IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 17 November 2022, accepted 17 December 2022, date of publication 26 December 2022,
date of current version 3 January 2023.

Digital Object Identifier 10.1109/ACCESS.2022.3232290

RESEARCH ARTICLE

Generalization of Forgery Detection With Meta Deepfake Detection Model

VAN-NHAN TRAN[1], SEONG-GEUN KWON[2], SUK-HWAN LEE[3], HOANH-SU LE[4], AND KI-RYONG KWON[1]

[1]Department of Artificial Intelligence Convergence, Pukyong National University, Busan 48513, South Korea
[2]Department of Electronics Engineering, Kyungil University, Gyeongsan 38428, South Korea
[3]Department of Computer Engineering, Dong-A University, Busan 49315, South Korea
[4]Faculty of Information Systems, University of Economics and Law, Vietnam National University Ho Chi Minh City, Ho Chi Minh 700000, Vietnam

Corresponding author: Ki-Ryong Kwon (krkwon@pknu.ac.kr)

This work was supported in part by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education under Grant 2020R1I1A306659411 and Grant 2020R1F1A1069124; and in part by the Ministry of Science and ICT (MSIT), South Korea, through the Information Technology Research Center (ITRC) Support Program supervised by the Institute for Information & Communications Technology Planning & Evaluation (IITP) under Grant IITP-2022-2020-0-01797.

ABSTRACT Face forgery generating algorithms that produce a range of manipulated videos/images have developed quickly. Consequently, this causes an increase in the production of fake information, making it difficult to identify. Because facial manipulation technologies raise severe concerns, face forgery detection is gaining increasing attention in the area of computer vision. In real-world applications, face forgery detection systems frequently encounter and perform poorly in unseen domains, due to poor generalization. In this paper, we propose a deepfake detection method based on meta-learning called Meta Deepfake Detection (MDD). The goal of the model is to develop a generalized model capable of directly solving new unseen domains without the need for model updates. The MDD algorithm establishes various weights for facial images from various domains. Specifically, MDD uses meta-weight learning to shift information from the source domains to the target domains with meta-optimization steps, which aims for the model to generate effective representations of the source and target domains. We build multi-domain sets using meta splitting strategy to create a meta-train set and meta-test set. Based on these sets, the model determines the gradient descent and obtains backpropagation. The inner and outer loop gradients were aggregated to update the model to enhance generalization. By introducing pair-attention loss and average-center alignment loss, the detection capabilities of the system were substantially enhanced. In addition, we used some evaluation benchmarks established from several popular deepfake datasets to compare the generalization of our proposal in several baselines and assess its effectiveness.

INDEX TERMS Deepfake detection, meta-learning, artificial intelligence, computer vision.

I. INTRODUCTION

Face recognition systems have progressed substantially in recent times. In particular, deep learning technologies have significantly improved the performance of this task. However, the sophistication of face image manipulation puts existing facial recognition algorithms in danger of being considered inefficient. With the development of technologies such as Generative Adversarial Networks (GAN) [1], GANs family, and Variational AutoEncoders [2], [3]. Fake facial images and videos can be made and utilized to deceive recognition systems. Many manipulation algorithms [4], [5], [6] person without specific skills to produce high-quality fake faces without expert skills and special knowledge for training. As a result, it can be often challenging for the human eyes to identify the difference between actual and manipulated images. This has led to an increase in the usage of modified multimedia content in various cybercrime activities. The technology may be utilized maliciously, resulting in a major trust issue for modern society. Due to the fact that such methods may produce

The associate editor coordinating the review of this manuscript and approving it for publication was Liangxiu Han.

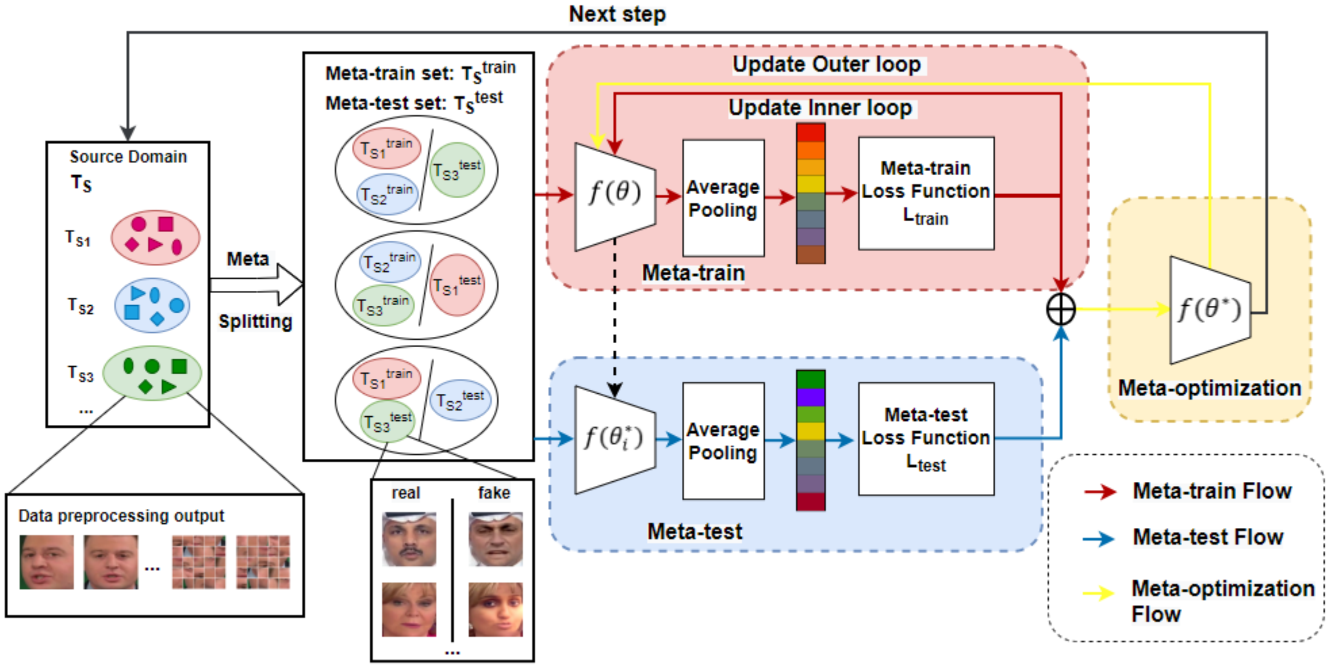

FIGURE 1. **Overview architecture of our proposed MDD.**

high-quality fake images that are even indistinguishable from human eyes. Therefore, the scientific community has shown a lot of interest in the need to develop techniques for identifying authentic faces from fraudulent images. Many methods for deepfake detection have been proposed in [7], [8], [9], [10], and [11]. These proposals primarily take inspiration from the binary classification problem, applying its models to the deepfake detection challenge in order to differentiate between real and fake photos. The common model for these proposals typically uses the data preprocessing associated with backbone networks to extract features from faces in images or videos. Then uses a binary classifier network to classify them into real and fake ones. However, due to the rapid advancement of face forgery generation algorithms, some samples seem extremely similar to one another and only differ from one another by a few small features, it is getting harder to determine the difference between fake and real features in fake images. In addition, there is a lot of variety in fake images which are produced using different algorithms. Resulting in the ineffective performance of such global feature-based systems which used binary classifier networks.

Presently, Face forgery generation algorithms are increasing rapidly, which can be mentioned as expression swapping, identity swapping, face swapping, face synthesis, etc. Based on these algorithms, a variety of manipulated datasets is created to serve the research and development of face forgery detection. Several common datasets used in the experiment of this paper are DFDC [12], Celeb-DF-v2 [13], FaceForensics++ [9]. The synthetic faces in these datasets were produced using the same algorithm leading to similar data distribution in each one. When training and testing are completed on one dataset, then only one data distribution set is used to assess the outcomes. When testing with other databases, often the results are poor. However, in real-world applications, the model is frequently used in a significantly different domain (unseen domain) with a different distribution than the source domains. As a result, generalized face forgery detection is less researched and more difficult with unseen facial manipulations.

