Supplementary Material for
Lips Don't Lie: A Generalisable and Robust Approach to Face Forgery Detection

Alexandros Haliassos[1,†] Konstantinos Vougioukas[1] Stavros Petridis[1] Maja Pantic[1,2]
[1]Imperial College London [2]Facebook London
{alexandros.haliassos14,k.vougioukas,stavros.petridis04,m.pantic}@imperial.ac.uk

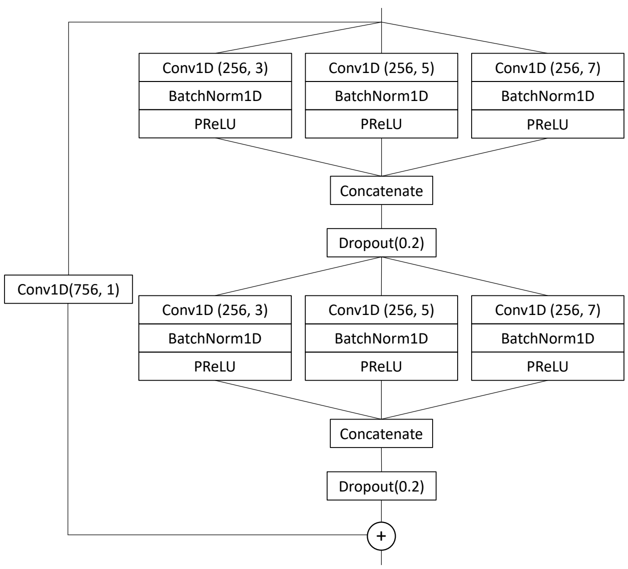

Figure 1. **Diagram of an MS-TCN block.** Diagram of a single block in the multi-scale temporal convolutional network (MS-TCN) that we employ. The stride is 1 for all convolutions. The full temporal network we use consists of four such blocks. The dilation rate in the first block is equal to 1, and each subsequent block's dilation rate is 2× the previous one's.

1. More Implementation Details

1.1. Architecture details

A single block of the multi-scale temporal convolutional network [18] (MS-TCN) used in our architecture is shown in Figure 1. The abbreviations are defined as follows:

- Conv1D(x, y): 1-D convolutional layer with x output channels and kernel size y. All use "same" padding and stride of 1.
- BatchNorm1D: 1-D batch normalisation [11] with momentum of 0.1.
- PReLU: Parametric ReLU activation [8] with a separate learnable parameter for each input channel.
- Dropout(x): Dropout layer [22] with probability x.

1.2. Datasets

FaceForensics++ (FF++) [20]. We download the dataset from the official webpage[1]. We use the provided training/validation/test splits.

FaceShifter [14]. We download the FaceShifter samples (at c23 compression) from the same place as FF++, since these have been recently added to the webpage. Note that when we refer to FF++, we are referring to the version described in the FF++ paper, *i.e.*, containing the 4 manipulation methods without FaceShifter. We use the same training/validation/test splits as in FF++.

DeeperForensics [12]. We download the dataset from the official webpage[2]. We use the same training/validation/test splits as in FF++.

Celeb-DF-v2 [17]. We download the dataset from the official webpage[3]. We use the test set, which consists of 518 videos.

DFDC [7]. We download the test set of the full DFDC dataset from the official webpage[4]. Some videos feature more than one person. To remove ambiguities in preprocessing, we only use single-person videos. Further, many videos have been filmed in extreme conditions (lighting, poses, etc) and/or have been post-processed with aggressive corruptions. As such, we only use videos for which the face and landmark detectors did not fail.

[†]Corresponding author.

All datasets were obtained by, and all training, testing, and ablation studies have been conducted at Imperial College London.

