Supplementary Material for Leveraging Real Talking Faces via Self-Supervision for Robust Forgery Detection

Alexandros Haliassos[1,†] Rodrigo Mira[1] Stavros Petridis[1,2] Maja Pantic[1,2]

[1]Imperial College London [2]Meta AI

{alexandros.haliassos14,rs2517,stavros.petridis04,m.pantic}@imperial.ac.uk

Method	Accuracy (%)			AUC (%)		
	Raw	c23	c40	Raw	c23	c40
Xception [22]	99.0	97.0	89.0	99.8	99.3	92.0
CNN-aug [26]	98.7	96.9	81.9	99.8	99.1	86.9
Patch-based [3]	**99.3**	92.6	79.1	**99.9**	97.2	78.3
Two-branch [20]	—	—	—	—	99.1	91.1
Face X-ray [17]	99.1	78.4	34.2	99.8	97.8	77.3
CNN-GRU [23]	98.6	97.0	90.1	**99.9**	99.3	92.2
LipForensics [11]	98.9	98.8	94.2	**99.9**	99.7	98.1
FTCN [28]	—	**99.1**	—	—	**99.8**	98.3
RealForensics (ours)	**99.3**	**99.1**	**96.1**	**99.9**	**99.8**	**99.5**

Table 1. **In-distribution performance.** Accuracy and AUC scores on the test set of FaceForensics++ (FF++) after training on FF++. We repeat experiments for the dataset's three compression types: raw (no compression), c23 (mild compression), and c40 (strong compression. Best results are in **bold**.

1. More Experiments

1.1. In-distribution performance

Although our approach has been developed for cross-manipulation generalisation and robustness, for completeness we present results for in-distribution performance in Table 1. For each compression level (raw, c23, c40), we train on the training set and show results on the corresponding test set. We are on par with the state-of-the-art in the no/low compression regime, while outperforming the other methods on the more compressed data.

1.2. Generalisation to ForgeryNet

In Table 2, we provide results on generalisation performance to the newly-released ForgeryNet dataset [14]. We compare our model with the publicly-available LipForensics and FTCN models (all trained on FF++). RealForensics significantly outperforms both.

[†]Corresponding author.

	Ours	LipForensics [11]	FTCN [28]
ForgeryNet	**71.8**	66.7	57.3

Table 2. **Generalisation to ForgeryNet.** AUC scores (%) on the val set of ForgeryNet after training on FF++. Best results are in **bold**.

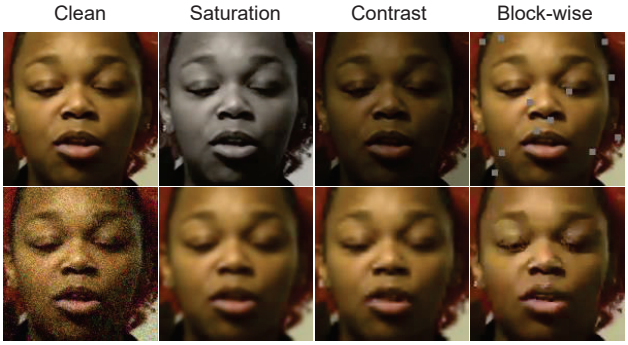

Figure 1. **Examples of corruptions.** A clean frame from a real FaceForensics++ video along with the same frame but corrupted with various perturbations. For more information on this set of corruptions, see [15].

1.3. Detailed analysis of robustness

Following [11], we present more detailed results on robustness by plotting AUC as a function of corruption severity (see Figure 2). On average, RealForensics deteriorates less abruptly as severity increases than other methods, with especially noteworthy results on video compression, which is ubiquitous on social media. We also highlight our significantly higher results over LipForensics on block-wise distortions (*i.e.*, occlusions), which are likely influenced by our method's use of the whole face rather than solely the mouth. For example, in some cases the mouth may be occluded while other parts of the face are not.