In this research, we design a generalized face forgery detection model to solve the face authentication issue. Without any model updating, the model can be evaluated directly on unseen domains after being trained on a number of source domains. Inspired by [14], [15], and [16], by using meta-learning, we propose a novel deepfake detection algorithm, termed Meta Deepfake Detection (MDD). With a meta-optimization objective, in order to learn efficient face representations on both synthetic source and target domains. The MDD shifts the source domain to the target domain. So as to increase model generalization, the gradients from the meta-train and the meta-test are combined using meta-optimization. The MDD can handle unseen domains without model updating for unseen domains. The followings are summary of our main contributions:

- We propose a Meta Deepfake Detection model (MDD) to handle the generalization of the deepfake forgery detection problem, which uses transferable knowledge across domains to learn from meta-learning to enhance model generalization.
- We emphasize the generalized deepfake detection challenge, which necessitates that a trained model

generalizes effectively on new domains without any updating.
- We propose two loss functions: Pair-Attention Loss (PAL), which is to concentrate on maximizing positive and negative pairings and separating positive samples from negative samples. Average-Center Alignment Loss (ACA), which is to minimize the variations in each class, while retaining the capacity to differentiate between features of various classes. Moreover, these two losses are aggregated with softmax loss to update the entire model and learn across domains.
- We apply data preprocessing along with the block shuffling transformation technique to increase the performance of the generalized model.
- Some generalized deepfake detection benchmarks are used for the evaluation of our proposal. A number of experiments on these evaluation benchmarks are conducted and compared with some related methods.

II. RELATED WORK

A. FACE FORGERY GENERATION

Deep generative models, which are gaining popularity, are being used to synthesize and produce fake videos and images. The manipulation algorithms also expand along with it. Several well-known algorithms include face swap, face manipulation, expression reenactment, etc.

1) FACE SWAP

Face swapping involves replacing the face of a source image with that of a target image. Some remarkable research such as RSGAN [17] proposed a region-separative generative adversarial network, which replaces the handles face and hair appearances in the latent-space representations of the faces and reconstructs the full face to achieve face swapping. FSGAN [18] proposed Face Swapping GAN, which derives a recurrent neural network (RNN) for face reenactment and adapts to changes in position and expression. FSGANv2 [19] offered a subject-agnostic swapping scheme for face reenactment which adjusts important pose and expression variation. MobileFaceSwap [20] proposed an advanced face swapping approach with a lightweight Identity-aware Dynamic Network (IDN) to modify the model parameters depending on the identification information dynamically.

2) FACE MANIPULATION

It is a generation task in which the facial attributes and styles of the output face are changed to point in the direction of the intended target. AttGAN [21] applied an attribute classification constraint to ensure the precise changing of the desired characteristics in the resulting image and preserve attribute-excluding details. Moreover, the suggested approach is enhanced to allow attribute style adjustment in an unsupervised setting. STGAN [22] presented a selective transfer perspective to utilize the target attribute vector to direct the flexible translation to the desired target domain. MaskGAN [6] proposed a model with two primary components: Dense Mapping Network (DMN) and Editing Behavior Simulated Training (EBST) to modify target images and learn style mapping by using a modified mask. StarGANv2 [23] proposed a framework that meets the variety of generated images and scalability across multiple domains when learning a mapping across several visual domains. FacialGAN [24] proposed a framework that allows for the simultaneous manipulation of dynamic face features and extensive style transfers.

3) EXPRESSION REENACTMENT

The conditional face synthesis problem of facial expression reenactment aims to transfer a source face shape to a target face while keeping the same target identity of the face and appearance. Some related research can be mentioned as MarioNETte [25] which creates professional reenactments of hidden identities in a few-shot environment to handle attention block of the image, facial landmark transformer, and focus feature alignment. DEA-GAN [26] presented a self-supervised hybrid model that learns an embedded face that is pose-invariant for each video by using a multi-frame deforming auto-encoder. FReeNet [27] proposed a multi-identity face reenactment framework to share a common model and transmit facial expressions from the source face to the target face. AD-NeRF [28] proposed an audio-driven talking head technique that renders portraits by directly mapping audio characteristics to dynamic neural radiance fields. FACEGAN [29] proposed a model that uses the Action Unit (AU) representation to transfer from the driving face to facial motion.

B. FACE FORGERY DETECTION

Face forgery detection is divided into different groups, such as spatial clue for detection, temporal clue for detection, and generalizable clue for detection.

1) SPATIAL CLUE FOR DETECTION

The work in [30] presented an innovative attention-based layer to boost classification efficiency and generate an attention map showing the altered face areas. Furthermore, the work in [31] designed an inconsistency-aware wavelet dual-branch network to recognize real and fake images. Capsule-forensics [32] proposed a method that employs a deep convolutional neural network and a capsule network to identify several types of spoofs, including replay attacks that use printed pictures or recorded movies and computer-generated videos. FakeLocator [33] introduced the attention mechanism by using face parsing and suggest a single sample clustering and partial data augmentation to improve the training data. In research [34], with the goal of developing a novel detection technique that can find a forensics trail concealed in images, we concentrate on the analysis of deep fakes of human faces.

2) TEMPORAL CLUE FOR DETECTION

MesoNet [35] presented a method for quickly and effectively identifying face tampering in videos with a focus on two recent methods for creating fake videos that appear to be extremely realistic. FakeCatcher [36] provided a fresh method for detecting phony content in portrait videos as a preventative measure against the growing danger of deepfake. Bita-Net [37] proposed a model to detect fake faces, which reflects the two-pathway architecture to enhance the forgery detection ability. Furthermore, the work in [38] proposed a spatiotemporal attention mechanism combined with Xception-LSTM algorithm to improve deepfake detection.

3) GENERALIZABLE CLUE FOR DETECTION

The work in [39] presented a multi-task incremental learning-based methodology for the detection and classification of manipulated images, the model can adapt to new classes without losing the existing information. OC-FakeDect [40] presented a model that only uses actual face images for training, and treats fake images like deepfakes as irregularities. The research in [41] recommended intensive training to increase generalization performance. The generalization ability significantly enhances by adversarially created training samples that are designed to challenge the classification models.

C. META-LEARNING

One of the most promising and popular research areas in the field of artificial intelligence currently is meta-learning. Basically, With the help of meta-learning, an adaptable AI model is created that can learn to perform various tasks without needing to be trained from scratch. The meta-learning model is trained on a variety of related tasks on sparse data points, allowing it to apply to learn from such tasks to new related tasks. Some famous research can be mentioned as MAML [42] which is to find a better initial parameter. So that the model can learn quickly on new tasks with fewer gradient steps. CAML [43] used context parameters and shared parameters to adapt and share information across tasks in order to avoid overfitting problems. Meta-SGD [16], a meta-learning algorithm that is used for performing learning quickly, Meta-SGD not only determines the optimal parameter but also the optimal learning rate and update direction. TAML [44], which prevents the problem that the model can be biased over some tasks during adapting to new tasks with meta-learning technique, especially the tasks that are sampled in the meta-training phase. MLDG [14] presented a new method of meta-learning for domain generalization and a training procedure for domain generalization by developing models that naturally generalize to new testing domains.

Some researches used the meta-learning to solve the problem of face forgery detection and face anti-spoofing can be mentioned in the work [45], they designed a novel meta-learning framework named Regularized Fine-grained Meta-learning to identify generalized learning directions in the meta-learning process, which is accomplished by performing effectively in the simulated domain shift scenarios. The work in [46] designed a domain generalization model, named learning-to-weight. The facial pictures from various domains are configured with various weights. The generalizability of the model can be balanced across many domains using their network. The gradient of the source domain is then calibrated by the meta-optimization, allowing for the learning of additional discriminative features. The work in [47] presented a frequency adversarial attack technique based on meta-learning for face forgery detection. Moreover, they performed a discrete cosine transform (DCT) on the input photos and applied a fusion module to capture the strong area in the frequency domain. NAS-FAS [48], presented an approach based on neural architecture search and created a brand-new search space using pooling and central difference convolution operators. The work in [49] suggested a learnable network to extract Meta Pattern (MP) in their architecture for learning to learn and created a two-stream network utilizing their suggested Hierarchical Fusion Module to hierarchically fuse the input RGB picture and the extracted MP. The discriminative features extracted from MP are capable of learning a more generalized model by substituting handmade features with the MP.

