

# SELECTIVE DOMAIN-INVARIANT FEATURE FOR GENERALIZABLE DEEPFAKE DETECTION

Yingxin Lai<sup>1</sup>Guoqing Yang<sup>1</sup>Yifan He<sup>2</sup>Zhiming Luo<sup>1,3\*</sup>Shaozi Li<sup>1,3</sup><sup>1</sup>Department of Artificial Intelligence, Xiamen University, Xiamen, China<sup>2</sup>Reconova Technologies, Xiamen, China<sup>3</sup> Fujian Key Laboratory of Big Data Application and Intellectualization for Tea Industry, Wuyi University, WuyiShan, China

## ABSTRACT

With diverse presentation forgery methods emerging continually, detecting the authenticity of images has drawn growing attention. Although existing methods have achieved impressive accuracy in training dataset detection, they still perform poorly in the unseen domain and suffer from forgery of irrelevant information such as background and identity, affecting generalizability. To solve this problem, we proposed a novel framework Selective Domain-Invariant Feature (SDIF), which reduces the sensitivity to face forgery by fusing content features and styles. Specifically, we first use a Farthest-Point Sampling (FPS) training strategy to construct a task-relevant style sample representation space for fusing with content features. Then, we propose a dynamic feature extraction module to generate features with diverse styles to improve the performance and effectiveness of the feature extractor. Finally, a domain separation strategy is used to retain domain-related features to help distinguish between real and fake faces. Both qualitative and quantitative results in existing benchmarks and proposals demonstrate the effectiveness of our approach.

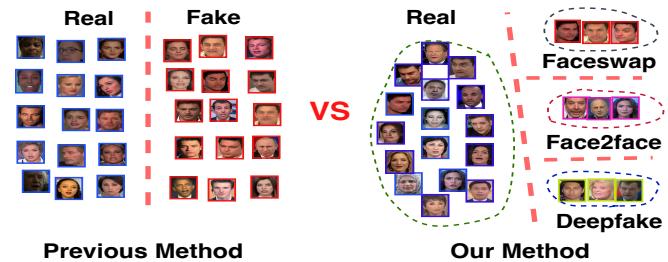
**Index Terms**— DeepFake Detection; Domain Adaption

## 1. INTRODUCTION

Face forgery technologies [1, 2, 3, 4] have been rapidly developed in recent years. The frequent use of these forged images for political framing, fraud, and privacy concerns has raised significant social issues, prompting researchers to seek ways to address these risks.

Convolutional neural networks (CNNs) [5, 6] have shown excellent performance in face forgery detection, but they learn only specific patterns of forgery in the training set and fail to achieve satisfactory performance in other domains. These methods use a convolutional backbone to extract image features and then use a classification header to classify the image, tackling forgery detection as a binary classification problem.

To improve cross-domain performance, several studies have explored how to extract features standard across forgery



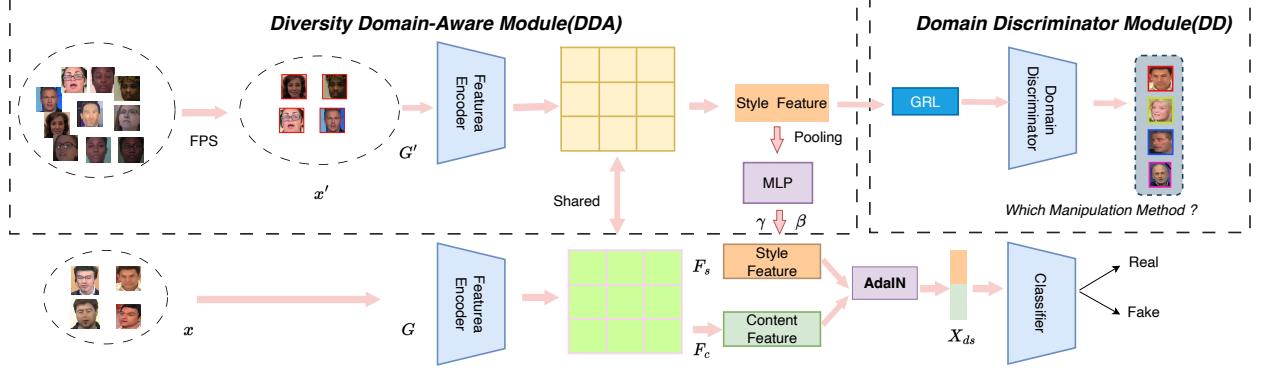
**Fig. 1.** Illustration of how our method differs from previous.

