feature map, computing a dot product between the filter weights and the input values at each location. The resulting output feature map has the same spatial dimensions as the input, but the number of output channels is determined by the number of filters used. In contrast, depthwise convolution applies a single filter per input channel. In other words, each filter in a depthwise convolution only convolves with a single channel of the input feature map. Depthwise convolution is computationally less expensive than standard convolution because it requires fewer parameters and computations.

Finally, a projection layer applies (1×1) convolutional layer to the output of the depthwise convolution layer to reduce the number of channels. This layer is used to match the number of channels in the input with the output of the block. The input and output feature maps are added using a residual connection only if they have the same dimensions and the stride used with a depthwise convolution equals

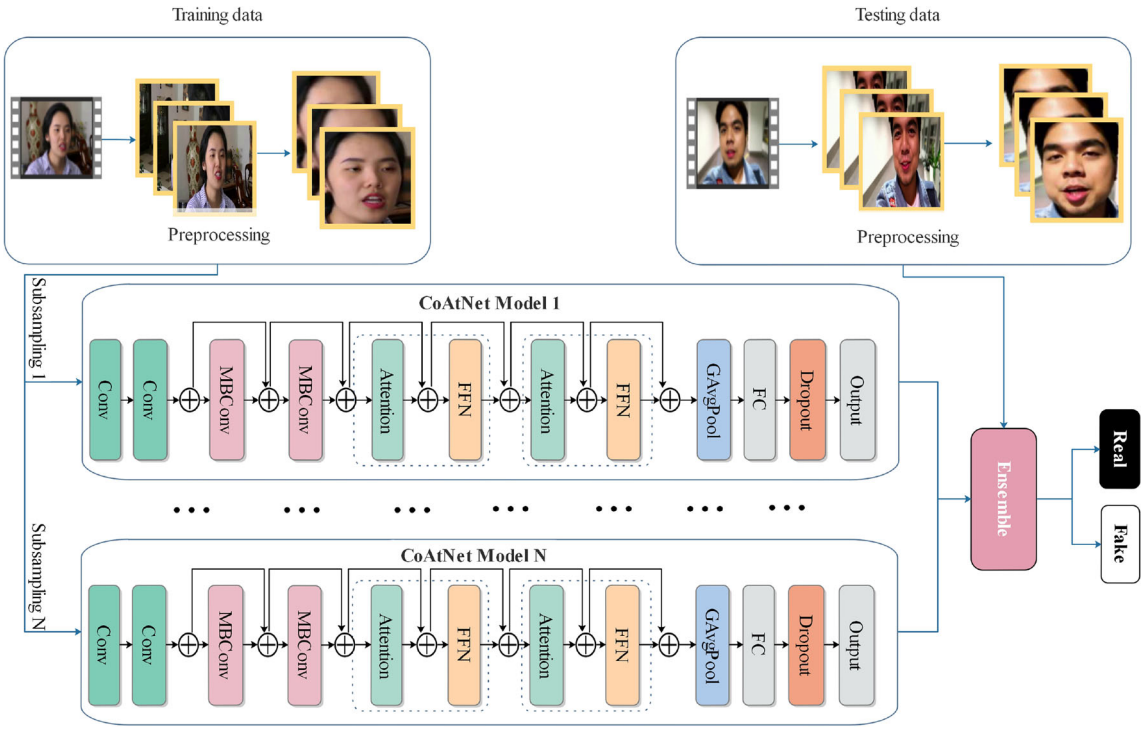

Fig. 1 An overview of the proposed method

one, as shown on the left side of Fig. 2. In the CoAtNet, the first repetition of the MBConv block uses a stride of 2 with the first convolutional layer for the downsampling process, as shown on the right side of Fig. 2.

3.2 Self-attention

In computer vision, CNNs have been the dominant model architecture [40, 46–48]. In contrast, self-attention-based architectures, in particular Transformers [45], have shown great success in natural language processing (NLP). The ViT architecture proposed in [39] uses an encoder composed of repetitive blocks of two layers: Multi-head self-attention (MSA) layer followed by a feed-forward network (FFN). The self-attention layer aggregates spatial information over patches, capturing global information and long-range dependencies by attending to relevant patches.

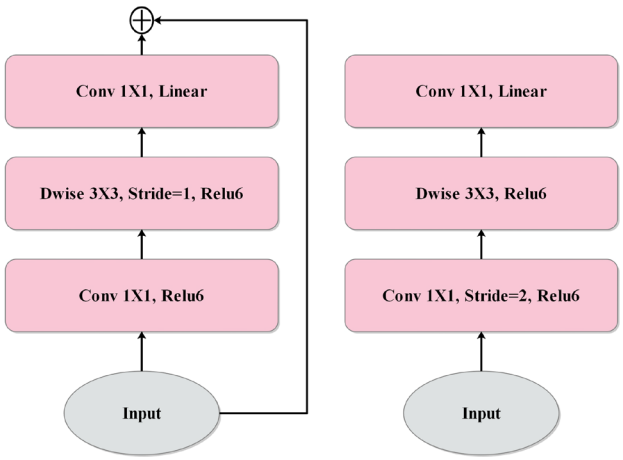

Fig. 2 The structure of the MBConv block

Consider a 2D image with feature maps $x \in \mathbb{R}^{H\times W\times C}$, where H is the height, W is the width, and C is number of feature maps. To form the input sequence x for self-attention, the image is divided into a fixed size of patches $x_p \in \mathbb{R}^{P\times P\times C}$, with the number of patches $m = WH/P^2$, where (P, P) is the size of each patch. These patches are flattened into a 1D to form the input sequence of token embeddings $x = (x_{p_1}, ..., x_{p_m})$ of m elements. Self-attention operates on this sequence and produces an output sequence $y = (y_1, ..., y_m)$, where $y_i \in \mathbb{R}^{d_y}$. The output of each element y_i is computed as a weighted sum of the linearly projected input elements:

$$y_i = \sum_{j=1}^{m} \alpha_{ij}(x_{p_j} W^V) \tag{1}$$

where each weight α_{ij} is computed by applying a softmax function:

$$\alpha_{ij} = \frac{\exp e_{ij}}{\sum_{k=1}^{m} \exp e_{ik}} \tag{2}$$

and e_{ij} computes the scaled pairwise similarity between two elements (x_{p_i}, x_{p_j}):

$$e_{ij} = \frac{(x_{p_i} W^Q)(x_{p_j} W^K)^T}{\sqrt{d_y}} \tag{3}$$

where W^Q, W^K, and W^V are all learned transformation matrices that map the input sequence into queries Q, keys K, and values V representations, respectively.