III. METHODOLOGY

A. OVERVIEW

We suggest a method based on meta-learning called the meta deepfake detection (MDD) algorithm. The model aims to enhance the performance of detecting manipulated images and videos produced by a certain method as well as enhance the generalization of the detector. In the training stage, we have N-related tasks: $T_S = \{T_{S1}, T_{S2}, \ldots, T_{SN}\}; N > 1$ and each task $T_{Si} = \{(x_i, y_i)\}$, where T_{Si} represents the i^{th} task, x_i is extracted feature vectors and y_i is its own set of labels. In the evaluating stage, the trained model is tested on one or more unseen target domains, $\geq T_T = \{T_{T1}, T_{T2}, \ldots, T_{TM}\}; M1$. The model learns from a variety of connected tasks, and the meta-learner process makes it fast learner with good generalization abilities. We define a single backbone during training, a parametrized function $f(\theta)$ with parameters θ. It will generalize parameters to predict accurately the target domain by training and optimizing for source domains. The overall architecture is displayed in Fig. 1.

B. META SPLITTING

We separated the source domains into the meta-train domain T_s^{train} and the meta-test domain T_s^{test} during training to obtain domain generalization. In order to simulate the domain shift problem that existed when used in real-world situations, the model is driven to acquire generalizable information about how to generalize well on the new domains with different distributions. We also create meta-batches for training and testing by randomly splitting N source domains of T_S; these data contain both real and fake face pairs and these patterns are not duplicated across domains. These pairs increase collation and comparison of information between real and fake images. Therefore, it also increases inter-class separability, which can be interpreted as a distinct dispersion of the feature distribution of samples, increasing differentiation during training as well as enhancing the model's quality. More distinguishable characteristics may be learned by the network with less effort during optimization.

The fact is that features learned by supervised learning have much less ability to generalize when subjected to unseen manipulation techniques. This suggests that supervised learning-based characteristics have a close relationship with manipulation techniques. The features of samples produced by various manipulation techniques make it challenging to combine all of the manipulated faces. Therefore, the model is easier to generalize when the source domain is split into meta-train and meta-test. In addition, samples in the meta-train and meta-test are also shuffled and selected at random, which minimizes the problem of overfitting. Additionally, the data in the unseen domain is very diverse in reality, which the model has never seen or been trained in before. Thus, meta-splitting makes the model easier to train and also to generalize to unseen data.

C. DATA PREPROCESSING

A lot of data is used to train deep learning models. Hence, proper dataset preparation is essential for their learning quality and prediction accuracy. In our paper, we use several existing datasets, including DFDC [12], Celeb-DF-v2 [13], FaceForensics++ [9]. These datasets include real and manipulated videos, accompanied by real or fake labels. These videos are sampled to obtain images. Afterward, face extraction is utilized for extracting the faces from the images and resizing them to 224 × 224 RGB format. Multitask cascaded convolutional network (MTCNN) [50] library is used to extract the faces. Fig. 2 and Fig. 3 show a sample of the extracted faces. Our approach does not use any data augmentation techniques in order to compare fairly with the study contents relevant to deepfake detection. After obtaining a set of extracted face images, *block pixel shuffling transformation* is applied on the part of extracted face images to increase the diversity of the data set during training. It is different from data augmentation as the amount of data after applying block shuffling transformation does not change. The overview of the data preprocessing process is shown in Fig. 4.

The local spatial structure of the local regions might be destroyed by the shuffling of the pixels in an image, which prevents the network from extracting valuable features. This is also mentioned in some research related to image encryption [51], [52], [53], [54]. However, if the blocks of the image are shuffled in a proper way, it can lead to preserving essential characteristics while also enhancing the quality of the model [55], [56]. Additionally, several researches in [57] and [58] have demonstrated that creating patches by using characteristics gathered in an image also increases the quality of the training process. Therefore, these demonstrate that block shifting and shuffling local regions greatly raise quality when applied properly. The block shuffling transformation is a data enhancement technique to increase the performance of the system. So as to improve the robustness and generalization of face forgery detection, more efficient local features are extracted using neural networks. A portion of the sample of the meta-train set is applied block shuffling transformation. The description for block shuffling transformation is shown in Fig. 5.

FIGURE 2. A few samples of extracted real images in FaceForensics++ dataset [9].

FIGURE 3. A few samples of extracted fake images in FaceForensics++ dataset [9].

We divide an image with RGB format has dimension of $X \times Y \times 3$ into block by using a window (with the window size of $W \times W \times 3$). The original image will be divided into

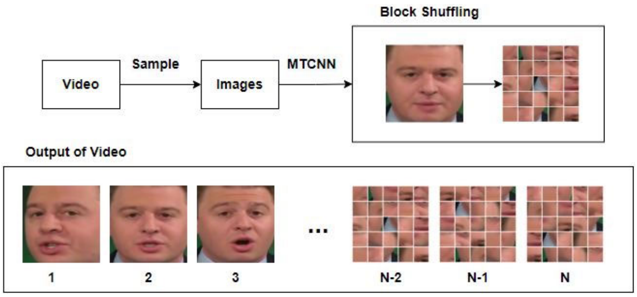

FIGURE 4. Overview of data processing.

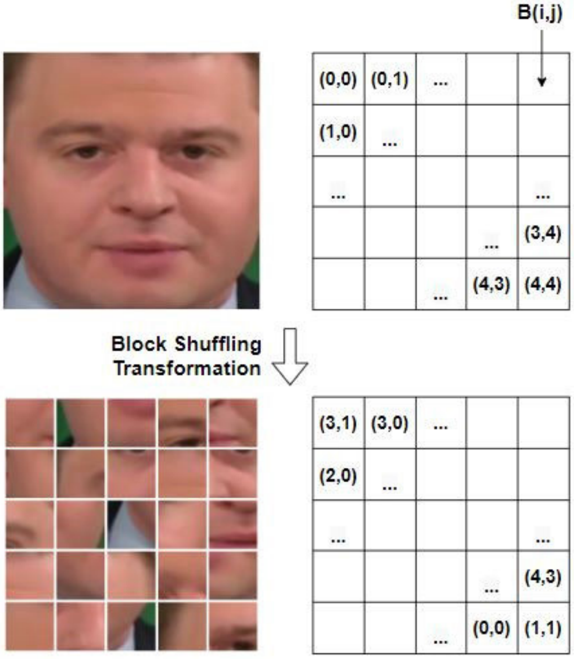

FIGURE 5. Visualization of block shuffling transformation. On top is an example of the original image and bottom is a shuffled image and coordinate blocks.

smaller blocks with the size of the window. If the original size of the image is not divisible by the block size, padding will be applied. Thus, we get $r \times c$ blocks and i and j are the horizontal and vertical indices, respectively, where r and c are the horizontal and vertical blocks, $i \in 0, 1, \ldots, r$ and $j \in 0, 1, \ldots, c$. The block $B(i, j)$ interpreted as block i^{th} row and j^{th} column. The block $B(i, j)$ is changed randomly. Where i^{th} is an integer from 0 to r that is randomly permuted, and j^{th} is a random permutation of integer from 0 to c.