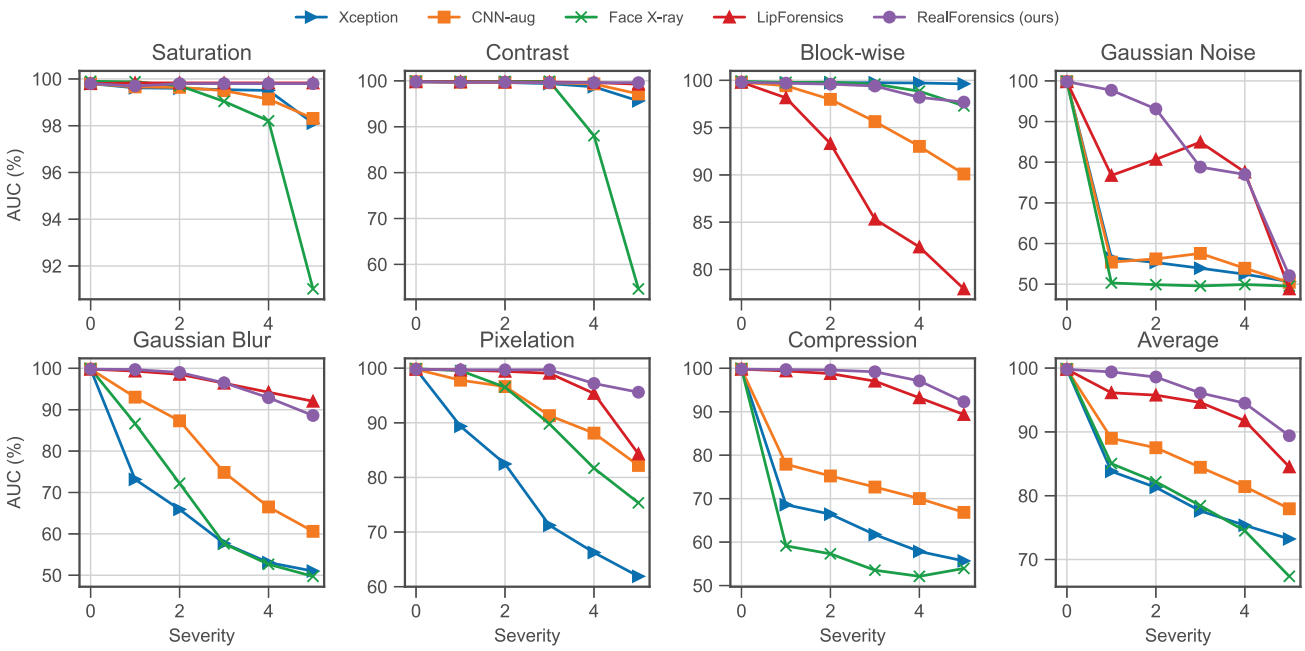

Figure 2. **Robustness to unseen perturbations.** AUC scores (%) on FaceForensics++ samples which have been corrupted by various unseen perturbations of varying severity. We also present the average scores across all perturbations. All methods were trained on FF++ without these corruptions. To avoid visual clutter in the plots, we show results for five representative methods. For more results, see [11] and [28].

1.4. More ablations

Full face versus mouth. In the main text, we argue that focusing only on the mouth region, like LipForensics [11], may be suboptimal for performance. We validate this by training (for both stages 1 and 2) on mouth crops and comparing the performance with the default setting. As shown in Table 4, our method consistently benefits from using the full face rather than the mouth, which was not observed for LipForensics [10]. This may be due to the cross-modal prediction task being more general than lipreading. For example, the video network is encouraged to retain information about the eyes to better model expression (which correlates with audio); on the other hand, a model trained to perform lipreading may focus predominantly on the mouth region.

Effect of clip size. Table 3 shows the effect on generalisation when changing the video clip size (default is 25 frames per clip). We observe that generalisation improves with clip size, up to a point.

Different backbone. Our default video backbone is a CSN network [24]. In Table 5 we also show generalisation results for ResNet+MS-TCN [19], used in [11]. We significantly outperform LipForensics with the same backbone and auxiliary dataset (compare with Table 3 in the main text), without requiring any auxiliary labels.

Projector and predictor. We propose in the main text to use a single linear layer as our projector and a shallow transformer as the predictor. In Table 6, we show generalisation results when using different types of projectors/predictors. Since we output dense representations, the linear layers in the MLPs can be thought of as convolutional layers with

Crop	Acc (%)		AUC (%)
	FSh	DFo	FS
Full face	**97.1**	**97.1**	**97.1**
Mouth	95.5	95.0	88.9

Table 3. **Full face versus mouth.** Accuracy and AUC scores when training on full faces and mouth crops. We test on FaceShifter (FSh) and DeeperForensics (DFo) after training on FaceForensis++ (FF++). We also test on FaceSwap (FS) after training on the remaining three FF++ types. Best results are in **bold**. Default setting is highlighted.