MSA extends the concept of self-attention by applying the self-attention operation h times, referred to as "heads", in parallel and projecting their concatenated outputs. MSA is employed to learn various distinctive representations of the input. MSA is defined as:

$$MSA(Q, K, V) = Concat(head_1, ..., head_h) W^O \tag{4}$$

$$head_i = SelfAttention(QW_i^Q, KW_i^K, VW_i^V) \tag{5}$$

where W_i^Q, W_i^K, and W_i^V are the projection matrices for each head i. W^O is a weight matrix that is applied to the concatenated outputs of the MSA.

The second layer of the encoder is a point-wise FFN consisting of two fully connected layers with a ReLU activation function in between. The primary function of this layer is to provide a layer of non-linear transformations that allows the model to learn more complex representations of the input sequence. The layer normalization is employed on the input of each layer (MSA and FNN) in such a way that the output of each layer is added to the input via a residual connection.

3.3 Relative attention

Relative attention is a variant of self-attention that takes into account the relative position of the elements in the input sequence. 2D relative attention is commonly used in image and video processing tasks, where the relationship between the pixels or frames is important for understanding the content of the image or video. The standard self-attention considers the absolute position [45]. Absolute positional encodings are available in a variety of formats, such as the sine and cosine functions. To generate the positional encoding vectors, a combination of sine and cosine functions with different frequencies is utilized. These vectors have the same dimension as the input embeddings. The positional encoding vectors are added to the input embeddings to inject the positional information of the input token.

There have been numerous variations for the relative position in previous works, such as [49–51]. Relative position can be split into two categories: (1) the dependent version, which depends on input embeddings; and (2) the independent version, where the encodings are independent of the input embeddings. In our proposed architecture, we use the independent version, similar to the one used in [49], where a 2D relative position w_{i-j} is added to the attention weight to achieve the property of translation equivariance. The Eq. (2) is modified to be:

$$\alpha_{ij} = \frac{\exp(e_{ij} + w_{i-j})}{\sum_{k=1}^{m} \exp(e_{ik} + w_{i-j})} \tag{6}$$

where w_{i-j} is a trainable scalar representing the relative position weight for all (i, j) pairs.

3.4 CoAtNet architecture

The CoAtNet model's architecture is composed of merging depthwise convolution with self-attention layers with relative position encoding (as discussed in the previous section). The CoAtNet model combines an MBConv block and self-attention with the relative position in Eq. (6). This combination achieves three properties namely, global receptive field, input-adaptive weighting, and translation equivariance. The architecture of CoAtNet basically consists of five levels ($L0$, $L1$, $L2$, $L3$, $L4$) namely, Conv2d, MBConv, MBConv, TFM_{Rel} and TFM_{Rel} layers respectively as shown in Fig. 3. Where Conv2d is a 2D convolutional layer, and TFM_{Rel} is a Transformer block that consists of MSA with relative position encoding followed by an FNN layer. Each level is repeated T times and the number of channels D from $L1$ to $L4$ is doubled, as shown

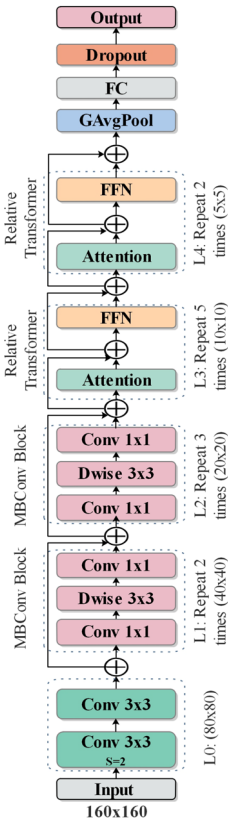

Fig. 3 The architecture of the CoAtNet model

Table 1 CoAtNet model architecture: each line represents a series of identical layers repeated T times, D is the number of channels and e is the expansion factor

Input	Operator	T	D	e
$160^2 \times 3$	Conv2d	2	64	–
$80^2 \times 64$	MBConv	2	96	4
$40^2 \times 96$	MBConv	3	192	4
$20^2 \times 192$	TFM_{Rel}	5	384	–
$10^2 \times 384$	TFM_{Rel}	2	768	–
$5^2 \times 768$	GAvgPool	1	–	
768	FC	1	512	–
512	Dropout	1	–	
512	FC (Output)	1	2	–

in Table 1. The kernel size is 3 for all Conv2d and MBConv blocks, and the size of each attention head is 32 for all the Transformer layers. Pre-activation is applied to MBConv and Transformer blocks, in which normalization is performed before the layer operation. This allows the residual connection to be defined as follows:

$$x \Leftarrow x + Layer(Norm(x)) \tag{7}$$

where *Layer* refers to self-attention, FFN, or MBConv layers. For self-attention and FFN layers, *Norm* denotes layer normalization and for MBConv blocks, *Norm* denotes batch normalization. For the first block inside each level from $L1$ to $L4$ the identity branch and the residual branch are down-sampled separately. In particular, for each Transformer block, the max-pooling of stride 2 is performed to the input states of both branches. In order to increase the hidden size, a channel projection is also performed to the identity branch:

$$x \Leftarrow Proj(MaxPool(x)) + SelfAttention(MaxPool(Norm(x))) \tag{8}$$

While for each MBConv block, a convolution of stride 2 is used to achieve the downsampling in the residual branch:

$$x \Leftarrow Proj(MaxPool(x)) + MBConv(Norm(x)) \tag{9}$$

Finally, we implement global average pooling on the output of the last level in order to simplify the representation of features. We then add a fully connected layer with 512 hidden neurons, followed by a dropout layer and the output layer.

3.5 CutMix data augmentation

CutMix [52] is an augmentation technique that utilizes the regional dropout techniques [53, 54] to enhance the localization and generalization performance of CoAtNet model by spreading the focus of the model to the entire object region not only to the most discriminative parts. The regional dropout technique removes random regions in input images. CutMix is similar to other techniques, such as MixUp and CutOut [53, 55], that alter the training images by removing or masking regions of the image. However, the main distinction of CutMix is that the removed regions are replaced with patches from another training image, rather than simply being replaced with a constant value as illustrated in Fig. 4. This allows the model to learn more robust feature representations by combining different parts of different training examples during training.

Given training images x and their corresponding labels y, the CutMix process creates new training samples by randomly cutting a section of one image (x_A, y_A) and placing it onto another image (x_B, y_B), forming a new sample (x^-, y^-). Then the CoAtNet model is trained on the newly generated samples, with the aim of introducing a wider variety of data to improve the model's robustness. The new image sample (x^-, y^-) is defined as:

$$x^- = B \odot x_A + (1 - B) \odot x_B \tag{10}$$

$$y^- = \omega y_A + (1 - \omega) y_B \tag{11}$$

where B is a binary mask that indicates where to drop and fill the pixels from the two images. Additionally, a coefficient ω is utilized to adjust the ratio of pixels being taken from each of the images and is sampled from the Beta distribution.