D. LOSS FUNCTION

1) PAIR-ATTENTION LOSS (PAL)

The basic idea of Pair-Attention Loss (PAL) is to focus on optimizing negative and positive pairs, along with distinguishing between positive samples and negative samples. A batch of each iteration contains B identities, each identity contains real and fake faces. We define the input as X. With B identities, we have $F_r = f(X_r, \theta), \in R^{P\times C}$, $F_f = f(X_f, \theta), \in R^{N\times C}$, where C is the dimension length, F_r is the embedding vector of real face obtained through model $f(\theta)$, F_f is the embedding vector of fake face obtained through model $f(\theta)$, with label $l = \{l_1, l_2, \ldots, l_C\}$, $l_i \in (0, 1)$, where "0" means a real sample and "1" mean a fake sample. P is the number of positive samples in a batch B and N is the number of negative samples in a batch B, $B = P + N$. The PAL function can be formulated as follows.

$$\begin{aligned} L_{PAL} = & \frac{1}{2\,(P+1)} \sum_{i\in P} |\boldsymbol{F_{ri}} - \boldsymbol{F_{fi}}|_2 \\ & - \frac{1}{2\,(N+1)} \sum_{j\in N} |\boldsymbol{F_{fj}} - \boldsymbol{F_{rj}}|_2 \end{aligned} \tag{1}$$

2) SOFTMAX LOSS (SOF)

The goal of softmax loss is to identify a decision boundary that divides several classes by mapping the samples to discrete labels. The softmax loss function is presented as follows.

$$L_{SOF} = \sum_{i=1}^{m} Log \frac{e^{\boldsymbol{w_{y_i}^T x_i + b_{y_i}}}}{\sum_{j=1}^{n} e^{\boldsymbol{w_{y_j}^T x_i + b_{y_j}}}} \tag{2}$$

3) AVERAGE-CENTER ALIGNMENT LOSS (ACA)

The purpose of average-center alignment loss (ACA) is to focus on minimizing the variations in each class while maintaining the ability to distinguish between characteristics of various classes. The domain gap between several meta-train domains can be reduced by adding average-center alignment loss to make the embedding domain invariant. We determine the embedding center for all mean embeddings of meta-train domains. After optimizing these embedding centers, the center points find out the better destination to gets closer to other data points of its class and reduce the gap between two classes ("1" and "0"). As embedding centers get close to each other, the embedding distribution of class samples get closer. As a result of that, the domain gap of different meta-train domains can also be reduced. Therefore, the alignment of all meta-train domains becomes easier to generalize. The average-center alignment loss is only used in meta-train domains. The loss is formulated as:

$$c_{r_i} = \frac{1}{P} \sum_{i=1}^{P} \boldsymbol{F_{ri}^{T_{sj}}} \tag{3}$$

$$c_{f_i} = \frac{1}{N} \sum_{i=1}^{N} \boldsymbol{F_{fi}^{T_{sj}}} \tag{4}$$

$$c_{avg} = \frac{1}{n} \sum_{i=1}^{n} \left(c_{r_i} + c_{f_i}\right) \tag{5}$$

$$L_{ACA} = \frac{1}{n} \sum_{i=1}^{n} |\left(c_{r_i} + c_{f_i}\right) - c_{avg}|_2 \tag{6}$$

where $F_{ri}^{T_{sj}}$, $F_{fi}^{T_{sj}}$ are embedding features of real and fake samples respectively in the meta-train domain j^{th}. In a batch $B(B = P + N)$ sampled from domain T_{sj}, c_{r_i} represents the mean embedding of real samples T_{sj}, c_{f_i} represents the mean embedding of fake samples, n is the number of meta-train domains, c_{avg} is embedding the center of all meta-train domains. The c_{r_i}, c_{f_i} should ideally be updated whenever the deep features get changed. However, it is inefficient and perhaps impracticable to calculate the mean of the embedding of each class in each iteration, when taking into consideration the entire training set. Therefore, instead of calculating and updating the centers for the entire training set, we perform an update on B number of identities in a batch.

E. LEARNING PROCESS

In the meta-learning algorithm, we train the model over a distribution of related tasks and look for a better model parameter that can be used in a variety of similar tasks and can easily be adopted in new tasks.

1) META-TRAIN

In each iteration, we apply meta-splitting, so as to get meta-train sets T_s^{train} and meta-test sets T_s^{test}. For meta-train, the T_s^{train} is used to calculate the loss function in the training stage. In each task T_{Si}^{train} of T_s^{train}, the data points are divided into batch sizes B, which contain fake samples and real samples. The purpose of the meta-training stage is to calculate the loss of each task based on the binary classification model $f(\theta)$, where θ represents the model parameters. The loss function of the meta-train stage L_{train} is formulated as:

$$L_{train} = L_{PAL} + L_{SOF} + L_{ACA} \tag{7}$$

Here λ is a hyper-parameter to balance the average-center alignment loss (ACA) and other losses. Because the average-center alignment loss can reduce the domain gap between several meta-train domains, it also makes the distribution of data closer to its center point.

2) META-TEST

After the meta-train, in each iteration, the model is validated on the meta-test sets T_s^{test} and meta sets of unseen target domains T_T with a different distribution. The pair-attention loss and softmax loss are calculated to update parameters. In order to allow the model to generalize across domains. The loss of the meta-test is calculated as follows:

$$L_{test} = L_{PAL} + L_{SOF} \tag{8}$$

After calculating the loss function of the meta-train and the meta-test, we need to update the parameters of the inner loop. The gradient is synthesized and updated as follows:

$$\nabla g_\theta = \gamma \nabla_\theta L_{train} + (1-\gamma)\, \nabla_\theta L_{test} \tag{9}$$

$$g_\theta^* \leftarrow g_\theta + \nabla g_\theta \tag{10}$$

3) META-OPTIMIZATION

After updating the model in the inner loop, the model needs to update thoroughly in the outer loop. We use stochastic gradient descent (SGD) to optimize.

Algorithm 1 Meta Deepfake Detection for Generalization of Deepfake Detection Problem

```
Input: Source domains and target domains:
  $T_S = \{T_{S_1}, T_{S_2}, \ldots, T_{S_N}\}; N > 1$
  $T_T = \{T_{T1}, T_{T2}, \ldots, T_{TM}\}; M1$
Init: A pre-train model $f(\theta)$ parameterized by a parameter $\theta$, distribution over task $p(T)$. Batch size of $B$, hyper-parameter: $\alpha$, $\beta$, $\gamma$, $\lambda$.
For iteration in max_iteration do:
  Initialize the gradient $g_\theta$ equal to 0:
  Meta Splitting:
    Meta-train: $T_s^{train} = \{T_{S_1}^{train}, T_{S_2}^{train}, \ldots, T_{S_N}^{train}$
    Meta-test: $T_s^{test} = \{T_{S_1}^{test}, T_{S_2}^{test}, \ldots, T_{S_N}^{test}$
  For each task $(T_{Si})$ in task $(T_S)$ do:
    Sample k data point and its label
    $T_{Si}^{train} = \{(x_1^{train}, y_1^{train}), (x_2^{train}, y_2^{train}), \ldots, (x_k^{train}, y_k^{train})$
    $T_{Si}^{test} = \{(x_1^{test}, y_1^{test}), (x_2^{test}, y_2^{test}), \ldots, (x_k^{test}, y_k^{test})$
    Meta-train:
    Create a batch $B$ samples of meta trainset $T_{Si}^{train}$,
    Create embedding features of real and fake samples $F_{ri}, F_{fi}$
    Calculate loss function:
      $\lambda L_{train} = L_{PAL} + L_{SOF} + L_{ACA}$
    Update model parameter by:
      $\nabla \theta_i^* = \theta - \alpha_\theta L_{train}$
    Meta-test:
      Test with $T_{Si}^{test}$: $L_{PAL}^{test}, L_{SOF}^{test}$
      Test with $T_T$: $L_{PAL}^{target}, L_{SOF}^{target}$
      $L_{PAL} = L_{PAL}^{test} + L_{PAL}^{target}$
      $L_{SOF} = L_{SOF}^{test} + L_{SOF}^{target}$
      $L_{test} = L_{PAL} + L_{SOF}$
    Gradient synthetic:
      $\nabla g_\theta = \gamma_\theta L_{train} + (1-\gamma)\nabla_\theta L_{test}$
      $\nabla g_\theta^* \leftarrow g_\theta + \nabla g_\theta$
  end
  Meta-optimization:
  Update $\theta^* \leftarrow \theta - \beta g_\theta^*$ by SGD
end
```