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

[2]https://github.com/EndlessSora/DeeperForensics-1.0

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

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

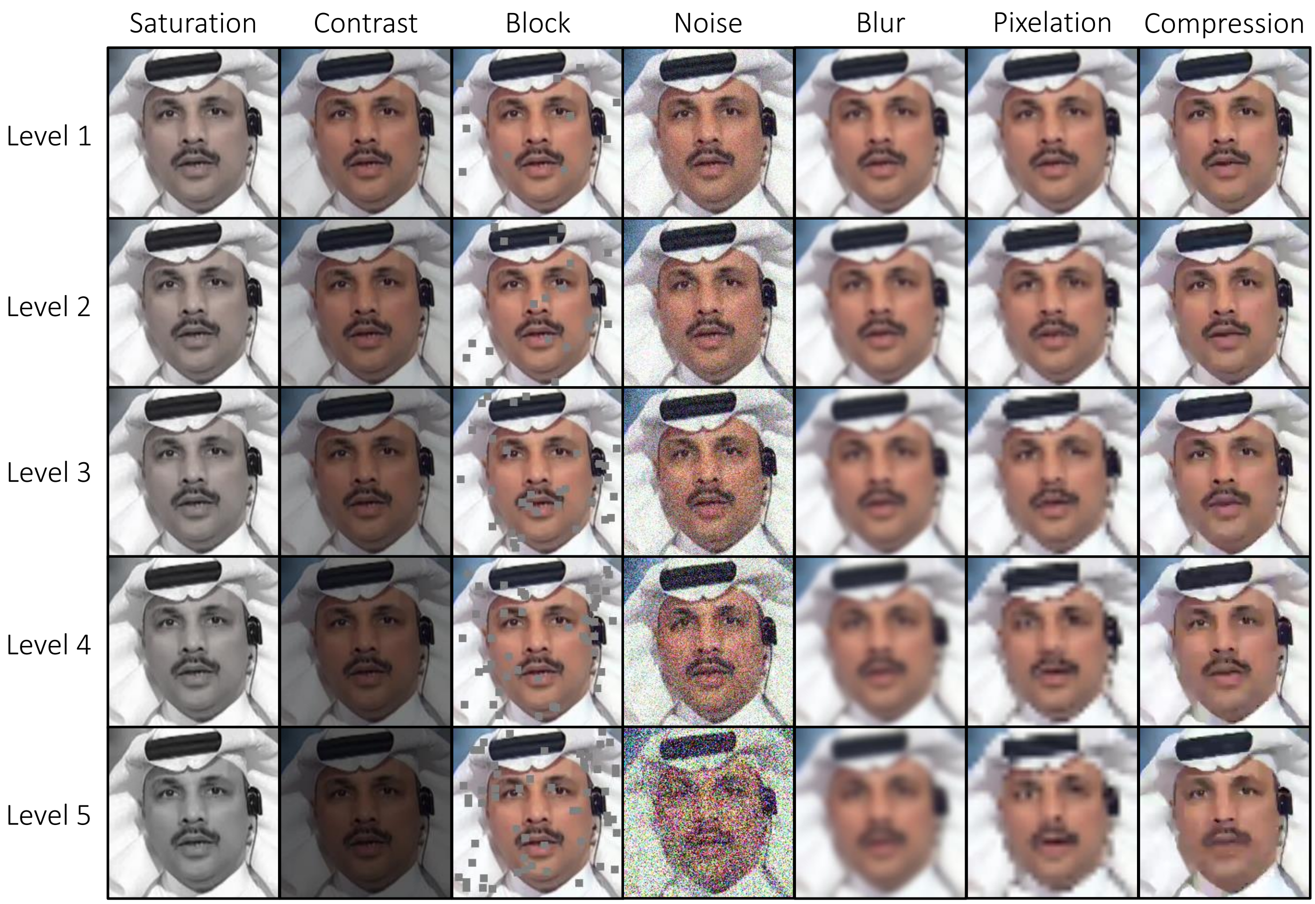

Figure 2. **All types of perturbations at all severity levels.** Visualisation of the seven perturbation types we used in our robustness experiments at all five of the severity levels. We note that in [12], "Pixelation" is named "JPEG compression," though the official code, at the time of writing, indeed performs pixelation (downscaling and upscaling).

1.3. Preprocessing

We use RetinaFace [5][5] to detect a face for each frame in the videos. As in [20], we only extract the largest face and use an enlarged crop, $1.3\times$ the tight crop produced by the face detector. To crop the mouths for LipForensics, we compute 68 facial landmarks using FAN [2][6]. The landmarks are smoothed over 12 frames to account for motion jitter, and each frame is affine warped to the mean face via five landmarks (around the eyes and nose). The mouth is cropped in each frame by resizing the image and then extracting a fixed 96×96 region centred around the mean mouth landmark. We note that alignment is performed to remove translation, scale, and rotation variations; it does not affect the way the mouth moves.

[5] https://github.com/biubug6/Pytorch_Retinaface
[6] https://github.com/1adrianb/face-alignment

1.4. Baselines

For the baselines we consider, we provide details on our implementations that are not given in the main text. Unless stated otherwise, Adam [13] optimisation is used with a learning rate of 2×10^{-4} and batch size of 32.

