Deepfake Detection

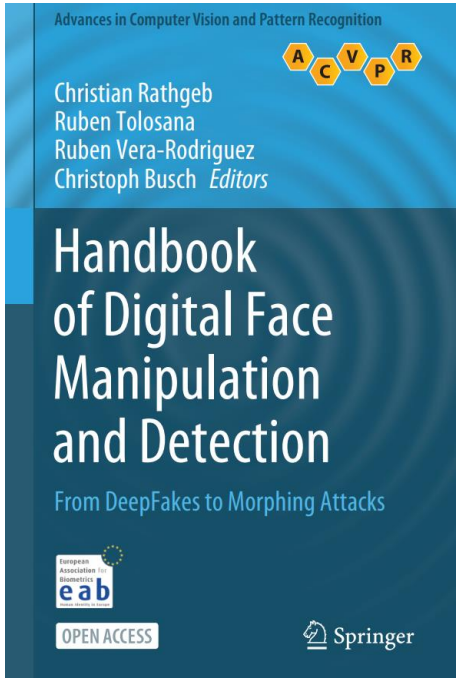
2022.05.20

최종욱

1. Survey

1. Survey

- Survey list
  - (1) Deepfake generation and detection, a survey (2020)
  - (2) Handbook of Digital Face Manipulation and Detection (2022)



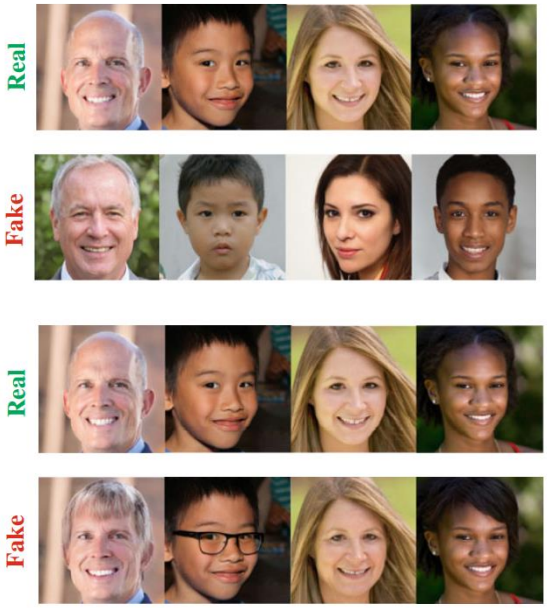
Deepfake generation and detection, a survey (2020)

- (1) Deepfake generation and detection, a survey (2020)
  1. Background
     - SVM, RNN, LSTM, CNN, GAN, Transfer learning
  2. Type of Deepfake
     - Face Swap, Facial reenactment
  3. Detection method
     - Feature based, Machine learning based
     - Dataset
  4. Challenge
     - Evolving technology
     - Lack of high-quality dataset
     - Lack of benchmark

Deepfake generation and detection, a survey (2020)

- Type of Deepfake Generation
  - Face Swap
    - Identity Swap
  - Facial reenactment
    - Face synthesis
    - Attribute manipulate

  + Face morphing, Expression swap



Deepfake generation and detection, a survey (2020)

- Type of Deepfake Detection
  - Feature based method
    - Biometric feature -> 얼굴의 생체적 특징
    - Model feature -> GAN fingerprint
    - Media feature -> temporal, inter-frame dissimilarities,
      semantic, noise, multilevel, frequency, optical flow, …

(a) Fake (b) Fake after GANprintR

  - Machine learning based method
    - Traditional machine learning methods
    - Deep learning methods

Deepfake generation and detection, a survey (2020)