IV. EXPERIMENTS

To evaluate the quality of our proposed MDD, we use open datasets of facial synthesis, such as DFDC [12], Celeb-DF-v2 [13], FaceForensics++ [9]. The DFDC dataset, which has over 100,000 total videos gathered from 3,426 paid actors and was created using a variety of Deepfake, GAN-based and unlearned algorithms, the DFDC dataset is a sizable face swap video dataset and freely accessible. The Celeb-DF-v2 presents a large-scale Deepfake video dataset based on the development and evaluation of improved deepfake synthesis algorithms. The Celeb-DF-v2 contains 5639 high-quality Deepfake videos. The FaceForensics++ dataset contains four state-of-the-art methods for facial manipulation, namely, Deepfakes, FaceSwap, Face2Face, and NeuralTextures. Each method has different manipulated techniques and algorithms. Corresponding to each method, it includes 1000 original videos (real videos) and 1000 manipulated videos (fake videos). This dataset released raw videos and compressed videos (high-quality videos and low-quality videos).

A. EVALUATION BENCHMARKS

Based on the popular datasets mentioned above, we use videos generated from different methods to use for source and target unseen domains. We utilize variety here to illustrate the large gap between target unseen domains and source domains. In real-world scenarios, after the models are trained, the model is validated with many different manipulated videos. Even the model needs to be evaluated with videos generated from the specific method that the model has never been trained before (which is called the unseen domain). Target unseen domain aims to simulate this situation. The detailed content of the evaluation benchmark is illustrated in Table 1. Inspired by [46], we take advantage of similar evaluation benchmarks. The purpose is also to compare with related researches.

TABLE 1. Seven evaluation benchmarks. The FaceForensics++ dataset use compressed videos. "C23" means higher quality (constant rate quantization parameter equal to 23), "C40" means lower quality (using quantization parameter equal to 40). For benchmarking in the unseen domain, source domains for training and target domains for testing are considered. CID: crossing intra-datasets. CVD: crossing variety of datasets.

Name	Source domains	Target domain(s)	COMPRESSION
CID-DF23/40	NeuralTextures FaceSwap Face2Face	DeepFakes	C23/40
CID-FF23/40	NeuralTextures FaceSwap DeepFakes	Face2Face	C23/40
CID-FS23/40	NeuralTextures Face2Face DeepFakes	FaceSwap	C23/40
CID-NT23/40	FaceSwap Face2Face DeepFakes	NeuralTextures	C23/40
CVD-CV23-1	Celeb-DF-v2	NeuralTextures FaceSwap Face2Face DeepFakes	C23
CVD-CV23-2	NeuralTextures FaceSwap Face2Face DeepFakes	NeuralTextures FaceSwap Face2Face DeepFakes Celeb-DF-v2 DFDC	C23
CVD-CV23-3	NeuralTextures FaceSwap Face2Face DeepFakes Celeb-DF-v2	NeuralTextures FaceSwap Face2Face DeepFakes Celeb-DF-v2 DFDC	C23

B. SETTINGS

The official release of each method in FaceForensics++ dataset included 720 videos for training, 140 videos for validation, and 140 videos for testing. In each method of FaceForensics++ dataset, we use a training set for source domains and a testing set for target domains. (For example, in CID-DF23, NeuralTexture, FaceSwap, Face2Face, and the original video are 720 videos for each method in the source domain. DeepFakes, and the original video are 140 videos for target domains). The Celeb-DF-v2 dataset contains 5639 high-quality fake videos and 890 real videos. As for the source domains of CVD-CV23-1 and CVD-CV23-3, we have used 6011 original and DeepFake videos from Celeb-DF-v2. For the target domains of CVD-CV23-2 and CVD-CV23-3, we have selected 518 test videos (official release) from Celeb-DF-v2. In CVD-CV23-2 and CVD-CV23-3, we have used DFDC test set for target domains. All these videos are sampled and extracted to the face, we have used multitask cascaded convolutional network (MTCNN) [50] library for facial extraction. We only choose 10 frames of facial extraction for training and testing. The extracted face images are resized to 224 × 224 RGB format.

C. IMPLEMENTATION DETAILS

We have used EfficientNet-B0 [59] as a single backbone $f(\theta)$ with 5.3M parameters. The meta-train step-size α, the meta optimization learning rate β, the hyper-parameter γ (which balances meta-train loss and meta-test loss), and the hyper-parameter λ to balance the PAL, SOF loss, and ACA loss are initially set to 0.0005, 0.0005, 0.5, 0.01, respectively. The batch-size B is set to 32. To evaluate the performance of the model, our comparisons include (i) Base: The model was pre-trained on ImageNet [60] without being fine-tuned on our benchmarks. (ii) FT-Base: Based on our benchmarks, the base model is fine-tuned with the same training datasets. This method is for a fair comparison with our MDD. (iii) Multi-task [61]: This method proposed a multi-task learning approach to improve the generalization of the model. There are two tasks: one task applied to share knowledge to improve the performance of both tasks, and another task shares the data it has collected with another. We have run official code with our benchmarks (iv) MLDG [14]: This method proposed a novel meta-learning method and a model agnostic training procedure for domain generalization. We have adapted it for face forgery detection and trained it with our benchmarks (v): Learning-to-weight (LTW) [46]: This method proposed a domain-general model, known as learning-to-weight, which can balance different weights for face forgery images from various domains. This method is also proposed to handle deepfake detection problems. Some experimental results have been completed on several similar benchmarks in their paper which are reused. Other benchmarks, which have not yet been tested in their proposal, are conducted in our research. (vi): Multi-attentional model (Multi-Att) [62]: This method proposed a multiple spatial attention network and combined the attention maps-guided high-level semantic information with low-level textural features for face forgery detection. We have run the official code with our benchmarks. (vii): Model-Agnostic Meta-Learning (MAML) [42]: This method is a well-known meta-learning

TABLE 2. Performance on the CID-DF23/40, CID-FF23/40, CID-FS23/40, and CID-NT23/40 benchmarks.