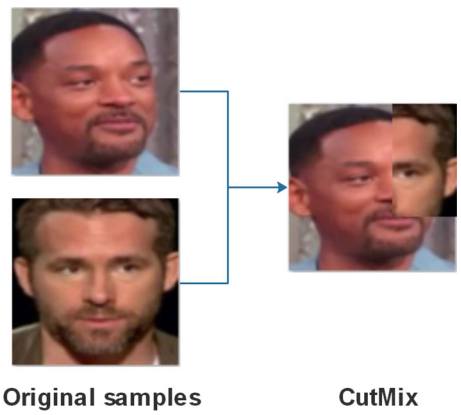

Fig. 4 An illustration of the CutMix data augmentation

3.6 Bagging ensemble

Bagging Ensemble [56] is an ensemble learning method based on bootstrap sampling. It is an effective technique for generating an ensemble of independent models, in which these models are trained in a parallel manner on a different sample of instances generated from the original data set with replacement. Replacement means that if the training data set contains D samples, we randomly select D_1 samples to generate the first subset then return the selected D_1 samples to the original data set to be re-selected again to generate the number of subsets depending on the number of bagging models.

The models' predictions could be combined using different techniques such as Majority Voting, Sum, Product rules, and stacking. The Majority Voting method works by taking a majority vote of the predictions of the base models. The Sum rule is a more powerful ensemble method, where the predictions of all base models are summed together, and the class with the highest sum is predicted. The Product rule is similar to the Sum rule, but the predictions of all base models are multiplied together before selecting the predicted class. Stacking, on the other hand, involves training a meta-model to predict the target class based on the predictions of the base models. The Majority Voting, Sum, and Product rules can be defined as follows:

$$\hat{y} = \text{argmax}_{c \in \mathcal{C}} \sum_{n=1}^{N} [\hat{y}_n = c] \tag{12}$$

$$\hat{y} = \text{argmax}_{c \in \mathcal{C}} \sum_{n=1}^{N} \hat{y}_n \tag{13}$$

$$\hat{y} = \text{argmax}_{c \in \mathcal{C}} \prod_{n=1}^{N} \hat{y}_n \tag{14}$$

where $\hat{y}_n$ is the prediction of the n^{th} base-learner, N is the number of base-learners, and $\mathcal{C}$ is the set of class labels. In this paper, we compare the performance of three ensemble techniques: Majority Voting, Sum, and Product rules. These methods are relatively simple and do not require any additional training while stacking is a more complex and computationally expensive method. The training of the bagging ensemble based on the CoAtNet model is shown in Algorithm 1.

Algorithm 1 Bagging ensemble method based on CoAtNet model

Input: Training dataset $D = \{(x_1, t_1), (x_2, t_2), ..., (x_n, t_n)\}$, the CoAtNet model M, number of models N, batch size B, number of epochs Ep
Output: Ensemble classifier E
for $i = 1$ to N **do**
  D_i = create bootstrap sample from D with replacement
  M_i = train a CoAtNet model with CutMix on D_i
end for
$E = \{M_1, \cdots, M_N\}$

4 Experiments and discussion

4.1 Datasets

In all our experiments, we used FaceForensics++ [17] and Celeb-DF [57] datasets. The FaceForensics++ dataset [17] is a large-scale collection of manipulated facial images, containing over 1.8 million images. This dataset includes 1, 000 authentic videos that were obtained from YouTube. These real videos were manipulated using four different methods, which include classical computer graphics-based approaches (Face2Face (F2F) [10], FaceSwap (FS) [5]), and deep learning-based techniques (Deepfakes (DF) [7], NeuralTextures (NT) [11]), to generate fake videos. For each method, 1, 000 fake videos were generated. The dataset also includes two compressed versions with different quality levels, namely low-quality videos (*LQ*) and high-quality videos (*HQ*). Our proposed method is evaluated on the (*LQ*) dataset. Samples of FaceForensics++ are shown in Fig. 5.

The Celeb-DF [57] is a new and challenging large-scale DeepFake video dataset, consisting of over 2 million images. It includes 590 real videos and 5, 639 DeepFake videos. The real videos were sourced from publicly available YouTube videos, featuring interviews of 59 celebrities with diverse distributions in gender, age, and ethnicity. The manipulated videos were generated using an advanced DeepFake synthesis algorithm [58, 59], which improves the visual quality of the manipulated videos by swapping the face of one person with the face of another person for each of the 59 subjects. Samples of Celeb-DF are shown in Fig. 6.

4.2 Implementation details

The FaceForensics++ dataset is divided as described in [17], with 720 videos selected for training, 140 for validation, and 140 for testing. For the Celeb-DF dataset, 518 real and fake videos are used for testing and the remaining videos are used for training the models.

To preprocess the data, we extract one frame every ten subsequent frames from both real and DeepFake videos. We use the MTCNN model [60] to detect face landmarks from the extracted frames. All the detected faces are then cropped around the center and resized to 160×160 pixels.

For the training, all the detectors namely, XceptionNet [61] and CoAtNet are pretrained on the Imagenet dataset

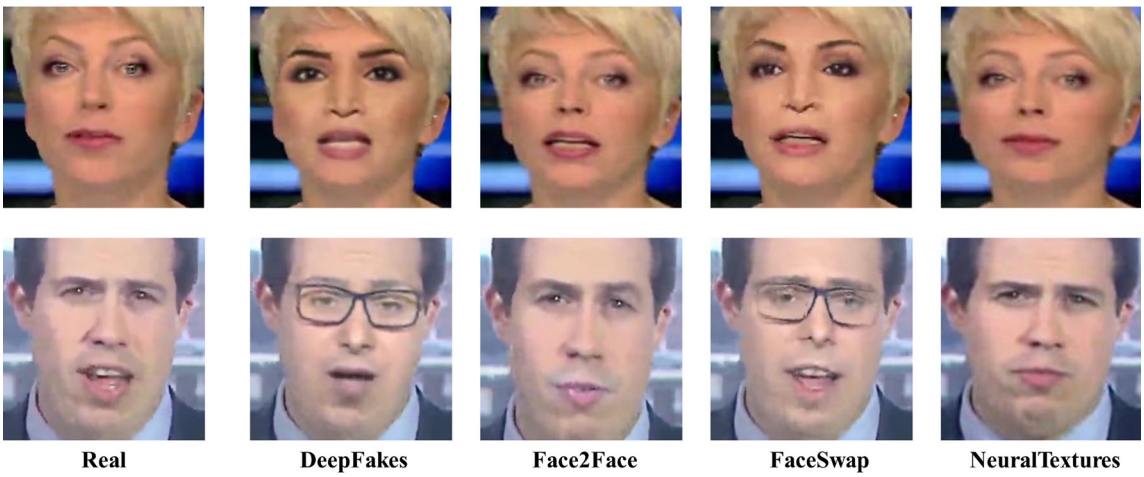

Fig. 5 Examples of FaceForensics++ dataset

Fig. 6 Examples of Celeb-DF dataset

and then the last layer is removed and we add a global average pooling layer followed by 512 fully connected layer, dropout layer, and finally the softmax output layer for deep fake detection. Each model is trained using the Adam optimizer with a learning rate of 0.0001 on the DeepFake datasets for 15 epochs. XceptionNet is a widely used model in DeepFake detection and is considered a baseline. XceptionNet is a CNN-based model that replaces the standard inception modules with a depthwise separable convolutional module. The models are trained on a single Nvidia TITAN Xp GPU with a batch size of 32. For our proposed bagging ensemble method, we train five CoAtNet models on different subsamples selected randomly from the training set and combine the models' predictions using one of three techniques: Majority Voting, Sum rule, or Product rule.