methods [7, 8]. Local-relation [9] by designing a two-stream branch that fuses transformer features with frequency domain features using a cross-attention mechanism. RCCE [10] learns the common representations of real faces by reconstructing face images and mines the differences between real and forgery faces through classification. However, they do not separate features irrelevant to forgery and fail to take full advantage of the intrinsic linkages between different forgery methods. Thus, the mixing of irrelevant information in the features may limit generalization.

Domain adaptation (DA) techniques can mitigate the differences between the training and target domains by using domain alignment, yet applying DA directly to the training of deep forgery detection tends to have very unsatisfactory accuracy due to the non-uniformity of the extracted style samples.

Based on the above observations, we propose a novel framework named SDIF, which preserves domain-related features to help distinguish between real and forgery faces. Specifically, we first design an FPS-based style sample sampling strategy by sampling the real and forgery samples separately with FPS, making all the style samples as far away as possible from each other. Thus, the samples are discrete and uniform. Inspired by [11] we also propose a Diversity Domain-Aware module (DDA), which fuses content features and style features, embeds domain knowledge in a high-dimensional feature-aware task and then preserves domain knowledge for constraints in the task.

\*Corresponding author: zhiming.luo@xmu.edu.cn



**Fig. 2.** The overall framework of our proposed method.

## 2. METHOD

The overall framework of our method is presented in Fig 2, which contains three major key modules: style sample generation, dynamic feature extractor, and feature fusion domain identification module.

### 2.1. Diversity Domain-Aware Module

In this module, the input  $x$  is first fed into the feature extractor  $G$  and performing FPS sampling of style samples  $x'$  into  $G'$  to obtain the feature embedding. The content features and the arbitrary statistic-modified style features are then blended using Adaptive Instance Normalization (AdaIN) to be fed into the domain invariant discriminator  $D$ , allowing  $D$  to concentrate on the input domain invariant features.

To maximize the coverage of rare styles as well as to ensure the diversity of sample space inspired by the point cloud sampling algorithms (point cloud sampling algorithms [12]), we use FPS to sample the style samples according to the proportion of different categories. In detail, we iteratively select  $C$  points from all the points in the  $N$  categories to prevent real and fake images from interacting with each other. The point selected in each iteration is the farthest point from the remaining points. After sampling, we use the feature extractor to get feature embeddings. Specifically, we first use a feature generator to capture multi-scale low-level image information. Then, for content features, we obtain  $F_c$  by batch normalization (BN), while for style features, we use instance normalization (IN) to obtain  $S$ . Subsequently, we utilize Global Average Pooling (GAP) and Multilayer Perceptron (MLP) to derive  $F_s$ , which is employed for the fusion of content information.

$$F_s = \text{MLP}(\text{GAP}(S)) \quad (1)$$

AdaIN is an adaptive stylization module; given a content input  $A$  and a randomly chosen input style input  $A'$ , the stylized normalized can be formed as follows:

$$\text{AdaIN}(A, A', \mu, \sigma) = \gamma(A') \left( \frac{A - \mu(A)}{\sigma(A)} \right) + \beta(A') \quad (2)$$

where  $\mu(\cdot)$  and  $\sigma(\cdot)$  denotes the channel-wise mean and standard deviation.  $\gamma(\cdot)$  and  $\beta(\cdot)$  are the affine parameters generated by the style input  $A'$ . For each category  $N$ , we sample the combination weight  $W^n = [w^1, \dots, w^c]$  from Dirichlet distribution  $B([\alpha^1, \dots, \alpha^c])$  with the concentration parameters  $[\alpha^1, \dots, \alpha^c]$  all set to  $1/C$ . Then, the basis styles of each class  $N$  are linearly combined by  $W^n$  for the aggregation.

$$\mu_{F_s} = W^n \cdot \mu_{base}, \quad \sigma_{F_s} = W^n \cdot \sigma_{base} \quad (3)$$

where  $\mu_{base}$  and  $\sigma_{base}$  are the C bias styles.