- Detection Dataset

Datasets	Type	Number of images	Number of Videos	Year
Flickr-Faces-HQ (FFHQ) [46]	Images	70,000 (fake)	–	2019
CelebA [45]	Images	30,000 (1024×1024)	-	2017
UADFV [58]	Videos	–	98 (49 real + 49 fake)	2018
WildDeepfake [99]	Videos	–	707	2020
Ding et al. [23]	Images	-	420,053	2019
FaceForensics [80]	Images, videos	1,500,000	1004	2019
FaceForensics++ [81]	Images, videos	1,800,000	3000	2019
DeepfakeTIMIT datasets [53]	Videos	–	320	2018
Celeb-DF [60]	Videos	–	1203 (408 real + 795 fake)	2020
MFC Datasets [35]	Videos, images	35,000,000(100,000 manipulated)	300,000 (4000 manipulated)	2019
FFW [50]	Images	53,000	150	2018
VidTIMIT [55]	Videos	–	620	2019
DFDC Preview [24]	Videos	–	5214	2019
DFDC [25]	Videos	–	4113	2020
DeepfakeDetection [21]	Videos	–	3363	2019

Table 1.2 Identity swap: Publicly available databases

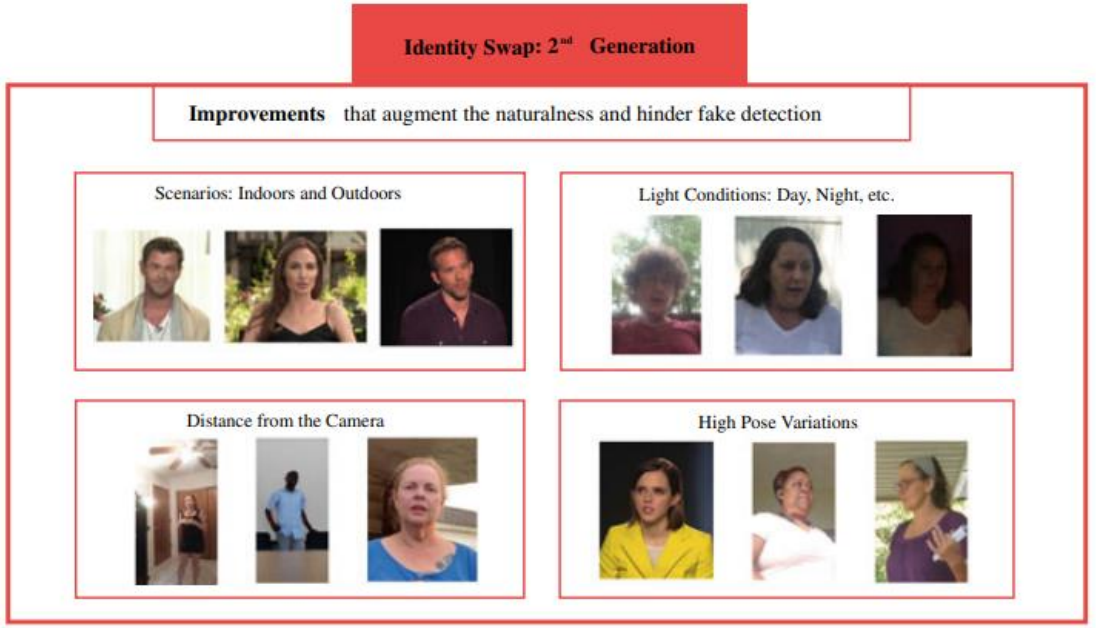
Database	Real videos	Fake videos
1st generation		
UADFV (2018) [58]	49 (Youtube)	49 (FakeApp)
DeepfakeTIMIT (2018) [11]	–	620 (faceswap-GAN)
FaceForensics++ (2019) [20]	1000 (Youtube)	1000 (FaceSwap) 1000 (DeepFake)
2nd generation		
DeepFakeDetection (2019) [66]	363 (Actors)	3068 (DeepFake)
Celeb-DF (2019) [56]	890 (Youtube)	5639 (DeepFake)
DFDC Preview (2019) [59]	1131 (Actors)	4119 (Multiple)
DFDC (2020) [67]	23,654 (Actors)	104,500 (Multiple)
DeeperForensics-1.0 (2020) [60]	50,000 (Actors)	1000 (DeepFake)
WildDeepfake (2020) [61]	3805 (Internet)	3509 (DeepFake)

Deepfake generation and detection, a survey (2020)

- 1st Generation
  - UADFV, DeepfakeTIMIT, FaceForensic++
  1. Face synthesis method
  2. Face detection and alignment method
  3. Rearrangement method

Deepfake generation and detection, a survey (2020)