Face X-ray [15]. To generate the blended images for training, we use provided code[7]. In addition to the random mask deformation and colour correction operations described in the paper, the following augmentations are applied as per the code: random horizontal flipping, JPEG compression (with quality $\sim$ Uniform$\{30, 31, \dots, 100\}$), and pixelation (downscaling image by a factor $\sim$ Uniform$[0.2, 1]$), each with probability 0.5. For fair comparison with the other methods, we also train with samples from FaceForensics++ (FF++). Following the code, each image sampled during training is either a real FF++ frame or a fake sample, with probability 0.5. In turn, each fake sample is either a blended image or an FF++ fake frame, again with probability 0.5.

[7] https://github.com/AlgoHunt/Face-Xray

The cropped faces are resized to 317×317 and then centre cropped to 256×256. The scaling factor, λ, corresponding to the segmentation loss is set to 100, as in the paper.

CNN-aug [25]. We use the official code[8]. The cropped faces are resized to 256×256. We use JPEG compression (with quality $\sim$ Uniform$\{60, 61, \ldots, 100\}$) and Gaussian blurring with standard deviation $\sim$ Uniform$[0, 3]$, both with probability 0.1. We also use horizontal flipping with probability 0.5.

Patch-based [3]. We use the official code[9]. We train the model ourselves, since no provided pretrained model was trained on full FF++. The faces are aligned by affine warping them to the mean face and then resized to 299×299. We use horizontal flipping with probability 0.5. Adam [13] with learning rate 1×10^{-3} is used, as suggested in the paper.

Xception [20]. We use the official code[10]. The cropped faces are resized to 299×299. We use horizontal flipping with probability 0.5.

CNN-GRU [21]. The cropped faces are resized to 224×224. We use horizontal flipping with probability 0.5. As recommended in [21], we first train only the DenseNet-161 [10] (by adding a linear classifier). We then append a single-layer, bi-directional GRU [4] with hidden size 128 and train the whole network end-to-end.

Multi-task [19]. We use the official code[11] and follow the paper recommendations for all hyperparameters. We use the "deep" version of the model. We train it ourselves since the provided pretrained model has only been trained on a subset of FF++. The cropped faces are resized to 256×256. We use horizontal flipping with probability 0.5. Adam [13] with learning rate 1×10^{-3} is used, as suggested in the paper.

DSP-FWA [16]. We use the official code[12] and pretrained model (on self-collected real faces), which uses a dual spatial pyramid approach. Each face is aligned and extracted at 10 different scales. They are all resized to 224×224.

R(2+1)D-18 [24] and ip-CSN-152 [23]. We use the official code[13] and finetune pretrained models. We perform the same preprocessing as for our LipForensics approach, except that RGB frames are used rather than grayscale, since the pretrained tasks use colour frames.

[8]https://github.com/peterwang512/CNNDetection
[9]https://github.com/chail/patch-forensics
[10]https://github.com/ondyari/FaceForensics
[11]https://github.com/nii-yamagishilab/ClassNSeg
[12]https://github.com/yuezunli/DSP-FWA
[13]https://github.com/facebookresearch/VMZ

Input type	Pretrain	Finetune	FSh	DFo
Full face	none	whole	68.2	67.1
Full face	LRW	whole	82.9	85.2
Full face	LRW	temporal	84.3	90.0
Mouth	none	whole	62.5	61.4
Mouth	LRW	whole	83.2	84.6
Mouth	LRW	temporal	**87.5**	**90.4**

Table 1. **Full face crops versus mouth crops.** Effect of training on tight full face crops compared with training on mouth crops. We report video-level accuracy (%) scores on FaceShifter (FSh) and DeeperForensics (DFo) when trained on FaceForensics++.

SE-ResNet50 [9]. We use the ArcFace [6] code[14] and finetune the backbone of the model pretrained on face recognition datasets. The cropped faces are resized to 112×112, since this is the size used during pretraining. We use horizontal flipping with probability 0.5.

1.5. Robustness experiments

To apply the corruptions in our robustness experiments, we use the DeeperForensics code[15]. All considered corruptions at all severity levels are depicted in Figure 2.

2. Full Face Versus Mouth Crops

In the main text, we always use mouth crops for LipForensics. Here, we increase the crop from 88×88 to 112×112 (after random cropping) to also include the whole nose and eyes in the input. We pretrain a new model on LRW using this input. As shown in Table 1, when training from scratch, using full faces rather than mouth crops yields better generalisation to FaceShifter and DeeperForensics, but when using lipreading pretraining, mouth crops perform better. For both types of input, lipreading pretraining improves accuracy significantly.