With the generated styles, diversity style  $X_{ds}$  are:

$$X_{ds} = \gamma^n(F_s) \left( \frac{F_c - \mu(F_c)}{\sigma(F_c)} \right) + \beta^n(F_s) \quad (4)$$

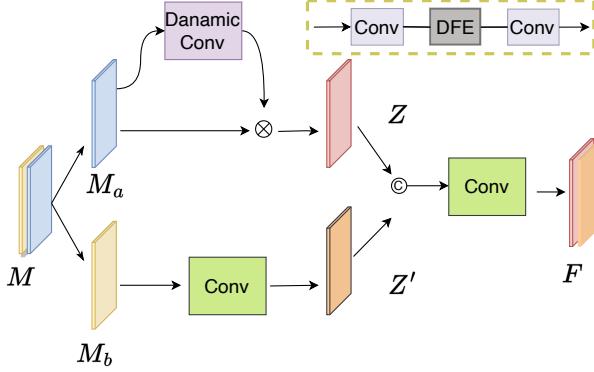
where  $\mu(\cdot)$  and  $\sigma(\cdot)$  denotes the channel-wise mean and standard deviation.  $\gamma(\cdot)$  and  $\beta(\cdot)$  are the affine parameters generated by the style input  $F_s$ .

### 2.2. Dynamic Feature Extractor

The utilization of static convolution poses certain challenges in extracting adaptive features, primarily stemming from the diverse array of forgery methods. To address this issue, inspired by [11], we have introduced a Dynamic Feature Extraction module (DFE). As shown in Fig ??, One part is fed into the dynamic convolutional branch, and the other part is passed to the static convolutional branch. The dynamic convolutional branch acquires the representation of half of the channels. This module is added after the convolutional layer of each feature extractor. The input feature  $M$  is split into two parts  $M_a$  and  $M_b$  after channel split.

$$Z = \nu(M_a) \otimes M_a, \quad Z' = \text{Conv}(M_b) \quad (5)$$

where  $\nu(\cdot)$  denotes dynamic convolution operation,  $\text{Conv}$  is a convolution block, in which  $\nu$  depends on the input instance  $M_a$  and  $\otimes$  denotes the element-wise multiplication. We then concatenate  $M_a$  and  $M_b$  in channel dimension and feed the



**Fig. 3.** Illustrations of our designed Dynamic feature Extractor.

result into a convolution block  $\delta(\cdot)$  to produce output features  $F$ , denoted as:

$$F = \delta(\text{concat}(Z, Z')) \quad (6)$$

### 2.3. Domain Discriminator Module

Relevant research [13] has shown evidence that it is crucial to establish an appropriate level of differentiation among different forged methods. At the same time, we also want to make more general content features, this process can be learned through the following optimization problem:

$$\begin{aligned} \min_D \max_G L_{adv}(G, D) = \\ -\mathbb{E}_{(x,y) \sim (X, Y_N)} \sum_{i=1}^N [i = y] \log D(G(x)), \end{aligned} \quad (7)$$

where  $N$  denotes the number of domains,  $x$  represents the training set of samples, and  $y$  is the set of binary labels.  $Y_N$  corresponds to the one-hot encoding of the domain label.  $G$  and  $D$  denote the content feature generator and domain discriminator, respectively. To simultaneously optimize  $G$  and  $D$ , the gradient reversal layer (GRL) [14] is employed by us to reverse the gradient during the backward process.

### 2.4. Loss Function

The total loss function  $L$  of the proposed framework includes the domain differentiation loss for domain learning and the cross-entropy loss  $L_{ce}$  for binary classification, which can be presented as follows:

$$L = L_{ce} + \lambda L_{adv}, \quad (8)$$

where  $\lambda$  is the hyperparameter for balancing the two losses.

**Table 1.** In-domain comparisons on FF++ dataset. Results contain ACC(%) and AUC (%) of high quality (HQ) and low quality (LQ).