- 2nd Generation
  - DeepFakeDetection
    - Quality 분류
  - Celeb-DF
    - Deepfake generation method 개선
  - DFDC
    - 8개의 서로 다른 face-swapping 방법 사용
  - DeeperForensics-1.0
    - End-to-End face-swapping framework 사용
  - WildDeepfake
    - Real-world 를 가정해 higher diversity dataset



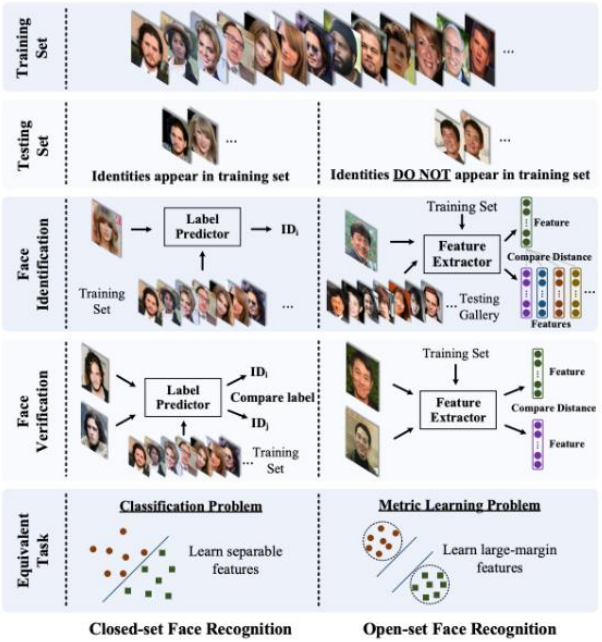
Deepfake generation and detection, a survey (2020)

- Challenges
  - Evolving technologies
    - Black-box, white-box, distortion-minimize attack
    - Add adversarial perturbation
    - Shallow reconstruction to diminish artifact pattern
  - Lack of high-quality dataset
    - 2nd generation dataset
  - Lack of benchmark
    - 범용성 있는 벤치마크 Detection method 필요

2. 최신연구경향

2022 CVPR accepted papers

1. Self-supervised Learning of Adversarial Examples: Towards Good Generalizations for Deepfake Detection
   - Open-set, pristine <-> reference
2. Detecting Deepfakes with Self-Blended Images
   - Open-set, pristine = reference
3. Protecting Celebrities from DeepFake with Identity Consistency Transformer
   - Closed-set
4. DeepFake Disrupter: The Detector of DeepFake Is My Friend
   - 논문 안 나옴
5. Voice-Face Homogeneity Tells Deepfake
   - 논문 안 나옴