3. Qualitative Analysis

3.1. High-level mouth inconsistencies

Our approach targets high-level temporal inconsistencies related to the mouth region. We show examples of such anomalies in Figure 3. Notice that in some cases, the mouth does not sufficiently close, as noted in [1]. In other cases, subtle temporal inconsistencies in the shape of the mouth or its interior (*e.g.*, teeth) are present.

3.2. Failure cases

Examples of failure cases are given in Figure 4. In general, we noticed that many of the failure cases involve rapid

[14]https://github.com/TreB1eN/InsightFace_Pytorch
[15]https://github.com/EndlessSora/DeeperForensics-1.0/tree/master/perturbation

head movements, poses that are uncommon in the training set (FF++), or very limited mouth movements.

3.3. Occlusion sensitivity

We show more visualisation examples using the occlusion sensitivity approach discussed in the main text. This approach was introduced in [26]. It relies on systematically covering up different portions of the frames with a grey block and measuring the effect on the predictions of the model. We found that a block size of $40 \times 40 \times t$, where t is the number of frames in the video, is suitable, as it is large enough to sufficiently occlude the mouth region. After each iteration, the block is displaced by one pixel, and the probability of predicting the correct class is recorded for each occluded pixel. Following this process, a heatmap can be created by averaging the probabilities at each pixel location. The heatmaps are finally normalised and overlaid on the first frame of the video.

We show visualisation examples for Xception [20] (see Figure 5) as well as for training the spatiotemporal network from scratch (see Figure 6) and LipForensics (see Figure 7). As mentioned in the main text, unlike Xception, LipForensics consistently relies on the mouth region. Interestingly, without lipreading pretraining, the network often seems to rely on regions other than the mouth (such as the nose), despite the (conservative) mouth crop. This is more the case for the face swapping methods, Deepfakes and FaceSwap.