Clip size (# frames)	5	10	15	20	25	30
DeeperForensics	88.2	95.0	96.1	96.4	97.1	**97.4**
FaceShifter	87.9	93.4	95.4	95.7	**97.1**	96.7

Table 4. **Effect of clip size.** Accuracy (%) as a function of the clip size. We test on FaceShifter and DeeperForensics after training on FaceForensis++. Best results are in **bold**.

Backbone	FSh	DFo
CSN	**97.1**	**97.1**
ResNet+MS-TCN	94.0	95.7

Table 5. **Backbones.** Accuracy scores (%) on FaceShifter (FSh) and DeeperForensics (DFo) after training on FaceForensics++. We show results for two different backbones. Best results are in **bold**. Default setting is highlighted.

Settings		Accuracy (%)	
Projector	Predictor	FSh	DFo
Linear	MLP	91.8	92.9
Linear	Transformer	**97.1**	97.1
MLP	MLP	91.1	92.5
MLP	Transformer	96.1	**97.5**

Table 6. **Projector and predictor.** We test different types of projectors and predictors for the representation learning stage of our method (stage 1), and see how generalisation to FaceShifter (FSh) and DeeperForensics (DFo) is affected after training on FaceForensics++. Refer to subsection "Projector and predictor" for a discussion. Best results are in **bold**. Default setting is highlighted.

# blocks	FSh	DFo
1	**97.1**	**97.1**
2	96.8	96.4

Table 7. **Number of transformer blocks.** Accuracy scores (%) on FaceShifter (FSh) and DeeperForensics (DFo) after training on FaceForensics++. We show results for a 1-block and a 2-block transformer predictor. Best results are in **bold**. Default setting is highlighted.

kernel size 1. We use a learning rate of 3×10^{-4} when employing MLP predictors, as we found it to perform best in that setting.

Notably, we observe that using a transformer improves results over the MLP variant. This suggests that allowing the predictor to model temporal dynamics can benefit representation learning for our task. Further, in Table 7 we show results for a 1-block and a 2-block transformer predictor. We find that the 1-block variant performs slightly better.

Different contrastive baselines. As mentioned in the main text, self-supervised methods that aim to learn representations for lipreading tend to contrast samples from the same video to achieve invariance to identity [1, 5, 6]. Here, instead of our proposed non-contrastive approach, we apply the strategy of the audiovisual method Perfect Match [6] for stage 1 of our method. For fair comparison, we use the same backbones as for RealForensics. We follow the instructions from the paper for implementation. In particular, the inputs to the video and audio backbones are 5-frame video clips and 20-frame log mel spectrograms. Each network yields a single feature (via a temporal pooling layer). Then, for a single video feature, a contrastive loss is employed to match it to its aligned audio feature while repelling misaligned ones from the same video. We found that symmetrising this loss by additionally adding the loss corresponding to the reversal of the roles of the video and audio features yielded improvements; we refer to this variant as Perfect Match++. The results in Table 8 suggest that our proposed method, which does not target identity invariance, is better suited for forgery detection.

Method	FSh	DFo
PMatch	91.4	87.9
PMatch++	91.8	90.2
RealForensics (ours)	**97.1**	**97.1**

Table 8. **Different contrastive baselines.** Accuracy scores (%) on FaceShifter (FSh) and DeeperForensics (DFo) after training on FaceForensics++. We show results by employing the learning strategy of Perfect Match [6] (PMatch) for stage 1 of our method. We also use a symmetrised version of Perfect Match, which we call PMatch++. Best results are in **bold**.

Visual-only representation learning. Although it is natural to use the correspondence between the visual and auditory modalities to capture information related to facial behaviour and appearance, we present here some preliminary results on using only the visual modality in the representation learning stage. To this end, we extend BYOL to the video setting by using a single student-teacher pair. As is the case for the cross-modal task, the network outputs temporally dense representations, and we use a transformer for the predictor. We apply the augmentations proposed in [9] to each frame, consistently across the whole video. The results in Table 9 indicate that our proposed cross-modal task strongly benefits generalisation, likely because audiovisual correspondence provides a richer signal for encoding natural facial movements and expressions. We leave for future work the investigation of more effective video augmentations that could further improve the visual-only baseline.