Name	Method	AUC	ACC	LOSS
CID-DF23	Base	0.476	0.501	1.391
	FT-Base	0.903	0.811	0.853
	Multi-task [61]	0.871	0.721	0.921
	MLDG [14]	0.907	0.837	**0.772**
	LTW [46]	0.927	0.856	0.792
	Multi-Att [62]	0.889	0.801	0.933
	MAML [42]	0.852	0.789	0.957
	MDD (Ours)	**0.931**	**0.861**	0.781
CID-DF40	Base	0.433	0.495	1.755
	FT-Base	0.742	0.668	1.275
	Multi-task [61]	0.751	0.631	1.075
	MLDG [14]	0.730	0.668	0.921
	LTW [46]	0.756	0.691	**0.715**
	Multi-Att [62]	0.739	0.674	0.939
	MAML [42]	0.721	0.621	1.130
	MDD (Ours)	**0.777**	**0.697**	0.825
CID-FF23	Base	0.541	0.531	1.673
	FT-Base	0.792	0.629	1.298
	Multi-task [61]	0.733	0.591	1.331
	MLDG [14]	0.769	0.634	**1.237**
	LTW [46]	0.802	0.656	1.422
	Multi-Att [62]	0.797	0.637	1.433
	MAML [42]	0.751	0.609	1.467
	MDD (Ours)	**0.821**	**0.658**	1.325
CID-FF40	Base	0.521	0.492	1.482
	FT-Base	0.669	0.601	1.584
	Multi-task [61]	0.621	0.541	1.358
	MLDG [14]	0.621	0.573	1.737
	LTW [46]	**0.724**	**0.657**	**1.025**
	Multi-Att [62]	0.671	0.622	1.249
	MAML [42]	0.611	0.658	1.499
	MDD (Ours)	0.691	0.641	1.155
CID-FS23	Base	0.528	0.449	1.973
	FT-Base	0.579	0.534	1.340
	Multi-task [61]	0.563	0.509	1.398
	MLDG [14]	0.621	0.538	1.521
	LTW [46]	0.640	0.549	**1.233**
	Multi-Att [62]	0.571	0.522	1.368
	MAML [42]	0.627	0.544	1.333
	MDD (Ours)	**0.658**	**0.552**	1.297
CID-FS40	Base	0.479	0.459	3.180
	FT-Base	0.609	0.574	1.403
	Multi-task [61]	0.533	0.521	1.708
	MLDG [14]	0.611	0.589	1.330
	LTW [46]	**0.681**	**0.625**	**1.179**
	Multi-Att [62]	0.528	0.489	1.698
	MAML [42]	0.601	0.584	1.429
	MDD (Ours)	0.649	0.611	1.268
CID-NT23	Base	0.487	0.481	3.473
	FT-Base	0.764	0.614	1.340
	Multi-task [61]	0.751	0.603	1.381
	MLDG [14]	0.783	0.635	1.531
	LTW [46]	0.773	0.653	1.561
	Multi-Att [62]	0.748	0.618	1.379
	MAML [42]	0.729	0.601	1.435
	MDD (Ours)	**0.791**	**0.668**	**1.321**
CID-NT40	Base	0.484	0.467	3.140
	FT-Base	0.618	0.581	1.740
	Multi-task [61]	0.608	0.561	1.781
	MLDG [14]	0.611	0.568	2.142
	LTW [46]	0.608	0.585	1.763
	Multi-Att [62]	0.577	0.521	2.344
	MAML [42]	0.615	0.578	2.019
	MDD (Ours)	**0.621**	**0.591**	**1.621**

model. We have adapted it for face forgery detection and trained it with our benchmarks.

D. EVALUATION METRICS

For performance evaluation, we use the area under the receiver operating characteristic curve (AUC). The receiver operating characteristic (ROC) is used to display a classifier to select the classification threshold. AUC is an area covered by the ROC curve. Moreover, we have used the accuracy score (ACC) for evaluating classification models. Another metric we have applied to measure our model performance is a log loss function. We have chosen a log loss metric because it measures how well the predicted probability close to the corresponding actual value and it is appropriate for binary classification, where "0" represents the real class and "1" represents the fake class.

$$LogLoss = -\frac{1}{n}\sum_{i=1}^{n}[y_i \log(\hat{y}_i) + (1 - y_i)log(1 - \hat{y}_i)] \qquad (11)$$

E. EVALUATION RESULTS

1) CID COMPARISONS

From the results in Table 2, our proposal achieves superior results in most of the benchmarks. The base model is pre-trained on ImageNet. Because without being fine-tuned on our benchmarks, the results are frequently insufficient to identify false information. The FT-Base model is fine-tuned on our benchmarks which can detect fake images but can not generalize well for the target domains, especially for low-quality images/videos. The method of Multi-task, MLDG, and Learning-to-weight (LTW) have different approaches. Each proposal offers different solutions to generalize the model in order to identify tampering from as many sources

TABLE 3. **Performance on the CVD-CV23-1, CVD-CV23-2, and CVD-CV23-3 benchmarks.**

Name	Method	AUC	ACC	LOSS
CVD-CV23-1	FT-Base	0.582	0.557	2.648
	MLDG [14]	0.682	0.649	1.699
	LTW [46]	0.693	0.663	1.728
	Multi-Att [62]	0.669	0.638	1.802
	MAML [42]	0.661	0.618	1.695
	MDD (Ours)	**0.708**	**0.668**	**1.689**
CVD-CV23-2	FT-Base	0.672	0.707	1.352
	MLDG [14]	0.765	0.719	**1.074**
	LTW [46]	0.772	0.734	1.519
	Multi-Att [62]	0.658	0.611	1.539
	MAML [42]	0.719	0.647	1.405
	MDD (Ours)	**0.788**	**0.754**	1.101
CVD-CV23-3	FT-Base	0.717	0.685	3.219
	MLDG [14]	0.782	0.728	2.190
	LTW [46]	0.793	0.762	1.898
	Multi-Att [62]	0.681	0.674	3.089
	MAML [42]	0.739	0.677	2.785
	MDD (Ours)	**0.821**	**0.779**	**1.789**

as possible. It is important to note that the outcomes of LTW are fairly promising. The results of our method achieve the best result on higher-quality images. Compare to FT-Base on AUC, our method improves the performance from 0.903 to 0.931 in CID-DF23, from 0.742 to 0.777 in CID-DF40, from 0.792 to 0.821 in CID-FF23, from 0.669 to 0.691 in CID-FF40, from 0.579 to 0.658 in CID-FS23, from 0.609 to 0.681 in CID-FS40, from 0.764 to 0.791 in CID-NT23, and from 0.618 to 0.621 in CID-NT40. This demonstrates that our method improves the generalization of the model in all of the CID benchmarks.

2) CVD COMPARISONS

Results on the CVD benchmark are shown in Table 3. We focus on the performance across the datasets, the target domains are the test sets from many datasets. The obtained results show that our proposal has improved the quality of the model in all benchmarks. Our proposal can compare with the most basic and commonly used model FT-Base. The performance improvements on AUC from 0.582 to 0.708 with the CVD-CV23-1 benchmark, from 0.672 to 0.788 with the CVD-CV23-2 benchmark, and from 0.717 to 0.821 with the CVD-CV23-3 benchmark. This shows that our proposal has increased the generalization of the basic model when tested with benchmarks.

Results in Table 4 show the effect of backbones with and without our proposal. We use the CID-DF23 benchmark and test with different architectures (light and heavy parameters). The observed results demonstrate that our method is model-independent and can improve the performance of the model irrespective of the types of architectures. The model is less effective for complex models than simples model. Because the higher the performance of the model, the harder it is to increase when the result is as high as a certain level. The backbones used for the experiment are the good backbones used in the image classification. Therefore, the difference in results obtained between the backbones is usually not large.

TABLE 4. **Performance on different backbone architectures with or without our proposal on the CID-DF23 benchmark.**

Backbone	AUC	ACC
EfficientNet-B0 [59]	0.903	0.811
EfficientNet-B0 + Ours	**0.931**	**0.861**
EfficientNet-B4 [59]	0.921	0.841
EfficientNet-B4 + Ours	**0.938**	**0.871**
EfficientNet-B7 [59]	0.956	0.903
EfficientNet-B7 + Ours	**0.967**	**0.911**
ResNet-50 [63]	0.911	0.820
ResNet-50 + Ours	**0.929**	**0.827**

TABLE 5. **Ablation Study of loss function on CID-DF23 benchmarks.**

Ablation Study- Loss Function			
	AUC	ACC	LOSS
FT-Base	0.903	0.811	0.853
Without PAL	0.918	0.838	0.819
Without ACA	0.921	0.842	0.802
Without PAL and ACA (only SOF)	0.907	0.829	0.846
Ours-Full	**0.931**	**0.861**	**0.781**

TABLE 6. **Ablation Study of data preprocessing technique on CID-DF23 benchmarks.**

	AUC	ACC	LOSS
Without Block Shuffling Transformation	0.919	0.832	0.847
Ours-Full	**0.931**	**0.861**	**0.781**

3) EFFECTIVENESS OF DIFFERENT COMPONENTS

We compare our entire MDD with three degraded versions on the CID-DF23 benchmarks to assess the efficacy of various components. The first component is pair-attention loss (PAL), which prioritizes increasing negative and positive pairings and distinguishing positive from negative input. The second component is the average-center alignment loss (ACA), which focuses on lowering the variability within each class while maintaining the ability to distinguish between attributes of other classes. The third component is both the pair-attention loss and the average-center alignment loss. The efficiency of each performance component is displayed in Table 5. When any of them are eliminated, the performance decreases. The quality of the model degrades the greatest when the pair-attention loss and average-center alignment loss are not employed. This demonstrates the impact of proposed loss functions on the quality of the model.