Methods	FF++ (HQ)		FF++ (LQ)	
	ACC	AUC	ACC	AUC
Xception [5]	95.04	96.30	84.11	92.50
F <sup>3</sup> -Net (Xception) [22]	97.31	98.10	86.89	93.30
EN-B4 [26]	96.63	99.18	86.67	88.20
MAT (EN-B4) [27]	97.60	99.29	88.69	90.40
SPSL [25]	91.50	95.32	81.57	82.82
RFM [7]	95.69	98.79	87.06	89.83
Local-relation [9]	97.59	99.56	91.47	95.21
RECCE [10]	97.06	99.32	91.03	95.02
<b>Ours</b>	<b>99.19</b>	<b>99.93</b>	<b>92.17</b>	<b>96.14</b>

## 3. EXPERIMENT

### 3.1. Experimental Setting

**Datasets.** Following previous studies [15, 16], we validated our method using four publicly available datasets: FF++ [17], Celeb-DF [18], WildDeepfake [19], and DFDC [20].

**Implementation details.** Our experiments were conducted using PyTorch on a single RTX-3090 GPU. For training, we utilized ResNet18 pre-trained on ImageNet. The batch size is set to 32, and Adam is used for optimization with an initial learning rate of 2e-5. Additionally, we used RetinaFace for face extraction and alignment, resizing the aligned faces to 229×229. The implementation of dynamic convolution was based on [21].

**Evaluation Metrics.** For evaluation, we utilize metrics commonly employed in related research [7, 10, 22, 23, 24], including ACC, AUC, and EER.

### 3.2. Comparison with Other Methods

In this section, we compare the performance of our method with those of the current state-of-the-art model [10, 7, 25, 9]. We first assess the performance within the FF++ [17] datasets. Following that, we examine cross-dataset generalization by training our model on FF++(HQ) [17] and evaluating its performance on Celeb-DF [18], DFDC [20], and FF++(LQ) [17] datasets, it was additionally trained using the NT forgery mode and subsequently evaluated on several different forgery types.

**Intra-dataset Testing.** In Table 1, we present our findings on intra-domain detection performance. Our accuracy scores (ACCs) on FF++(LQ) and Celeb-DF are impressive at 99.19% and 92.17%, respectively, surpassing the results of other methods listed in the table. For comparison, the recent RECCE network achieves ACCs that are 2.13% and 1.14% lower than ours. Although RECCE demonstrates strong performance in

**Table 2.** Cross-domain comparisons of generalization based on AUC (%). We train the model on the HQ dataset of FF++ and evaluate it on Celeb-DF and DFDC.

Methods	Celeb-DF	DFDC
Xception [5]	66.91	67.93
F <sup>3</sup> -Net (Xception) [22]	71.21	72.88
EN-B4 [26]	66.24	66.81
MAT(EN-B4) [27]	76.65	67.34
Face X-ray [25]	74.20	70.00
RFM [7]	67.64	68.01
SRM [8]	79.40	79.70
Local-relation [9]	78.26	76.53
RECCE [10]	77.39	76.75
LTW [28]	77.14	74.58
<b>Ours</b>	<b>81.79</b>	<b>80.65</b>

**Table 3.** Cross-database evaluation on FF++ database (HQ).

Training Set	Method	Testing Set(AUC)		
		DF	FS	F2F
NT	Xception [5]	76.98	70.73	73.57
	Face X-ray [23]	65.31	74.68	74.86
	F <sup>3</sup> -Net (Xception) [22]	81.50	64.49	77.18
	RFM [7]	80.05	64.93	74.50
	RECCE [10]	78.83	63.70	80.89
	<b>Our</b>	<b>83.54</b>	<b>75.14</b>	<b>83.69</b>

accurately learning the distribution of real faces through reconstruction, reaching impressive AUC scores of 99.32% and 95.02% on the corresponding datasets, it still lags behind our technique by an average margin of 0.86%. This highlights the effectiveness of our method, which focuses on different forgery categories, allowing us to capture more forgery-related information.

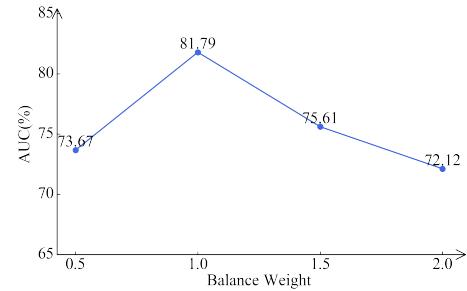
**Cross-dataset Testing.** To validate the generalization performance of our method, we trained it on FF+++HQ data and then tested it in other domains. Table 2 evident that our method achieves the best performance on DFDC and Celeb-DF, with AUC scores of 81.79% and 80.65%, respectively. In comparison, the SRM model [8], which integrates RGB and frequency-domain information, performs reasonably well with AUC scores of 79.40% and 74.70%, respectively. However, its performance remains inferior to our method. Our method shows superior performance on undetected forgery types compared to previous methods, as shown in table 3. These results confirm the feasibility of using domain separation and improved stylistic features to detect forgeries with unknown patterns.