1. Self-supervised Learning of Adversarial Examples

- Overview of model

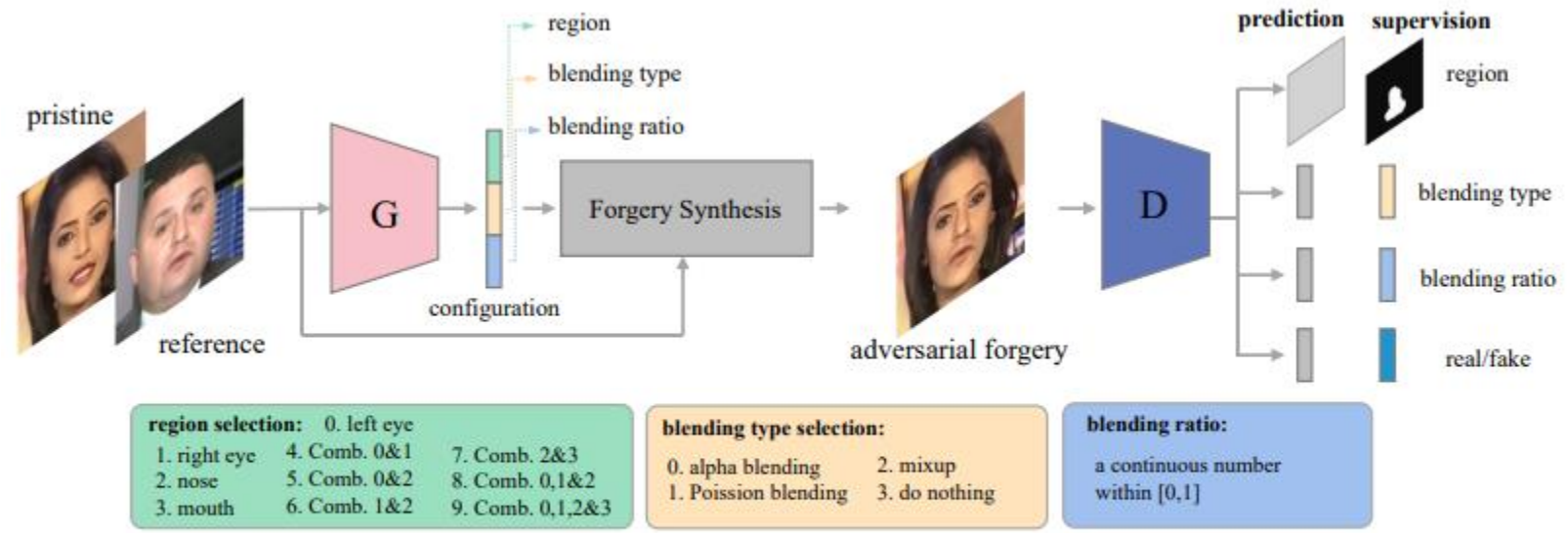
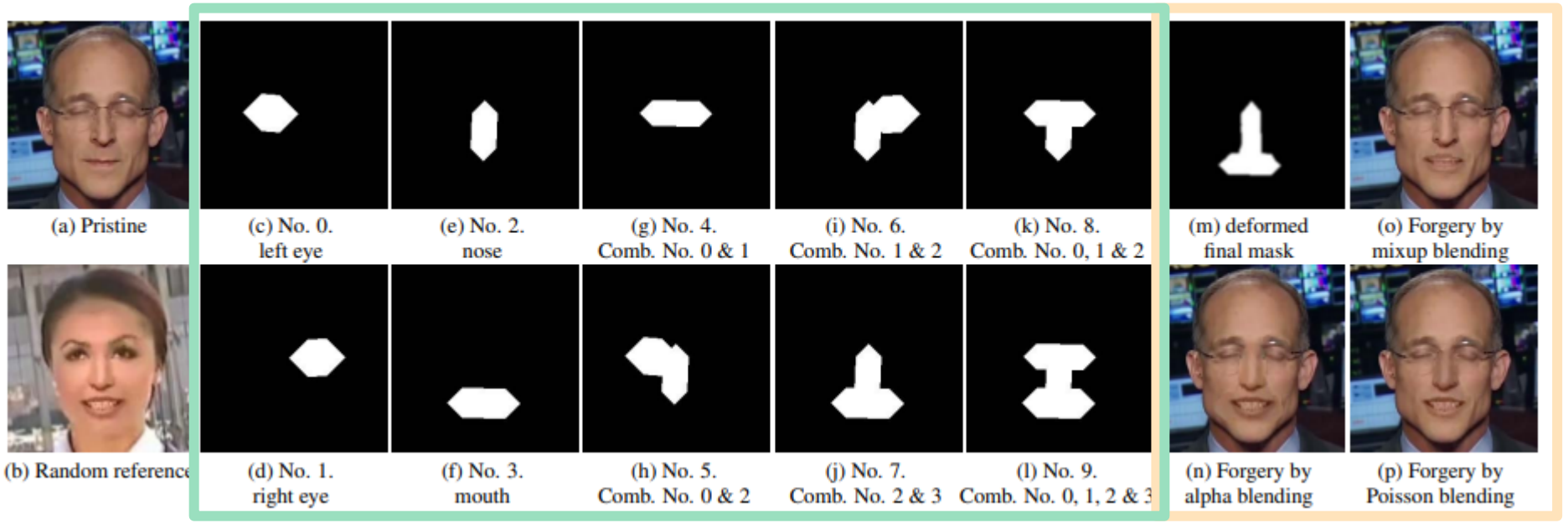


Figure 3. Overview of our model. The synthesizer network (*i.e.* generator) outputs three forgery configurations that are further used to synthesize a new forgery, and these forgery configurations are also used as labels to guide the detector network (*i.e.* discriminator). We train the generator and discriminator in an adversarial manner. Please refer to the text for details.