Type	FSh	DFo
Visual	92.9	89.7
Audiovisual	**97.1**	**97.1**

Table 9. **Visual versus audiovisual representation learning.** Accuracy scores (%) on FaceShifter (FSh) and DeeperForensics (DFo) after training on FaceForensics++. We compare visual-only with audiovisual representation learning (using BYOL-style training) for stage 1 of our method. Best results are in **bold**. Default setting is highlighted.

2. Further Implementation Details

2.1. Preprocessing

We use RetinaFace [7][1] for face detection and a 2-D FAN network [2][2] to extract 68 facial landmarks. For each frame, we take the mean landmarks around a 12-frame window to reduce motion jitter and then affine warp to LRW's mean face based on eight stable points.

2.2. Dataset details

We provide further details on the used datasets. The licenses of all datasets permit their use for research purposes.

FaceForensics++ [22] (FF++). We use the dataset from the official webpage[3]. We use the provided train/validation/test splits, which include 720 training, 140 validation, and 140 test videos, respectively.

FaceShifter [16]. We use the dataset (at compression c23) from the FF++ webpage. Its real videos come from FF++. Note that we do not treat FaceShifter as part of FF++, consistent with the original paper [22].

DeeperForensics [15]. We use the dataset from the official webpage[4]. Its real videos also come from FF++ c23.

CelebFD-v2 [18]. We use the dataset from the official webpage[5].

DFDC [8]. We use a subset of the dataset from the official webpage[6]. This subset was used in [11] and features single-subject videos for which the face and landmark detectors did not fail (since many videos have been subjected to extreme perturbations).

[1] https://github.com/biubug6/Pytorch_Retinaface

[2] https://github.com/1adrianb/face-alignment

[3] https://github.com/ondyari/FaceForensics

[4] https://github.com/EndlessSora/DeeperForensics-1.0/tree/master/perturbation

[5] https://github.com/yuezunli/celeb-deepfakeforensics

[6] https://ai.facebook.com/datasets/dfdc

2.3. Architecture and training details

Supervised loss details. As described in Section 3.2 of the main text, we use a cosine classifier for our supervised head and also employ logit adjustment [21] to address data imbalance. Given the (average-pooled) output e of the backbone network and the weight vector w of the supervised head's linear layer, the normalised score of a sample's "fakeness" *during training* is given as

$$p = \frac{1}{1 + e^{-\left(s \frac{w \cdot e}{\|w\|_2 \|e\|_2} + \log \frac{\pi}{1-\pi}\right)}}, \tag{1}$$

where $s = 64$ scales the cosine similarity, as in e.g., [25], and π is the prior probability of a sample being fake, as described in [21]. We set π to be the ratio of fake samples to the batch size. We found using cosine similarity (*i.e.*, normalising the feature and weight vectors) yielded slight improvements; the ablation on logit adjustment is given in Table 5 of the main text. The supervised loss $\mathcal{L}_s(\mathcal{D}; \theta_b, \theta_s)$, introduced in Section 3.2 of the main text, is simply the standard binary cross entropy acting on these scores.

Random masking. We apply random erasing to video frames with probability 0.5, scale of $(0.02, 0.33)$, and ratio of $(0.3, 3.3)$. Moreover, we randomly erase a random number of video frames, ranging from 0 to 12, a random number of audio frames, ranging from 0 to 48, and a random number of mel filters, ranging from 0 to 27. This is applied with probability 0.5.

Backbones. Our video backbone is a modified Channel-Separated Convolutional Network (CSN) [24], chosen for its high accuracy in video action recognition [24] in conjunction with its relatively low parameter count. Unlike the original architecture, we set the temporal strides to 1 for all layers, thus preserving the temporal dimension. See Table 10 for more information.

Our audio backbone is a ResNet18 [13]. We modify the temporal strides to match the output size of the video backbone. In particular, the stem subsamples the temporal dimension by 4, after which no further temporal subsampling is performed. See Table 11 for more information.

Details on MLPs used in ablations. In the ablations where we use MLPs for the projector and/or predictor, we follow the design proposed in [4], as we found it to perform well. Thus, the projector MLP has 3 layers with hidden dimension 2048, and each layer is followed by batch normalisation (BN); the output layer has no ReLU activation. The predictor MLP has 2 layers with hidden dimension 512 and output dimension 2048, and the output layer has no BN nor ReLU.