### 3.3. Ablation Studies and Visualization

**Importance of different modules.** The key components of the performance of our method are shown in Table 4, where the model is trained on FF++ and tested on WildDeepfake. We progressively integrate the DDA, DFE, and DD modules into the baseline model, leading to a gradual enhancement

**Table 4.** Ablation study on the proposed components in our method. Cross-dataset evaluation on WildDeepfake dataset.

Baseline	DDA	DFE	DD	AUC	EER
✓				61.21	45.67
✓	✓			62.76	43.72
✓	✓	✓		63.22	43.04
✓	✓	✓	✓	<b>64.01</b>	<b>41.54</b>



**Fig. 4.** Comparative experiments with different balance weight of  $\lambda$  from FF++(HQ) to Celeb-DF.

in the performance of the model. The greatest impact is the addition of the DDA module, e.g., the AUC increases from 61.21% to 62.76%. It indicates that the performance of the in-domain representation of diversity is significantly improved, while the ACC and EER changed by 0.79% and 1.49%, respectively, indicating that the DD module contributes to the domain generalization ability of the model.

**Parameter sensitivity analysis.** In this section, we investigate the effect of balance weight  $\lambda$  on the generalization performance, and we retrain the model using different weight coefficients; as can be seen, the overall best performance of ACC is obtained at  $\lambda$  equals 1.

## 4. CONCLUSION

This paper presents a novel framework to enhance face forgery detection performance in cross-domain. We employ FPS for diverse representations and utilize dynamic convolution to generate varied samples. Additionally, we decouple domain labels to reduce interference from intradomain correlations unrelated to forgery. The effectiveness of our method is demonstrated across multiple benchmark datasets.

**Acknowledgement:** This work is supported by the National Natural Science Foundation of China (No. 62276221, No. 62376232), and the Science and Technology Plan Project of Xiamen (No. 3502ZZ20221025), the Open Project Program of Fujian Key Laboratory of Big Data Application and Intellectualization for Tea Industry, Wuyi University (No. FKLBDAITI202304).