4.3 Evaluation metrics

Two common evaluation metrics used in face forgery detection are the Accuracy rate (ACC) and the Area Under Receiver Operating Characteristic Curve (AUC). Accuracy is a widely understood measure for evaluating the performance of face forgery detection systems. It is calculated as the proportion of correct predictions made by a model in relation to the total number of predictions made.

Let *TP* represent the count of correctly classified real images, *TN* represent the count of correctly classified fake images, *FP* represent the count of fake images mistakenly classified as real, and *FN* represent the count of real images mistakenly classified as fake. The Accuracy is defined as:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{15}$$

The AUC is widely used in DeepFake detection problems to measure the performance of a model that classifies examples as belonging to one of two classes, such as real and fake images. The calculation of AUC involves the plotting of a graph known as the ROC curve, which demonstrates the relationship between the true positive rate (TPR) and the false positive rate (FPR) at various thresholds. The TPR and FPR are defined as:

$$TPR = \frac{TP}{TP + FN} \tag{16}$$

$$FPR = \frac{FP}{FP + TN} \tag{17}$$

Table 2 Results on FaceForensics++ dataset for each manipulation techniques DeepFakes (DF), Face2Face (F2F), FaceSwap (FS), NeuralTextures (NT) using XceptionNet and CoAtNet models

Model	DF		F2F		FS		NT	
	ACC	AUC	ACC	AUC	ACC	AUC	ACC	AUC
XceptionNet	95.27	98.94	88.23	95.50	92.19	97.21	75.15	83.26
CoAtNet	95.81	99.12	87.55	94.71	92.95	97.71	77.33	85.32
CoAtNet + aug	95.63	99.10	87.88	95.78	93.56	**98.03**	76.44	84.82
CoAtNet + CM	**96.87**	**99.48**	**89.88**	**96.32**	**94.00**	97.99	**77.35**	**86.18**

Bold values indicate the best results for each evaluation metric on each model and higher value means better result

AUC represents the total area under the ROC curve. A perfect classifier would have a TPR of 1 and a FPR of 0, resulting in an AUC of 1.

4.4 Results on faceforensics++ dataset

Several experiments have been conducted on both FaceForensics++ and Celeb-DF datasets to evaluate the performance of the proposed DeepFake detection model. First, we compare the performance of XceptionNet and CoAtNet models in detecting different fake videos generated by different manipulation techniques on the FaceForensics++ dataset as shown in Table 2.

When comparing the performance of the XceptionNet and CoAtNet models on the FaceForensics++ dataset, it was observed that the CoAtNet model performed relatively better across all manipulations, with an average AUC of 94.22% and an average accuracy of 88.41% on the four FaceForensics++ manipulations. On the other hand, the XceptionNet model demonstrated an average AUC of 93.73% and an average accuracy of 87.71%. The CoAtNet model outperformed the XceptionNet in terms of AUC and accuracy for three out of the four manipulation techniques. For instance, on the NT manipulation technique, the CoAtNet model has an AUC and accuracy of 77.33% and 85.32% respectively, while the XceptionNet model has an AUC and accuracy of 75.15% and 83.26% respectively. The only manipulation technique where XceptionNet achieved the highest AUC and accuracy was for the F2F manipulation.

Next, we compare the performance of the proposed CutMix (CM.) augmentation technique to the traditional augmentation (aug.) techniques, using the CoAtNet model, as shown in Table 2. Traditional augmentation techniques include operations such as rotation, flipping, scaling, zooming, and shifting. The goal of these techniques is to expose the model to a wider range of variations during training, in order to improve its robustness and ability to generalize to new data. The performance of the CoAtNet model with traditional augmentation is almost similar to the performance of the CoAtNet model without augmentation except for the FS manipulation technique, where the CoAtNet model with traditional augmentation showed a slight increase in accuracy (93.56%) compared to the CoAtNet model without augmentation (92.95%). The CoAtNet model with CutMix outperforms both the CoAtNet model with traditional augmentation and the CoAtNet model without augmentation in performance. Specifically, the average AUC and accuracy of the CoAtNet model with CutMix are 95.74% and 89.53% respectively, this is compared to 94.43% and 88.38% for the CoAtNet model with traditional augmentation and 94.22% and 88.41% for the CoAtNet model without augmentation over the four manipulation techniques. These results indicate that the CutMix augmentation technique is more effective in enhancing the capability of the CoAtNet model for DeepFake detection tasks.

Table 3 reports the evaluation of different ensemble techniques of CoAtNet models with CutMix augmentation on the FaceForensics++ dataset for four different manipulation techniques. Three ensemble methods are compared: Majority Voting, Sum, and Product rules. It can be observed that the ensemble methods are capable of attaining higher AUC and accuracy scores compared to the single CoAtNet model. For example, for the F2F manipulation technique, the accuracy of the Majority Voting, Sum, and Product ensemble methods are 91.36%, 91.60%, and 91.69% respectively, compared to the 89.88% of the single CoAtNet model. For the NT manipulation technique, the accuracy of the Majority Voting, Sum, and Product ensemble methods are 79.10%, 79.43%, and 79.32% respectively, compared to the 77.35% of the single CoAtNet model. The sum and product rules are more powerful than majority voting because they take into account the confidence level of each model's prediction. For example, for the F2F manipulation technique, the accuracy of the Sum and Product ensemble methods are 91.60%, and 91.69% respectively, compared to the 91.36% of the Majority Voting.