Further details on contrastive experiments. We provide more details on the experiments with contrastive learning

stage	filters	output size
conv_1	$3 \times 7 \times 7$, stride $1 \times 2 \times 2$	$25 \times 56 \times 56$
pool_1	max, $1 \times 3 \times 3$, stride $1 \times 2 \times 2$	$25 \times 28 \times 28$
res_1	$\begin{bmatrix} 1 \times 1 \times 1, 256 \\ 3 \times 3 \times 3, 64 \\ 1 \times 1 \times 1, 256 \end{bmatrix} \times 3$	$25 \times 28 \times 28$
res_2	$\begin{bmatrix} 1 \times 1 \times 1, 512 \\ 3 \times 3 \times 3, 128 \\ 1 \times 1 \times 1, 512 \end{bmatrix} \times 4$	$25 \times 14 \times 14$
res_3	$\begin{bmatrix} 1 \times 1 \times 1, 1024 \\ 3 \times 3 \times 3, 256 \\ 1 \times 1 \times 1, 1024 \end{bmatrix} \times 23$	$25 \times 7 \times 7$
res_4	$\begin{bmatrix} 1 \times 1 \times 1, 2048 \\ 3 \times 3 \times 3, 512 \\ 1 \times 1 \times 1, 2048 \end{bmatrix} \times 3$	$25 \times 4 \times 4$
pool_2	global spatial average pool	$25 \times 1 \times 1$

Table 10. **Video backbone architecture.** The architecture of the modified CSN [24] network that we employ for the video backbone. The layers in the bottleneck blocks, shown in brackets, use *depthwise convolutions*. Next to the brackets we give the number of times the blocks are repeated in each stage. The output size is of the form $T \times H \times W$, where T denotes time, H height, and W width. Note that differently from the original architecture [24], we do not subsample the temporal dimension at any stage and also only use spatial pooling at the end, rather than spatio-temporal, since we employ dense learning.

given in Table 6 of the main text. For dense representation learning, the output of the network consists of 25 embeddings (one for each video frame); we select a random embedding to add to the queue of negative samples. We also use shuffling batch normalisation to prevent the network from cheating on the pretext task [12].

3. Visualisation

We use occlusion sensitivity analysis [27] for visualisation, as in [11]. We systematically occlude, in a sliding-window fashion, parts of the video via random erasing of size $40 \times 40 \times T$ (where T is the number of frames). We record for each occluded pixel the effect that the occlusion has on the model predictions. A heatmap is produced by averaging the output probabilities for each pixel. After normalisation, we overlay the heatmap on the first video frame. We show examples for FaceForensics++ in Figure 3. We see that for NeuralTextures and Face2Face (first two examples), which modify expressions, our network usually focuses on the mouth region. On the face-swapping types, we observe that sometimes the network focuses on the mouth and sometimes on other facial regions.

stage	filters	output size
conv_1	7×7, stride 2×2	50×40
pool_1	max, 3×3, stride 2×2	25×20
res_1	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2$	25×20
res_2	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$	25×10
res_3	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 2$	25×5
res_4	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 2$	25×3
pool_2	global frequency average pool	25×1

Table 11. **Audio backbone architecture.** The architecture of our modified ResNet18 [13] network that we employ for the audio backbone. The layers in a residual blocks are in brackets, next to which we give the number of times the blocks are repeated in each stage. The output size is of the form $T \times F$, where T denotes time and F mel filters. Note that differently from the original architecture [13], we do not subsample the temporal dimension at any stage and also only use mel frequency pooling at the end, since we employ dense learning.

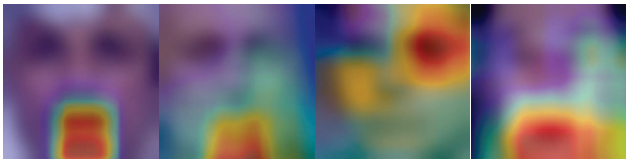

Figure 3. **Occlusion sensitivity analysis.** Occlusion sensitivity examples for FaceForensics++ types. The faces have been blurred to preserve anonymity.