## 5. REFERENCES

- [1] Justus Thies, Michael Zollhofer, Marc Stamminger, Christian Theobalt, and Matthias Nießner, “Face2face: Real-time face capture and reenactment of rgb videos,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2387–2395.
- [2] Justus Thies, Michael Zollhöfer, and Matthias Nießner, “Deferred neural rendering: Image synthesis using neural textures,” *ACM Transactions on Graphics (TOG)*, vol. 38, no. 4, pp. 1–12, 2019.
- [3] “Faceswap,” <https://github.com/MarekKowalski/FaceSwap>, Accessed 2022-10-29.
- [4] “Deepfakes,” <https://github.com/deepfakes/faceswap>, Accessed 2022-10-29.
- [5] Andreas Rossler, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies, and Matthias Nießner, “Faceforensics++: Learning to detect manipulated facial images,” in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 1–11.
- [6] Darius Afchar, Vincent Nozick, Junichi Yamagishi, and Isao Echizen, “Mesonet: a compact facial video forgery detection network,” in *IEEE international workshop on information forensics and security (WIFS)*. IEEE, 2018, pp. 1–7.
- [7] Chengrui Wang and Weihong Deng, “Representative forgery mining for fake face detection,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 14923–14932.
- [8] Yuchen Luo, Yong Zhang, Junchi Yan, and Wei Liu, “Generalizing face forgery detection with high-frequency features,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 16317–16326.
- [9] Shen Chen, Taiping Yao, Yang Chen, Shouhong Ding, Jilin Li, and Rongrong Ji, “Local relation learning for face forgery detection,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021, vol. 35, pp. 1081–1088.
- [10] Junyi Cao, Chao Ma, Taiping Yao, Shen Chen, Shouhong Ding, and Xiaokang Yang, “End-to-end reconstruction-classification learning for face forgery detection,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 4113–4122.
- [11] Qianyu Zhou, Ke-Yue Zhang, Taiping Yao, Xuequan Lu, Ran Yi, Shouhong Ding, and Lizhuang Ma, “Instance-aware domain generalization for face anti-spoofing,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 20453–20463.
- [12] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas, “Pointnet++: Deep hierarchical feature learning on point sets in a metric space,” *Advances in neural information processing systems*, vol. 30, 2017.
- [13] Ying Guo, Cheng Zhen, and Pengfei Yan, “Controllable guide-space for generalizable face forgery detection,” *arXiv preprint arXiv:2307.14039*, 2023.
- [14] Yaroslav Ganin and Victor Lempitsky, “Unsupervised domain adaptation by backpropagation,” in *International conference on machine learning*. PMLR, 2015, pp. 1180–1189.
- [15] Qianyu Zhou, Ke-Yue Zhang, Taiping Yao, Xuequan Lu, Ran Yi, Shouhong Ding, and Lizhuang Ma, “Instance-aware domain generalization for face anti-spoofing,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 20453–20463.
- [16] Zhuo Wang, Zezheng Wang, Zitong Yu, Weihong Deng, Jiahong Li, Tingting Gao, and Zhongyuan Wang, “Domain generalization via shuffled style assembly for face anti-spoofing,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 4123–4133.
- [17] Andreas Rossler, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies, and Matthias Nießner, “Faceforensics++: Learning to detect manipulated facial images,” in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 1–11.
- [18] Yuezun Li, Xin Yang, Pu Sun, Honggang Qi, and Siwei Lyu, “Celeb-df: A large-scale challenging dataset for deepfake forensics,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 3207–3216.
- [19] Bojia Zi, Minghao Chang, Jingjing Chen, Xingjun Ma, and Yu-Gang Jiang, “Wilddeepfake: A challenging real-world dataset for deepfake detection,” in *Proceedings of the 28th ACM international conference on multimedia*, 2020, pp. 2382–2390.
- [20] Brian Dolhansky, Joanna Bitton, Ben Pflaum, Jikuo Lu, Russ Howes, Menglin Wang, and Cristian Canton Ferrer, “The deepfake detection challenge (dfdc) dataset,” *arXiv preprint arXiv:2006.07397*, 2020.
- [21] Yinpeng Chen, Xiyang Dai, Mengchen Liu, Dongdong Chen, Lu Yuan, and Zicheng Liu, “Dynamic convolution: Attention over convolution kernels,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 11030–11039.
- [22] Yuyang Qian, Guojun Yin, Lu Sheng, Zixuan Chen, and Jing Shao, “Thinking in frequency: Face forgery detection by mining frequency-aware clues,” in *European conference on computer vision*. Springer, 2020, pp. 86–103.
- [23] Lingzhi Li, Jianmin Bao, Ting Zhang, Hao Yang, Dong Chen, Fang Wen, and Baining Guo, “Face x-ray for more general face forgery detection,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 5001–5010.
- [24] Yuezun Li and Siwei Lyu, “Exposing deepfake videos by detecting face warping artifacts.,” in *CVPR Workshops*, 2019.
- [25] Honggu Liu, Xiaodan Li, Wenbo Zhou, Yuefeng Chen, Yuan He, Hui Xue, Weiming Zhang, and Nenghai Yu, “Spatial-phase shallow learning: rethinking face forgery detection in frequency domain,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 772–781.
- [26] Mingxing Tan and Quoc Le, “Efficientnet: Rethinking model scaling for convolutional neural networks,” in *International conference on machine learning*. PMLR, 2019, pp. 6105–6114.
- [27] Hanqing Zhao, Wenbo Zhou, Dongdong Chen, Tianyi Wei, Weiming Zhang, and Nenghai Yu, “Multi-attentional deepfake detection,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 2185–2194.
- [28] Ke Sun, Hong Liu, Qixiang Ye, Yue Gao, Jianzhuang Liu, Ling Shao, and Rongrong Ji, “Domain general face forgery detection by learning to weight,” in *Proceedings of the AAAI conference on artificial intelligence*, 2021, vol. 35, pp. 2638–2646.