1. Self-supervised Learning of Adversarial Examples

- Example of region selection and blending

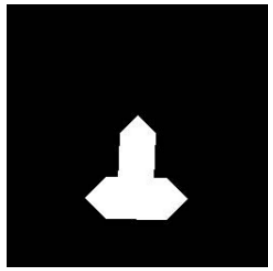


1. Self-supervised Learning of Adversarial Examples

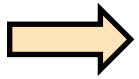
- Blending type
  - 0. alpha blending

(a) Pristine

(b) Random reference



(j) No. 7.
Comb. No. 2 & 3



(n) Forgery by alpha blending

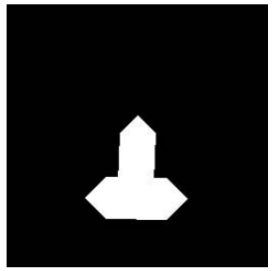
1. Self-supervised Learning of Adversarial Examples

- Blending type
  - 1. poisson blending

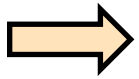
(a) Pristine

(b) Random reference

Comb. No. 2 & 3



(j) No. 7.
Comb. No. 2 & 3



(p) Forgery by
Poisson blending

1. Self-supervised Learning of Adversarial Examples

- Blending type
  - 2. mix-up blending

(a) Pristine

(b) Random reference

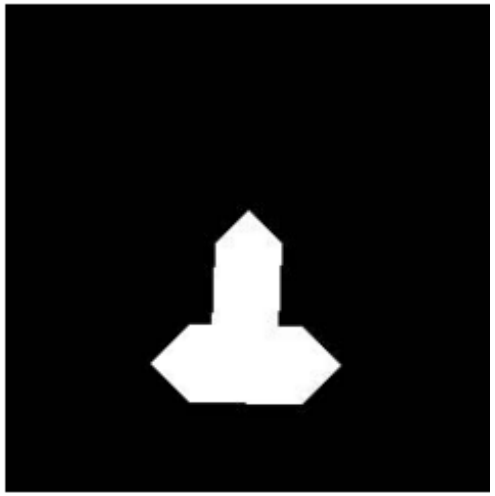
(m) deformed final mask

$$\mathbf{I}_a = A_g \times \mathbf{M}_d * (\mathbf{I}_f - \mathbf{I}_p) + \mathbf{I}_p$$

(o) Forgery by mixup blending

1. Self-supervised Learning of Adversarial Examples

- Blending type, 2. mix-up blending
  - **Gaussian blur with random kernel size**



(j) No. 7.
Comb. No. 2 & 3

(m) deformed
final mask

1. Self-supervised Learning of Adversarial Examples

- Blending type, **2. mix-up blending**
  - Blending ratio -> continuous number, [0, 1]

(m) deformed final mask

(m) deformed final mask

1. Self-supervised Learning of Adversarial Examples

- Joint Training with Self-Supervised Tasks
  - Main task loss → $\mathcal{L}_{Main}$
    - AM-Softmax Loss

    (Angular -> intra-class variability ⬇, Margin -> inter-class variability ⬆)

  - Forgery region estimation loss → $\mathcal{L}_R$
    - $\mathcal{L}_R = \frac{\|\mathbf{M}_{gt} - \mathbf{M}_e\|_1}{H/16 \times W/16}$
  - Blending type estimation loss → $\mathcal{L}_T$
    - T → {0, 1, 2, 3, 4}
    - AM-Softmax Loss
  - Blending ratio estimation loss → $\mathcal{L}_A$
    - $\mathcal{L}_A = \tau \times \|A_{gt} - A_e\|_1$
    - T=2, $\mathcal{T}$ = 1 <-> otherwise, $\mathcal{T}$ = 0

$$\mathcal{L}_{Main} + \alpha\mathcal{L}_R + \mu\mathcal{L}_T + \gamma\mathcal{L}_A$$