References

[1] Triantafyllos Afouras, Andrew Owens, Joon Son Chung, and Andrew Zisserman. Self-supervised learning of audio-visual objects from video. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVIII 16*, pages 208–224. Springer, 2020. 3

[2] Adrian Bulat and Georgios Tzimiropoulos. How far are we from solving the 2d & 3d face alignment problem? (and a dataset of 230,000 3d facial landmarks). In *International Conference on Computer Vision*, 2017. 4

[3] Lucy Chai, David Bau, Ser-Nam Lim, and Phillip Isola. What makes fake images detectable? understanding properties that generalize. In *European Conference on Computer Vision*, pages 103–120. Springer, 2020. 1

[4] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15750–15758, 2021. 4

[5] Joon Son Chung and Andrew Zisserman. Out of time: automated lip sync in the wild. In *Asian conference on computer vision*, pages 251–263. Springer, 2016. 3

[6] Soo-Whan Chung, Joon Son Chung, and Hong-Goo Kang. Perfect match: Improved cross-modal embeddings for audio-visual synchronisation. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3965–3969. IEEE, 2019. 3

[7] Jiankang Deng, Jia Guo, Evangelos Ververas, Irene Kotsia, and Stefanos Zafeiriou. Retinaface: Single-shot multi-level face localisation in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5203–5212, 2020. 4

[8] 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. 4

[9] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent: A new approach to self-supervised learning. *arXiv preprint arXiv:2006.07733*, 2020. 3

[10] Alexandros Haliassos, Konstantinos Vougioukas, Stavros Petridis, and Maja Pantic. Supplementary material for lips don't lie: A generalisable and robust approach to face forgery detection. 2

[11] Alexandros Haliassos, Konstantinos Vougioukas, Stavros Petridis, and Maja Pantic. Lips don't lie: A generalisable and robust approach to face forgery detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5039–5049, 2021. 1, 2, 4, 5

[12] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9729–9738, 2020. 5

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 4, 5

[14] Yinan He, Bei Gan, Siyu Chen, Yichun Zhou, Guojun Yin, Luchuan Song, Lu Sheng, Jing Shao, and Ziwei Liu. Forgerynet: A versatile benchmark for comprehensive forgery analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4360–4369, 2021. 1

[15] Liming Jiang, Ren Li, Wayne Wu, Chen Qian, and Chen Change Loy. Deeperforensics-1.0: A large-scale dataset for real-world face forgery detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2889–2898, 2020. 1, 4

[16] Lingzhi Li, Jianmin Bao, Hao Yang, Dong Chen, and Fang Wen. Advancing high fidelity identity swapping for forgery detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5074–5083, 2020. 4

[17] 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*, pages 5001–5010, 2020. 1

[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*, pages 3207–3216, 2020. 4

[19] Brais Martinez, Pingchuan Ma, Stavros Petridis, and Maja Pantic. Lipreading using temporal convolutional networks. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6319–6323. IEEE, 2020. 2

[20] Iacopo Masi, Aditya Killekar, Royston Marian Mascarenhas, Shenoy Pratik Gurudatt, and Wael AbdAlmageed. Two-branch recurrent network for isolating deepfakes in videos. In *European Conference on Computer Vision*, pages 667–684. Springer, 2020. 1

[21] Aditya Krishna Menon, Sadeep Jayasumana, Ankit Singh Rawat, Himanshu Jain, Andreas Veit, and Sanjiv Kumar. Long-tail learning via logit adjustment. *arXiv preprint arXiv:2007.07314*, 2020. 4

[22] 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*, pages 1–11, 2019. 1, 4

[23] Ekraam Sabir, Jiaxin Cheng, Ayush Jaiswal, Wael AbdAlmageed, Iacopo Masi, and Prem Natarajan. Recurrent convolutional strategies for face manipulation detection in videos. *Interfaces (GUI)*, 3(1), 2019. 1

[24] Du Tran, Heng Wang, Lorenzo Torresani, and Matt Feiszli. Video classification with channel-separated convolutional networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5552–5561, 2019. 2, 4, 5

[25] Feng Wang, Xiang Xiang, Jian Cheng, and Alan Loddon Yuille. Normface: L2 hypersphere embedding for face verification. In *Proceedings of the 25th ACM international conference on Multimedia*, pages 1041–1049, 2017. 4

[26] Sheng-Yu Wang, Oliver Wang, Richard Zhang, Andrew Owens, and Alexei A Efros. Cnn-generated images are surprisingly easy to spot... for now. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8695–8704, 2020. 1

[27] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014. 5

[28] Yinglin Zheng, Jianmin Bao, Dong Chen, Ming Zeng, and Fang Wen. Exploring temporal coherence for more general video face forgery detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15044–15054, 2021. 1, 2