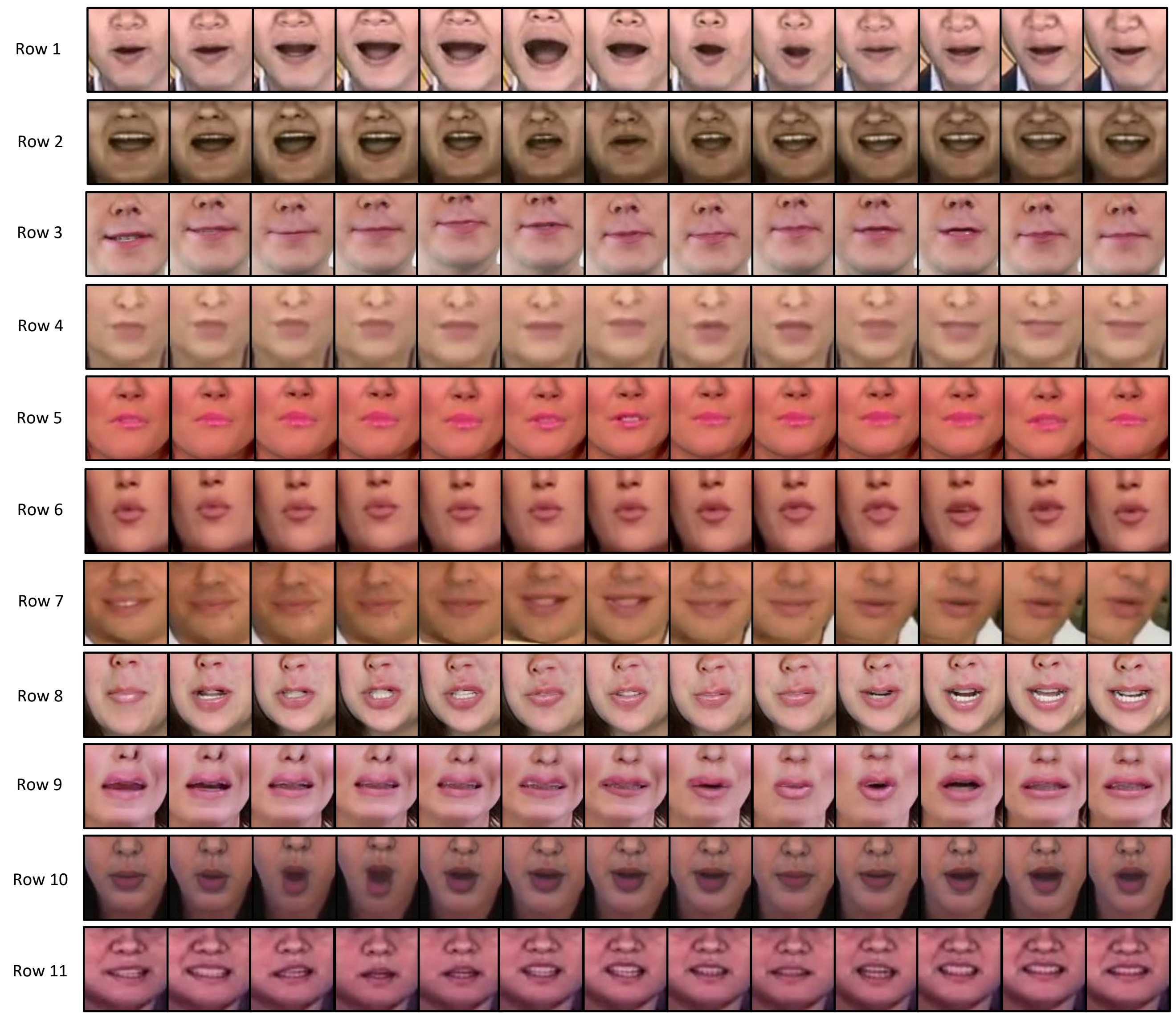

Figure 3. **Examples of semantically high-level inconsistencies around the mouth region.** Rows 1-2 show mouths that do not sufficiently close; rows 3-7 show mouths with limited mouth movements but which still exhibit anomalous behaviour; rows 8-9 show inconsistencies in the teeth and lip shape; rows 10-11 show temporal irregularities in mouth shape (*e.g.*, see frames 3 and 4 in row 10 and frame 3 in row 11). Subtle anomalies are more readily observed in video form.

Figure 4. **Failure cases.** Top two rows are real videos predicted as fake and bottom two are fake videos predicted as real.