1. Self-supervised Learning of Adversarial Examples

- Adversarial Training
  - Synthesizer network → $G(\cdot, \theta)$
  - Discriminator network → $D(\cdot, w)$

$$\min_{w} \max_{\theta} \mathcal{L}, \quad \text{s.t. } \mathcal{L}(\theta, w) = \mathcal{L}_{Main} + \alpha\mathcal{L}_R + \mu\mathcal{L}_T + \gamma\mathcal{L}_A$$

  - Minimize → $w^{t+1} = w^t - \eta \frac{1}{N} \sum_{n=1}^{N} \nabla_{w^t} \mathcal{L}_n(\theta^t, w^t)$
  - Maximize → $\theta^{t+1} = arg \max_{\theta^t} \mathcal{L}(w^{t+1}, \theta^t)$

$$\theta^{t+1} = \theta^t + \epsilon \nabla_{\theta^t} \mathcal{L}_b \qquad \mathcal{L}_b = \frac{1}{N} \sum_{n=1}^{N} \mathcal{L}_n(w^{t+1}, \theta^t)$$

$$\approx \theta^t + \epsilon \frac{1}{M} \sum_{m=1}^{M} \mathcal{L}_b \nabla_{\theta^t} \log p_m$$

REINFORCE algorithm

1. Self-supervised Learning of Adversarial Examples

- Generalizability Comparisons
  - Dataset
  - Compression

Method	DF			F2F			FS			NT			Avg.
	DFDC	CelebDF	DF1.0	DFDC	CelebDF	DF1.0	DFDC	CelebDF	DF1.0	DFDC	CelebDF	DF1.0	
Xception [41]	0.654	0.681	0.617	0.708	0.598	0.745	0.708	0.601	0.605	0.646	0.625	0.838	0.669
Face X-ray [24]	0.609	0.554	0.668	0.633	0.684	0.766	0.646	0.697	**0.795**	0.613	0.703	0.866	0.686
F3Net [39]	0.682	0.664	0.658	0.679	0.654	0.761	0.679	0.636	0.651	0.672	0.689	0.932	0.696
RFM [47]	0.758	0.723	0.717	0.736	0.663	0.732	0.714	0.591	0.714	0.726	0.600	0.846	0.710
SRM [30]	0.679	0.650	0.720	0.687	0.693	0.775	0.671	0.643	0.771	0.656	0.651	**0.936**	0.711
Ours	**0.772**	**0.730**	**0.742**	**0.787**	**0.781**	**0.786**	**0.742**	**0.800**	0.695	**0.741**	**0.759**	0.889	**0.769**

Table 1. Generalizability comparisons with state-of-the-art methods in the term of AUC. The best results are in bold. The first row denotes the training data, and the second row shows the corresponding test dataset. Our method performs favorably among the models compared.

Training set	Method	Test set			
		LQ		HQ	
		DF	FS	DF	FS
NT	Xception [41]	0.587	0.517	0.770	0.718
	Face X-ray [24]	0.571	0.510	0.585	0.779
	F3Net [39]	0.583	0.519	0.805	0.612
	RFM [47]	0.558	0.516	0.798	0.639
	SRM [30]	0.555	0.529	0.838	**0.795**
	Ours	**0.628**	**0.568**	**0.846**	0.721

Table 2. Generalizability comparisons across different compression levels in the term of AUC. Our method achieves comparable performance against existing methods.

1. Self-supervised Learning of Adversarial Examples

- State-of-the-art Comparisons
  - Multi-task →classify, localization
  - State-of-the-art detector

Training set	Method	Test set	
		F2F	FS
F2F	LAE [14]	0.909	0.632
	ClassNSeg [34]	0.928	0.541
	Forensic-Trans [9]	0.945	0.726
	Ours	**0.960**	**0.848**

Table 3. Comparisons with models adopt multi-task learning in the term of ACC. Our model performs favorably against these arts.

Method	FF++	CelebDF
Two-stream [59]	0.701	0.538
Meso4 [1]	0.847	0.548
MesoInception4 [1]	0.830	0.536
FWA [27]	0.801	0.569
DSP-FWA [27]	0.930	0.646
Xception [41]	0.997	0.653
VA-MLP [33]	0.664	0.550
Headpose [53]	0.473	0.546
Capsule [35]	0.966	0.575
SMIL [25]	0.968	0.563
Two-branch [32]	0.932	0.734
SPSL [29]	0.969	0.724
MADD [57]	**0.998**	0.674
Ours	0.984	**0.797**

Table 4. Extensive evaluations with other state-of-the-art methods in the term of AUC. The models are trained on FF++ dataset. Our method performs favorbly when tested on FF++, and it outperforms others when tested on CelebDF.