Table 6 displays the effects of block shuffling transformation. The results of the model fluctuate around an average of 0.919 of AUC if block shuffling transformation is not used in the data preprocessing. We enhance performance using the block shuffling transformation approach, going from an AUC of 0.919 to 0.931. Improvement with ACC is from 0.832 to 0.861. The loss then decreases from 0.84 to 0.78. This demonstrates the block shuffling transformation improves the performance of the model.

V. CONCLUSION

In this paper, we propose an approach that can improve the generalization of the model, named Meta Deepfake Detection model (MDD). We also apply block shuffling transformation to enhance the performance and reduce the overfitting problem. Moreover, we design two loss functions Pair-Attention Loss and Average-Center Alignment Loss, aggregate with softmax loss to update and learn across domains. We show that by using MDD, we can generalize the unseen domain, as demonstrated in the experiment using several benchmarks. For future work, we will find a new strategy to develop MDD and experiment with more benchmarks.

REFERENCES

[1] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *Commun. ACM*, vol. 63, no. 11, pp. 139–144, 2020.

[2] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," 2013, *arXiv:1312.6114*.

[3] D. J. Rezende, S. Mohamed, and D. Wierstra, "Stochastic backpropagation and approximate inference in deep generative models," in *Proc. Int. Conf. Mach. Learn.*, 2014, pp. 1278–1286.

[4] R. Wu, G. Zhang, S. Lu, and T. Chen, "Cascade EF-GAN: Progressive facial expression editing with local focuses," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 5021–5030.

[5] Y. Shen, J. Gu, X. Tang, and B. Zhou, "Interpreting the latent space of GANs for semantic face editing," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 9243–9252.

[6] C.-H. Lee, Z. Liu, L. Wu, and P. Luo, "MaskGAN: Towards diverse and interactive facial image manipulation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 5549–5558.

[7] F. Matern, C. Riess, and M. Stamminger, "Exploiting visual artifacts to expose deepfakes and face manipulations," in *Proc. IEEE Winter Appl. Comput. Vis. Workshops (WACVW)*, Jan. 2019, pp. 83–92.

[8] L. Li, J. Bao, T. Zhang, H. Yang, D. Chen, F. Wen, and B. Guo, "Face X-ray for more general face forgery detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 5001–5010.

[9] A. Rossler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Niessner, "FaceForensics++: Learning to detect manipulated facial images," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 1–11.

[10] X. Wu, Z. Xie, Y. Gao, and Y. Xiao, "SSTNet: Detecting manipulated faces through spatial, steganalysis and temporal features," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2020, pp. 2952–2956.

[11] Y. Qian, G. Yin, L. Sheng, Z. Chen, and J. Shao, "Thinking in frequency: Face forgery detection by mining frequency-aware clues," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2020, pp. 86–103.

[12] B. Dolhansky, J. Bitton, B. Pflaum, J. Lu, R. Howes, M. Wang, and C. C. Ferrer, "The DeepFake detection challenge (DFDC) dataset," 2020, *arXiv:2006.07397*.

[13] Y. Li, X. Yang, P. Sun, H. Qi, and S. Lyu, "Celeb-DF: A large-scale challenging dataset for DeepFake forensics," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 3207–3216.

[14] D. Li, Y. Yang, Y.-Z. Song, and T. Hospedales, "Learning to generalize: Meta-learning for domain generalization," in *Proc. AAAI Conf. Artif. Intell.*, 2018, vol. 32, no. 1, pp. 1–8.

[15] K. Hsu, S. Levine, and C. Finn, "Unsupervised learning via meta-learning," 2018, *arXiv:1810.02334*.

[16] Z. Li, F. Zhou, F. Chen, and H. Li, "Meta-SGD: Learning to learn quickly for few-shot learning," 2017, *arXiv:1707.09835*.

[17] R. Natsume, T. Yatagawa, and S. Morishima, "RSGAN: Face swapping and editing using face and hair representation in latent spaces," 2018, *arXiv:1804.03447*.

[18] Y. Nirkin, Y. Keller, and T. Hassner, "FSGAN: Subject agnostic face swapping and reenactment," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 7184–7193.

[19] Y. Nirkin, Y. Keller, and T. Hassner, "FSGANv2: Improved subject agnostic face swapping and reenactment," 2022, *arXiv:2202.12972*.

[20] Z. Xu, Z. Hong, C. Ding, Z. Zhu, J. Han, J. Liu, and E. Ding, "MobileFaceSwap: A lightweight framework for video face swapping," 2022, *arXiv:2201.03808*.

[21] Z. He, W. Zuo, M. Kan, S. Shan, and X. Chen, "AttGAN: Facial attribute editing by only changing what you want," *IEEE Trans. Image Process.*, vol. 28, no. 11, pp. 5464–5478, Nov. 2019.

[22] M. Liu, Y. Ding, M. Xia, X. Liu, E. Ding, W. Zuo, and S. Wen, "STGAN: A unified selective transfer network for arbitrary image attribute editing," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3673–3682.

[23] Y. Choi, Y. Uh, J. Yoo, and J.-W. Ha, "StarGAN V2: Diverse image synthesis for multiple domains," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 8188–8197.

[24] R. Durall, J. Jam, D. Strassel, M. H. Yap, and J. Keuper, "FacialGAN: Style transfer and attribute manipulation on synthetic faces," 2021, *arXiv:2110.09425*.

[25] S. Ha, M. Kersner, B. Kim, S. Seo, and D. Kim, "Marionette: Few-shot face reenactment preserving identity of unseen targets," in *Proc. AAAI Conf. Artif. Intell.*, 2020, vol. 34, no. 7, pp. 10893–10900.

[26] X. Zeng, Y. Pan, M. Wang, J. Zhang, and Y. Liu, "Realistic face reenactment via self-supervised disentangling of identity and pose," in *Proc. AAAI Conf. Artif. Intell.*, 2020, vol. 34, no. 7, pp. 12757–12764.

[27] J. Zhang, X. Zeng, M. Wang, Y. Pan, L. Liu, Y. Liu, Y. Ding, and C. Fan, "FReeNet: Multi-identity face reenactment," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 5326–5335.

[28] Y. Guo, K. Chen, S. Liang, Y.-J. Liu, H. Bao, and J. Zhang, "AD-NeRF: Audio driven neural radiance fields for talking head synthesis," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 5784–5794.

[29] S. Tripathy, J. Kannala, and E. Rahtu, "FACEGAN: Facial attribute controllable rEenactment GAN," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2021, pp. 1329–1338.

[30] H. Dang, F. Liu, J. Stehouwer, X. Liu, and A. K. Jain, "On the detection of digital face manipulation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 5781–5790.

[31] G. Jia, M. Zheng, C. Hu, X. Ma, Y. Xu, L. Liu, Y. Deng, and R. He, "Inconsistency-aware wavelet dual-branch network for face forgery detection," *IEEE Trans. Biometrics, Behav., Identity Sci.*, vol. 3, no. 3, pp. 308–319, Jul. 2021.

[32] H. H. Nguyen, J. Yamagishi, and I. Echizen, "Capsule-forensics: Using capsule networks to detect forged images and videos," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2019, pp. 2307–2311.

[33] Y. Huang, F. Juefei-Xu, Q. Guo, Y. Liu, and G. Pu, "FakeLocator: Robust localization of GAN-based face manipulations," *IEEE Trans. Inf. Forensics Security*, vol. 17, pp. 2657–2672, 2022.

[34] L. Guarnera, O. Giudice, and S. Battiato, "DeepFake detection by analyzing convolutional traces," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2020, pp. 666–667.