In Fig. 7, we present the ROC curves for our proposed ensemble method using the Product rule and the baseline methods on the four manipulations techniques of the FaceForensics++ dataset. The ROC curve for the ensemble method (red curve) is closer to the top-left corner of the graph, indicating that it has a higher TPR and a lower FPR compared to the XceptionNet (blue curve), CoAtNet (orange curve), and CoAtNet + CM. (green curve). This suggests that our proposed method is better at correctly identifying manipulated videos while minimizing the number of false positives. Additionally, we can see the

Table 3 Results of the ensemble of CoAtNet models on the different manipulation techniques of the FaceForensics++ dataset (DF, F2F, FS, and NT)

Model	DF		F2F		FS		NT	
	ACC	AUC	ACC	AUC	ACC	AUC	ACC	AUC
Majority Voting	97.65	–	91.36	–	95.67	–	79.10	–
Sum Rule	97.67	99.72	91.60	97.49	95.70	98.87	79.43	87.81
Product Rule	97.70	99.70	91.69	97.49	95.74	98.90	79.32	87.62

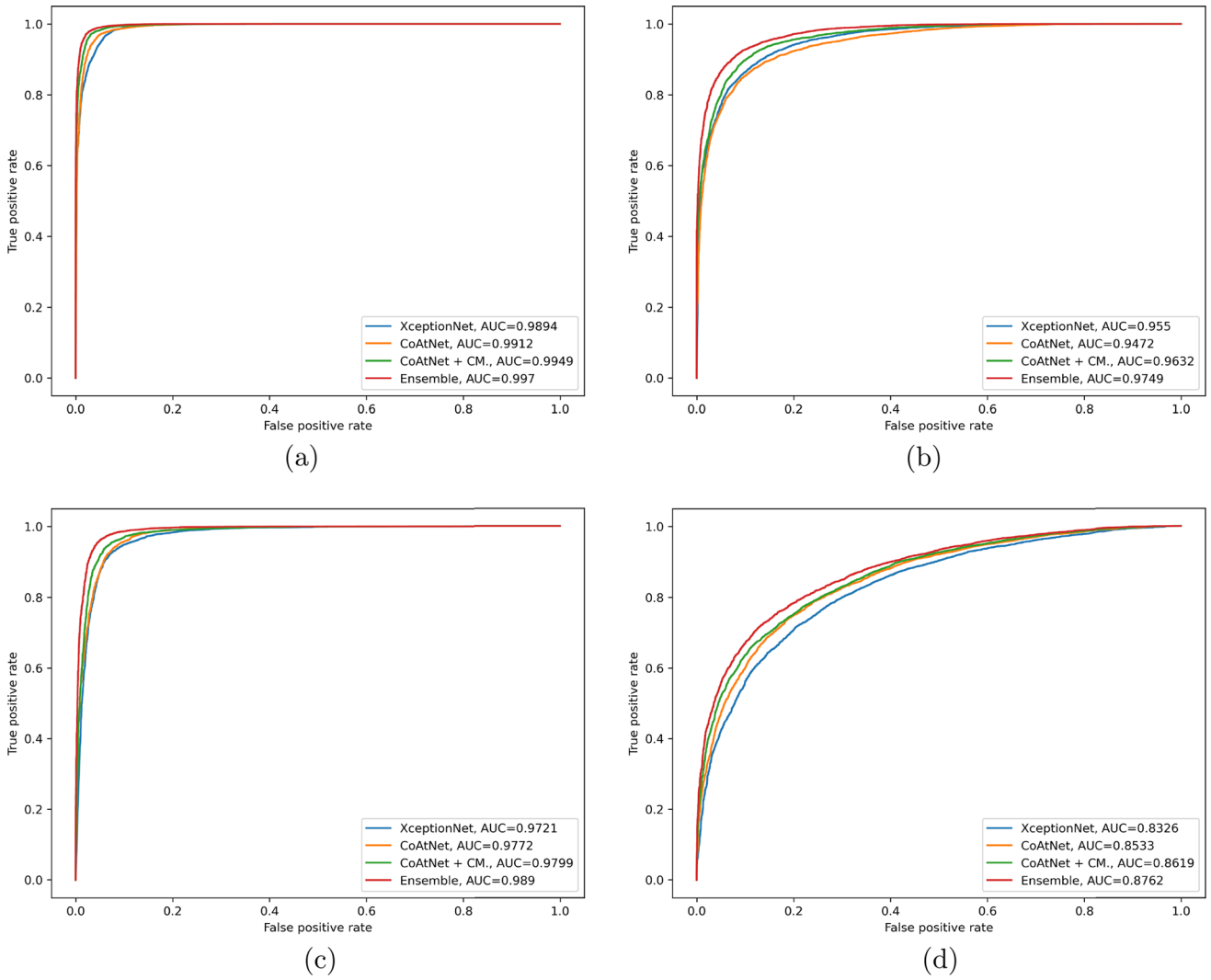

Fig. 7 Comparison of ROC curves and AUC values of the proposed ensemble method and baseline methods on the FaceForensics++ dataset. **a** DF. **b** F2F. **c** FS. **d** NT

AUC values in the figure. For example, for FS manipulation, the AUC for the ensemble method is 98.90%, while the AUC for the XceptionNet (97.21%), CoAtNet (97.71%), and CoAtNet + CM. (97.99%).

In Fig. 8, the accuracy of each individual model within the bagging ensemble is presented. For example, for the NT manipulation, the average accuracy of the ensemble is 76.90%, with a standard deviation of 0.49. The accuracy of the ensemble using the Product rule is 79.33%. For the DF manipulation, the highest accuracy of an individual model is 96.78%, while the accuracy of the ensemble using the Product rule is 97.70%. The results show that the accuracy of the ensemble is consistently higher than that of the individual models, indicating that the ensemble method is able to improve the performance of the individual models and that models within the ensemble are diverse and able to make good predictions.

The results of evaluating our proposed ensemble method against various state-of-the-art methods on different manipulation techniques (DF, F2F, FS, NT) of the FaceForensics++ dataset are presented in Table 4. The ensemble method achieves the highest accuracy and AUC for all types of manipulations compared to all other methods such as MesoNet [19], Rahmouni et al. [20], Steg. Features [18], Dong et al. [24], Capsule networks [22], Liu et al. [23], and Hu et al. [28]. The accuracy of the proposed ensemble method ranges from 79.32% for NT to 97.70% for DF, while the AUC ranges from 87.62% for NT to 99.70% for DF. For NT manipulation, the proposed method comes in second place after Hu et al. [28] method. The results indicate that the proposed ensemble method is able to improve the performance of the individual models and is an effective method for detecting manipulated faces in videos.