1. Self-supervised Learning of Adversarial Examples

- Ablation Study
  - Adversarial learning (adv, ran)
  - Data augmentations (ops, aug)

Method	DF			F2F			FS			NT			Avg.
	DFDC	CelebDF	DF1.0	DFDC	CelebDF	DF1.0	DFDC	CelebDF	DF1.0	DFDC	CelebDF	DF1.0	
Xception [41]	0.654	0.681	0.617	0.708	0.598	0.745	0.708	0.601	0.605	0.646	0.625	0.838	0.669
Xception w/ adv	0.717	0.703	0.674	0.739	0.778	0.735	0.737	0.644	0.602	0.662	0.722	0.794	0.709
Ours w/ ran	0.763	0.663	0.690	0.763	0.745	0.696	0.738	0.700	0.650	0.705	0.666	0.810	0.716
Ours	**0.772**	**0.730**	**0.742**	**0.787**	**0.781**	**0.786**	**0.742**	**0.800**	**0.695**	**0.741**	**0.759**	**0.889**	**0.769**

Table 5. Ablation studies regarding the effectiveness of the adversarial training. The metric is AUC. Please see Sec. 4.4 for detailed experiment settings.

Method	DF			F2F			FS			NT			Avg.
	DFDC	CelebDF	DF1.0	DFDC	CelebDF	DF1.0	DFDC	CelebDF	DF1.0	DFDC	CelebDF	DF1.0	
Ours w/ ran ops	0.735	0.666	**0.782**	0.694	0.647	0.827	0.737	0.601	0.646	0.691	0.685	0.812	0.710
Ours w/ ops [55]	0.721	0.726	0.777	0.686	0.647	**0.840**	0.737	0.534	**0.720**	0.678	0.642	0.829	0.711
Ours w/ aug [24]	0.722	0.692	0.712	0.722	0.710	0.739	0.719	0.726	0.640	0.714	0.669	0.841	0.717
Ours w/ aug [58]	0.754	0.679	0.687	0.746	0.604	0.753	0.726	0.697	0.674	**0.770**	0.713	0.863	0.722
Ours	**0.772**	**0.730**	0.742	**0.787**	**0.781**	0.786	**0.742**	**0.800**	0.695	0.741	**0.759**	**0.889**	**0.769**

Table 6. Ablation studies regarding the effectiveness of the data augmentation strategies. The metric is AUC. Please see Sec. 4.4 for detailed experiment settings.

1. Self-supervised Learning of Adversarial Examples

- Ablation Study
  - Self-supervised tasks

			DF			F2F			FS			NT			Avg.
$\mathcal{L}_R$	$\mathcal{L}_T$	$\mathcal{L}_A$	DFDC	CelebDF	DF1.0	DFDC	CelebDF	DF1.0	DFDC	CelebDF	DF1.0	DFDC	CelebDF	DF1.0	
✓	✓	-	0.770	0.686	0.687	0.768	0.714	0.779	0.722	0.709	0.653	0.735	0.720	0.856	0.733
✓	-	✓	0.763	0.716	0.709	0.760	0.734	**0.800**	**0.776**	0.636	**0.707**	0.724	0.683	0.835	0.737
-	✓	✓	0.722	0.685	0.656	0.765	0.733	0.771	0.713	0.698	0.659	0.735	0.709	0.838	0.724
-	-	-	0.717	0.703	0.674	0.739	0.778	0.735	0.737	0.644	0.602	0.662	0.722	0.794	0.709
✓	✓	✓	**0.772**	**0.730**	**0.742**	**0.787**	**0.781**	0.786	0.742	**0.800**	0.695	**0.741**	**0.759**	**0.889**	**0.769**

Table 7. Effectiveness of the proposed self-supervised auxiliary tasks. The metric is AUC. We disable the self-supervised task by assigning a zero weight to the corresponding loss.

감사합니다