[35] D. Afchar, V. Nozick, J. Yamagishi, and I. Echizen, "MesoNet: A compact facial video forgery detection network," in *Proc. IEEE Int. Workshop Inf. Forensics Secur. (WIFS)*, Dec. 2018, pp. 1–7.

[36] U. A. Ciftci, I. Demir, and L. Yin, "FakeCatcher: Detection of synthetic portrait videos using biological signals," *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, Jul. 15, 2020, doi: 10.1109/TPAMI.2020.3009287.

[37] Y. Ru, W. Zhou, Y. Liu, J. Sun, and Q. Li, "Bita-Net: Bi-temporal attention network for facial video forgery detection," in *Proc. IEEE Int. Joint Conf. Biometrics (IJCB)*, Aug. 2021, pp. 1–8.

[38] B. Chen, T. Li, and W. Ding, "Detecting DeepFake videos based on spatiotemporal attention and convolutional LSTM," *Inf. Sci.*, vol. 601, pp. 58–70, Jul. 2022.

[39] F. Marra, C. Saltori, G. Boato, and L. Verdoliva, "Incremental learning for the detection and classification of GAN-generated images," in *Proc. IEEE Int. Workshop Inf. Forensics Secur. (WIFS)*, Dec. 2019, pp. 1–6.

[40] H. Khalid and S. S. Woo, "OC-FakeDect: Classifying DeepFakes using one-class variational autoencoder," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2020, pp. 656–657.

[41] Z. Wang, Y. Guo, and W. Zuo, "DeepFake forensics via an adversarial game," *IEEE Trans. Image Process.*, vol. 31, pp. 3541–3552, 2022.

[42] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 1126–1135.

[43] L. Zintgraf, K. Shiarli, V. Kurin, K. Hofmann, and S. Whiteson, "Fast context adaptation via meta-learning," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 7693–7702.

[44] M. A. Jamal and G.-J. Qi, "Task agnostic meta-learning for few-shot learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 11719–11727.

[45] R. Shao, X. Lan, and P. C. Yuen, "Regularized fine-grained meta face anti-spoofing," in *Proc. AAAI Conf. Artif. Intell.*, 2020, vol. 34, no. 7, pp. 11974–11981.

[46] K. Sun, H. Liu, Q. Ye, Y. Gao, J. Liu, L. Shao, and R. Ji, "Domain general face forgery detection by learning to weight," in *Proc. AAAI Conf. Artif. Intell.*, 2021, vol. 35, no. 3, pp. 2638–2646.

[47] S. Jia, C. Ma, T. Yao, B. Yin, S. Ding, and X. Yang, "Exploring frequency adversarial attacks for face forgery detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 4103–4112.

[48] Z. Yu, J. Wan, Y. Qin, X. Li, S. Z. Li, and G. Zhao, "NAS-FAS: Static-dynamic central difference network search for face anti-spoofing," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 9, pp. 3005–3023, Sep. 2021.

[49] R. Cai, Z. Li, R. Wan, H. Li, Y. Hu, and A. C. Kot, "Learning meta pattern for face anti-spoofing," *IEEE Trans. Inf. Forensics Security*, vol. 17, pp. 1201–1213, 2022.

[50] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, "Joint face detection and alignment using multitask cascaded convolutional networks," *IEEE Signal Process. Lett.*, vol. 23, no. 10, pp. 1499–1503, Oct. 2016.

[51] T. Chuman, W. Sirichotedumrong, and H. Kiya, "Encryption-then-compression systems using grayscale-based image encryption for JPEG images," *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 6, pp. 1515–1525, Jun. 2019.

[52] W. Sirichotedumrong and H. Kiya, "Grayscale-based block scrambling image encryption using YCbCr color space for encryption-then-compression systems," *APSIPA Trans. Signal Inf. Process.*, vol. 8, no. 1, pp. 1–15, 2019.

[53] W. Sirichotedumrong, Y. Kinoshita, and H. Kiya, "Pixel-based image encryption without key management for privacy-preserving deep neural networks," *IEEE Access*, vol. 7, pp. 177844–177855, 2019.

[54] M. Du, S. Pentyala, Y. Li, and X. Hu, "Towards generalizable DeepFake detection with locality-aware AutoEncoder," in *Proc. 29th ACM Int. Conf. Inf. Knowl. Manag.*, Oct. 2020, pp. 325–334.

[55] M. Maung, A. Pyone, and H. Kiya, "Encryption inspired adversarial defense for visual classification," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2020, pp. 1681–1685.

[56] Z. Zhong, L. Zheng, G. Kang, S. Li, and Y. Yang, "Random erasing data augmentation," in *Proc. AAAI Conf. Artif. Intell.*, 2020, vol. 34, no. 7, pp. 13001–13008.

[57] D. A. Coccomini, N. Messina, C. Gennaro, and F. Falchi, "Combining EfficientNet and vision transformers for video DeepFake detection," in *Proc. Int. Conf. Image Anal. Process.* Cham, Switzerland: Springer, 2022, pp. 219–229.

[58] D. Wodajo and S. Atnafu, "DeepFake video detection using convolutional vision transformer," 2021, *arXiv:2102.11126*.

[59] M. Tan and Q. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 6105–6114.

[60] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.

[61] H. H. Nguyen, F. Fang, J. Yamagishi, and I. Echizen, "Multi-task learning for detecting and segmenting manipulated facial images and videos," in *Proc. IEEE 10th Int. Conf. Biometrics Theory, Appl. Syst. (BTAS)*, Sep. 2019, pp. 1–8.

[62] H. Zhao, T. Wei, W. Zhou, W. Zhang, D. Chen, and N. Yu, "Multi-attentional DeepFake detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 2185–2194.

[63] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

VAN-NHAN TRAN received the B.E. degree in electronics-telecommunications engineering from the Vietnam National University Ho Chi Minh City, University of Technology, Vietnam, in 2018. He is currently pursuing the M.S.E. degree in artificial intelligence convergence with Pukyong National University, South Korea. His research interests include computer vision, multimedia security, machine learning, and AI.

SEONG-GEUN KWON received the B.S., M.S., and Ph.D. degrees in electrical engineering from Kyungpook National University, South Korea, in 1996, 1998, and 2002, respectively. He worked as a Senior Engineer with the Mobile Division, Samsung Electronics, from 2002 to 2011. He is currently working as a Professor with the Department of Electronic Engineering, Kyungil University. His research interests include mobile device, multimedia security, and computer vision.

SUK-HWAN LEE received the B.S., M.S., and Ph.D. degrees in electrical engineering from Kyungpook National University, South Korea, in 1999, 2001, and 2004, respectively. He is currently a Professor with the Department of Computer Engineering, Dong-A University. His research interests include multimedia security, digital image processing, and computer graphics. He is a Thesis Editor-in-Chief of *KMMS* journal.

HOANH-SU LE received the B.E. degree in electronics and telecommunication, the M.Sc. and M.B.A. degrees in MIS from the Vietnam National University HCM City, and the Ph.D. degree in MIS from Pukyong National University, South Korea. From 2006 to 2011, he was a Senior Engineer and the Project Team Leader at Global CyberSoft. Since 2011, he has been a Faculty Member with the University of Economics and Law, Vietnam National University Ho Chi Minh City, where he is currently the Dean of the Faculty of Information Systems. His research interests include data analytics, big data, robotics, and AI.

KI-RYONG KWON received the B.S., M.S., and Ph.D. degrees in electronics engineering from Kyungpook National University, in 1986, 1990, and 1994, respectively. He worked at Hyundai Motor Company, from 1986 to 1988, and at the Pusan University of Foreign Language, from 1996 to 2006. He was a Postdoctoral Researcher at the University of Minnesota, USA, from 2000 to 2002. He is currently the Dean of the Engineering College as well as a Professor with the Department of IT Convergence and Application Engineering, Pukyong National University. His research interests include digital image processing, multimedia security, bioinformatics, and machine learning. He was the President of the Korea Multimedia Society, from 2015 to 2016.