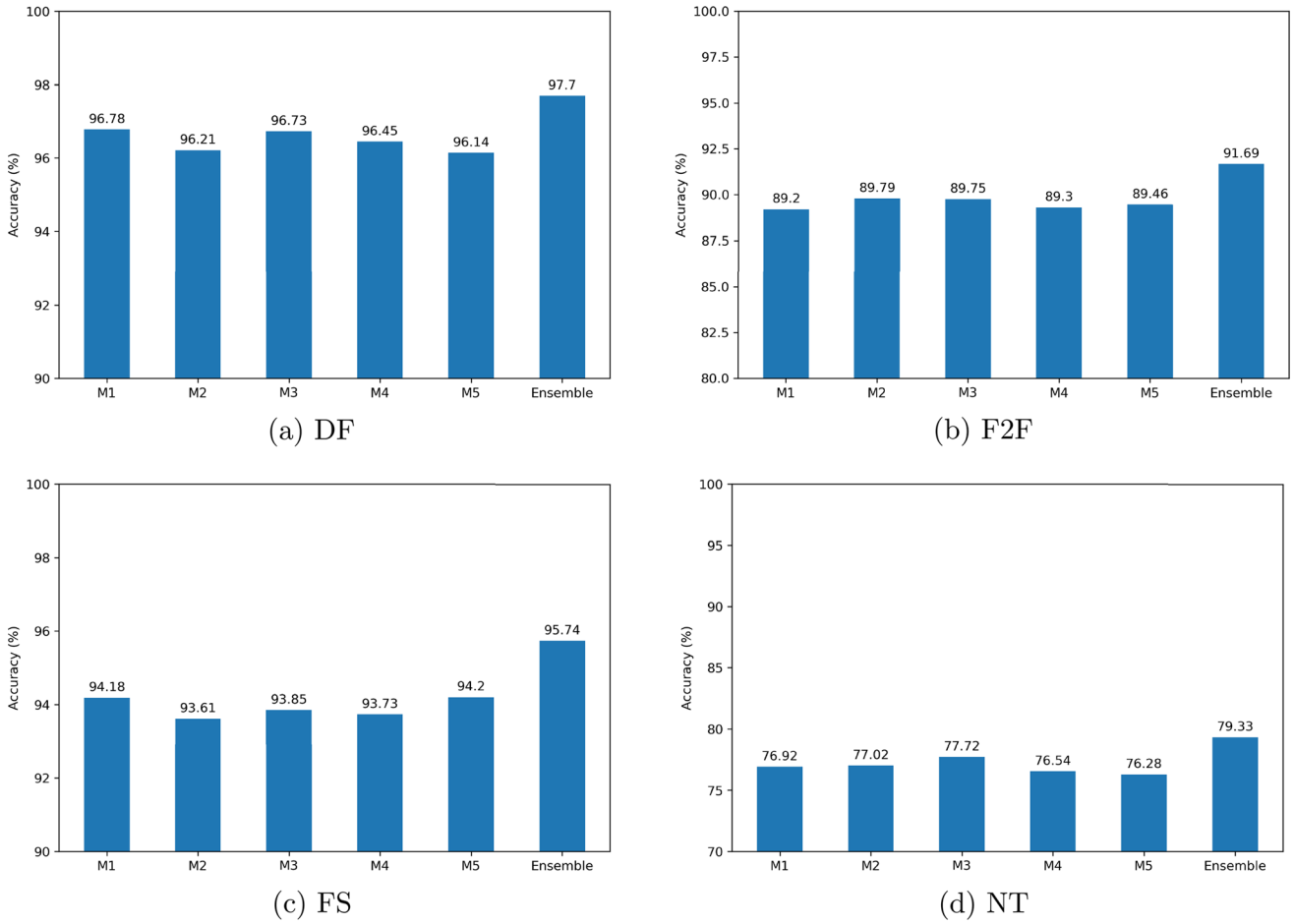

Fig. 8 The accuracy of each model in the bagging ensemble on the FaceForensics++ dataset

Table 4 A comparison of the proposed ensemble method with the state-of-the-arts on different manipulation techniques (DF, F2F, FS, NT) of the FaceForensics++ dataset

Model	DF		F2F		FS		NT	
	ACC	AUC	ACC	AUC	ACC	AUC	ACC	AUC
MesoNet [19]	87.27	–	56.20	–	61.17	–	40.67	–
Rahmouni et al. [20]	85.45	–	64.23	–	56.31	–	60.07	–
Steg. Features [18]	73.64	–	73.72	–	68.93	–	63.33	–
Dong et al. [24]	86.30	94.10	68.30	81.40	58.20	65.60	67.80	79.20
Capsule [22]	92.17	–	90.36	–	92.79	–	–	–
Wodajo et al. [34]	93.00	–	69.39	–	69.00	–	60.00	–
Hu et al. [28]	94.64	98.00	86.48	94.00	85.27	94.00	**80.05**	**90.00**
Liu et al. [23]	93.48	98.50	86.02	94.62	92.26	98.10	76.78	80.49
Ensemble (Ours)	**97.70**	**99.70**	**91.69**	**97.49**	**95.74**	**98.90**	79.32	87.62

Bold values indicate the best results for each evaluation metric on each model and higher value means better result

The MesoNet method [19] achieves the lowest ACC scores for F2F, and NT manipulation techniques but with relatively high accuracy for DF. The Rahmouni et al. method [20] performs the lowest ACC score for FS. The Steg. Features method [18] achieves better ACC scores than both MesoNet and Rahmouni et al. methods for F2F, FS, and NT manipulation techniques, but fails behind both for detecting DF manipulation. Dong et al. method [24] only shows some improvement in detecting NT compared to Steg. Features, MesoNet, and Rahmouni et al. methods. The Capsule networks [22] have relatively high accuracy for DF and F2F but have no results for FS and NT. The

Table 5 Results on Celeb-DF dataset using XceptionNet and CoAtNet models

Model	Celeb-df	
	ACC	AUC
XceptionNet	95.35	98.97
CoAtNet	96.79	99.45
CoAtNet + aug	96.90	99.39
CoAtNet + CM	**97.10**	**99.49**

Bold values indicate the best results for each evaluation metric on each model and higher value means better result

Table 6 Results of the ensemble of CoAtNet models on the Celeb-DF dataset using different ensemble techniques

Model	Celeb-df	
	ACC	AUC
Majority Voting	97.61	–
Sum Rule	97.71	99.70
Product Rule	97.81	99.74

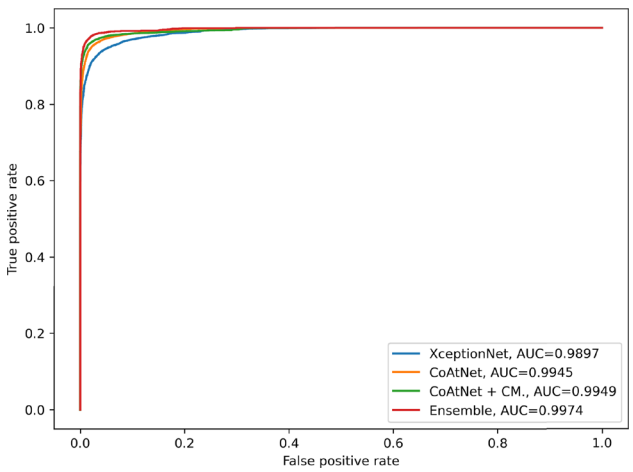

Fig. 9 Comparison of ROC curves and AUC values of the proposed ensemble method and baseline methods on the Celeb-DF dataset

Wodajo et al. method [34], which is a ViT-based model, has high accuracy for DF compared to the Capsule networks but performs poorly for FS, F2F, and NT. The Hu et al. and Liu et al. methods [23, 28] achieve high accuracy and AUC scores for all manipulation techniques compared to the previously mentioned methods. The proposed ensemble method outperforms all previous methods in terms of accuracy and AUC for all manipulation techniques, except for NT, the proposed method achieves ACC of 79.32% and AUC of 87.62% compared to ACC of 80.05% and AUC of 90.00% for Hu et al [28] method. NT manipulation is considered challenging to detect because it involves synthesizing new textures to replace the original ones, resulting in a more realistic and difficult-to-detect forgery.

Table 7 A comparison of the proposed ensemble method with the state-of-the-art methods on the Celeb-DF dataset

Model	Celeb-df	
	ACC	AUC
Yang et al. [16]	–	54.60
MesoNet [19]	–	54.80
Capsule [22]	–	57.50
Li et al. [57]	–	64.60
Dang et al. [25]	–	71.20
Hu et al. [28]	80.74	87.00
Chen et al. [31]	–	90.56
Ciftci et al. [21]	91.50	–
Silva et al. [29]	93.64	98.41
Heo et al. [33]	–	99.30
Ensemble (Ours)	**97.81**	**99.74**

Bold values indicate the best results for each evaluation metric on each model and higher value means better result

Overall, the proposed method outperforms other methods for several reasons. Firstly, it employs a deep learning bagging ensemble classifier which reduces overfitting and improves generalization performance, particularly for complex and heterogeneous datasets like the FaceForensics++ dataset. Moreover, the proposed ensemble method utilizes the CoAtNet model as a base learner, enabling it to efficiently capture long-range dependencies of spatial details and improve the model's capacity and efficiency. Additionally, the proposed ensemble utilizes CutMix data augmentation on the extracted faces, which enhances the performance of the ensemble model. However, some of the other methods only give a good performance for specific manipulation techniques, indicating that they may not be generalized enough to detect face manipulations in various videos.

4.5 Results on celeb-DF dataset

We evaluate XceptionNet and CoAtNet models on the Celeb-DF dataset and compare their performance for DeepFake detection. Table 5 shows that there is a significant increase in the accuracy of CoAtNet model by almost 1.5% compared to XceptionNet. Additionally, the table shows the results of using data augmentation on the CoAtNet model further improve its performance. The results show that the CutMix augmentation technique leads to the best performance, with an AUC of 99.49% and an

accuracy of 97.10%. This is compared to the CoAtNet model with traditional data augmentation which has an AUC of 99.39% and an accuracy of 96.90%. These results indicate that the CutMix augmentation is more effective in enhancing the performance of the CoAtNet model on the DeepFake detection task.

Finally, we evaluate three different ensemble techniques of CoAtNet with CutMix augmentation using Majority Voting, Sum, and Product. The results on the Celeb-DF dataset show that all three ensemble methods improve the performance of the single CoAtNet model. When comparing the performance of the three methods, we find that the Product rule slightly increases the accuracy of DeepFake detection, as shown in Table 6. This indicates that the Product rule is more effective in combining the predictions of multiple models for DeepFake detection on the Celeb-DF dataset.

We also compare the ROC curve for the proposed ensemble method using the Product rule on the Celeb-DF dataset to those of the XceptionNet, CoAtNet, and CoAtNet + CM. models as shown in the Fig. 9. We can see that the ROC curve for our ensemble method (red curve) is better than the XceptionNet model (blue curve) with a higher AUC value of 99.74% compared to the XceptionNet's AUC value of 98.97%. This indicates that our proposed method generalizes well on the Celeb-DF dataset and outperforms the baseline method on both datasets.

To assess the performance of our proposed method, we conduct a comparative analysis with some of the state-of-the-art techniques using the Celeb-DF dataset as shown in Table 7. The proposed method achieves the highest accuracy and AUC scores with 97.81% and 99.74% compared to the Silva et al. [29] with 93.64% and 98.41% which comes in the second place. The results show that the proposed ensemble method is an effective method for detecting DeepFake manipulations.

5 Conclusion

In this paper, we presented a new method for detecting manipulated faces using the bagging ensemble technique. Our method uses a CoAtNet model as the base learner for the ensemble method. Our method was tested on the Celeb-DF and FaceForensics++ datasets and showed superior results compared to existing state-of-the-art techniques. We improved the performance of the CoAtNet model by incorporating the CutMix augmentation technique. The proposed ensemble of CoAtNet models with the CutMix augmentation technique achieved the best results in detecting manipulated faces on both datasets. In future work, we will evaluate the proposed method on various other DeepFake datasets and incorporate alternative augmentation techniques to enhance its performance.

Funding Open access funding provided by The Science, Technology & Innovation Funding Authority (STDF) in cooperation with The Egyptian Knowledge Bank (EKB).

Data availability The used datasets are public and citations are included.

Declarations

Conflict of interest The authors declare no conflict of interest.

References

1. Bitesize B (2019) deepfakes: What are they and why would i make one? 2019.[Online]
2. Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y (2020) Generative adversarial networks. Commun ACM 63(11):139–144
3. Kingma DP, Welling M (2013) Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114
4. FaceApp: Perfect Face Editor. [Online; accessed 21-December-2022]. https://apps.apple.com/gb/app/faceapp-ai-face-editor/id1180884341
5. FaceSwap github. [Online; accessed 05-December-2022]. https://github.com/MarekKowalski/FaceSwap/
6. Karras T, Laine S, Aila T (2019) A style-based generator architecture for generative adversarial networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 4401–4410
7. Deepfakes github. [Online; accessed 05-December-2022]. https://github.com/deepfakes/faceswap
8. ZAO App. [Online; accessed 05-December-2022]. https://apps.apple.com/cn/app/id1465199127
9. facebook. [Online; accessed 21-December-2022]. https://www.bbc.com/news/technology-48607673
10. 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
11. Thies J, Zollhöfer M, Nießner M (2019) Deferred neural rendering: image synthesis using neural textures. ACM Trans Graph (TOG) 38(4):1–12
12. Choi Y, Choi M, Kim M, Ha J-W, Kim S, Choo J (2018) Stargan: unified generative adversarial networks for multi-domain image-

to-image translation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 8789–8797
13. Yerushalmy I, Hel-Or H (2011) Digital image forgery detection based on lens and sensor aberration. Int J Comput Vision 92(1):71–91
14. 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
15. Agarwal S, Farid H, Gu Y, He M, Nagano K, Li H (2019) Protecting world leaders against deep fakes. In: CVPR Workshops, vol 1, p 38
16. Yang X, Li Y, Lyu S (2019) Exposing deep fakes using inconsistent head poses. In: ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp 8261–8265. IEEE
17. 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
18. Fridrich J, Kodovsky J (2012) Rich models for steganalysis of digital images. IEEE Trans Inf Forensics Secur 7(3):868–882
19. 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), pp 1–7.
20. Rahmouni N, Nozick V, Yamagishi J, Echizen I (2017) Distinguishing computer graphics from natural images using convolution neural networks. In: 2017 IEEE Workshop on Information Forensics and Security (WIFS), pp 1–6.
21. Ciftci UA, Demir I, Yin L (2020) Fakecatcher: detection of synthetic portrait videos using biological signals. In: IEEE transactions on pattern analysis and machine intelligence
22. Nguyen H, Yamagishi J, Echizen I (2019) Use of a capsule network to detect fake images and videos. arxiv 2019. arXiv preprint arXiv:1910.12467
23. Liu H, Li X, Zhou W, Chen Y, He Y, Xue H, Zhang W, Yu N (2021) Spatial-phase shallow learning: rethinking face forgery detection in frequency domain. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 772–781
24. Dong F, Zou X, Wang J, Liu X (2023) Contrastive learning-based general deepfake detection with multi-scale RGB frequency clues. J King Saud Univ-Comput Inf Sci 35(4):90–99
25. Dang H, Liu F, Stehouwer J, Liu X, Jain AK (2020) On the detection of digital face manipulation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 5781–5790
26. Khalid F, Javed A, Ilyas H, Irtaza A et al (2023) DFGNN: An interpretable and generalized graph neural network for deepfakes detection. Expert Syst Appl 222:119843
27. de Lima O, Franklin S, Basu S, Karwoski B, George A (2020) Deepfake detection using spatiotemporal convolutional networks. arXiv preprint arXiv:2006.14749
28. Hu J, Liao X, Wang W, Qin Z (2021) Detecting compressed deepfake videos in social networks using frame-temporality two-stream convolutional network. IEEE Trans Circuits Syst Video Technol 32(3):1089–1102
29. Silva SH, Bethany M, Votto AM, Scarff IH, Beebe N, Najafirad P (2022) Deepfake forensics analysis: an explainable hierarchical ensemble of weakly supervised models. Forensic Sci Int: Synergy 4:100217
30. Rana MS, Sung AH (2020) Deepfakestack: a deep ensemble-based learning technique for deepfake detection. In: 2020 7th IEEE International Conference on Cyber Security and Cloud Computing (CSCloud)/2020 6th IEEE International Conference on Edge Computing and Scalable Cloud (EdgeCom), pp 70–75.
31. Chen H-S, Rouhsedaghat M, Ghani H, Hu S, You S, Kuo C-CJ (2021) Defakehop: a light-weight high-performance deepfake detector. In: 2021 IEEE International Conference on Multimedia and Expo (ICME), pp 1–6
32. Heo Y-J, Choi Y-J, Lee Y-W, Kim B-G (2021) Deepfake detection scheme based on vision transformer and distillation. arXiv preprint arXiv:2104.01353
33. Heo Y-J, Yeo W-H, Kim B-G (2023) Deepfake detection algorithm based on improved vision transformer. Appl Intell 53(7):7512–7527
34. Wodajo D, Atnafu S (2021) Deepfake video detection using convolutional vision transformer. arXiv preprint arXiv:2102.11126
35. Wang X, Girshick R, Gupta A, He K (2018) Non-local neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 7794–7803
36. Bello I, Zoph B, Vaswani A, Shlens J, Le QV (2019) Attention augmented convolutional networks. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp 3286–3295
37. Srinivas A, Lin T-Y, Parmar N, Shlens J, Abbeel P, Vaswani A (2021) Bottleneck transformers for visual recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 16519–16529
38. Shen Z, Zhang M, Zhao H, Yi S, Li H (2021) Efficient attention: attention with linear complexities. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp 3531–3539
39. Dosovitskiy A, Beyer L, Kolesnikov A, Weissenborn D, Zhai X, Unterthiner T, Dehghani M, Minderer M, Heigold G, Gelly S et al. (2020) An image is worth 16x16 words: transformers for image recognition at scale. arXiv preprint arXiv:2010.11929
40. Tan M, Le Q (2019) Efficientnet: Rethinking model scaling for convolutional neural networks. In: International Conference on Machine Learning, pp 6105–6114. PMLR
41. Vaswani A, Ramachandran P, Srinivas A, Parmar N, Hechtman B, Shlens J (2021) Scaling local self-attention for parameter efficient visual backbones. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 12894–12904
42. Liu Z, Lin Y, Cao Y, Hu H, Wei Y, Zhang Z, Lin S, Guo B (2021) Swin transformer: hierarchical vision transformer using shifted windows. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp 10012–10022
43. Dai Z, Liu H, Le QV, Tan M (2021) Coatnet: marrying convolution and attention for all data sizes. Adv Neural Inf Process Syst 34:3965–3977
44. Sandler M, Howard A, Zhu M, Zhmoginov A, Chen L-C (2018) Mobilenetv2: inverted residuals and linear bottlenecks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 4510–4520
45. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, Polosukhin I (2017) Attention is all you need. In: Advances in neural information processing systems vol 30
46. Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556
47. 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
48. Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A (2015) Going deeper with convolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 1–9

49. Shaw P, Uszkoreit J, Vaswani A (2018) Self-attention with relative position representations. arXiv preprint arXiv:1803.02155
50. Huang C-ZA, Vaswani A, Uszkoreit J, Shazeer N, Simon I, Hawthorne C, Dai AM, Hoffman MD, Dinculescu M, Eck D (2018) Music transformer. arXiv preprint arXiv:1809.04281x
51. Ramachandran P, Parmar N, Vaswani A, Bello I, Levskaya A, Shlens J (2019) Stand-alone self-attention in vision models. In: Advances in Neural Information Processing Systems vol. 32
52. Yun S, Han D, Oh SJ, Chun S, Choe J, Yoo Y (2019) Cutmix: regularization strategy to train strong classifiers with localizable features. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp 6023–6032
53. DeVries T, Taylor GW (2017) Improved regularization of convolutional neural networks with cutout. arXiv preprint arXiv:1708.04552
54. Zhong Z, Zheng L, Kang G, Li S, Yang Y (2020) Random erasing data augmentation. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol 34, pp 13001–13008
55. Zhang H, Cisse M, Dauphin YN, Lopez-Paz D (2017) mixup: beyond empirical risk minimization. arXiv preprint arXiv:1710.09412
56. Zhou Z-H (2021) Ensemble learning. In: Machine Learning, pp 181–210. Springer
57. Li Y, Yang X, Sun P, Qi H, Lyu S (2020) Celeb-df: a large-scale challenging dataset for deepfake forensics. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 3207–3216
58. Reinhard E, Adhikhmin M, Gooch B, Shirley P (2001) Color transfer between images. IEEE Comput Graph Appl 21(5):34–41
59. Ma L, Jia X, Sun Q, Schiele B, Tuytelaars T, Van Gool L (2017) Pose guided person image generation. In: Advances in neural information processing systems vol 30
60. Zhang K, Zhang Z, Li Z, Qiao Y (2016) Joint face detection and alignment using multitask cascaded convolutional networks. IEEE Signal Process Lett 23(10):1499–1503
61. Chollet F (2017) Xception: deep learning with depthwise separable convolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 1251–1258