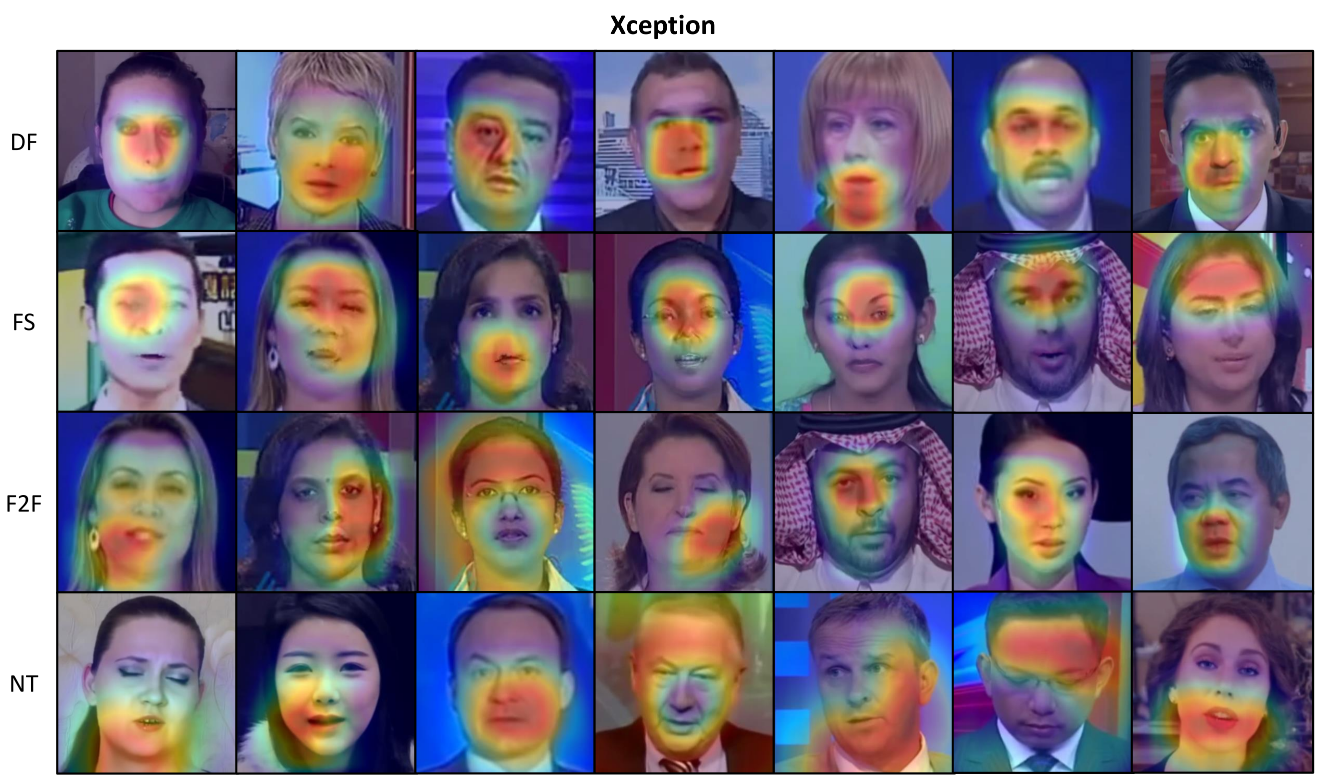

Figure 5. **Visualisation examples for Xception.** We show examples for Deepfakes (DF), FaceSwap (FS), Face2Face (F2F), and Neural-Textures (NT).

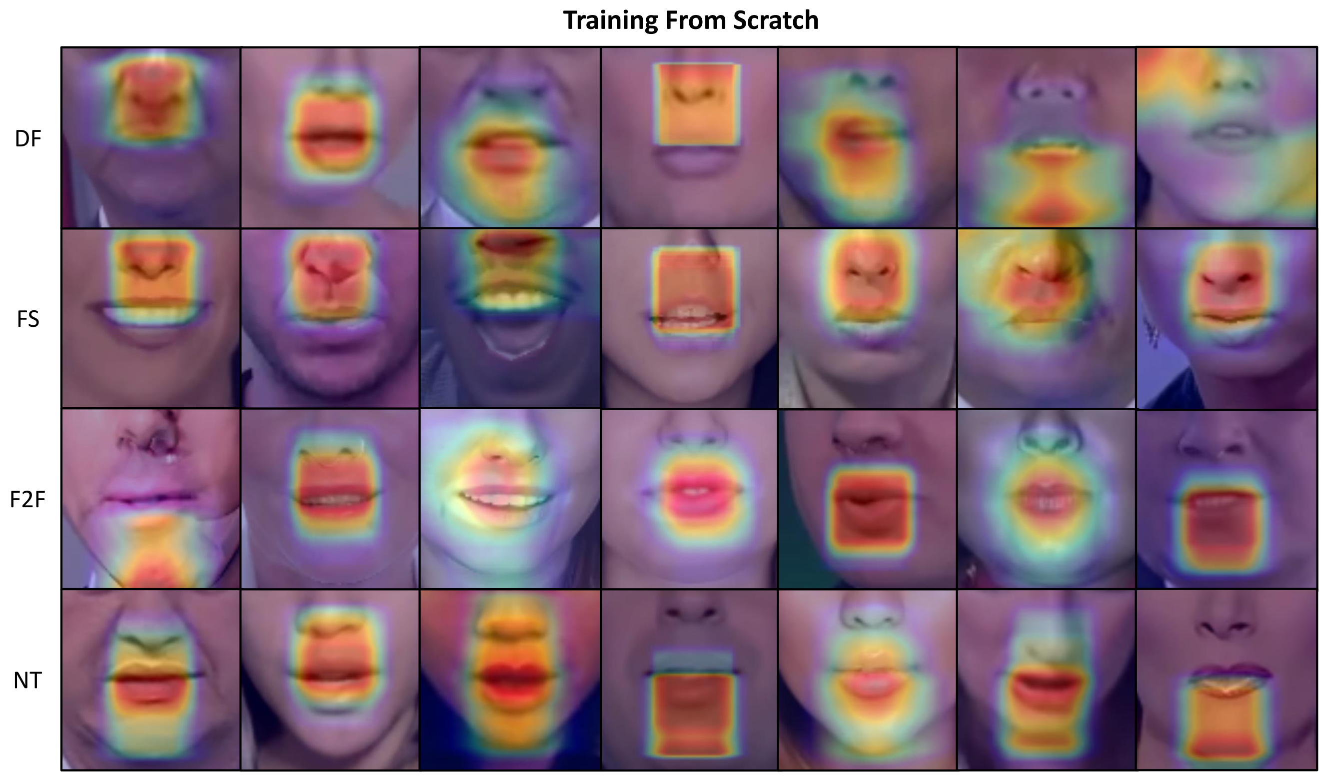

Figure 6. **Visualisation examples for spatiotemporal network without lipreading pretraining.** We show examples for Deepfakes (DF), FaceSwap (FS), Face2Face (F2F), and NeuralTextures (NT).

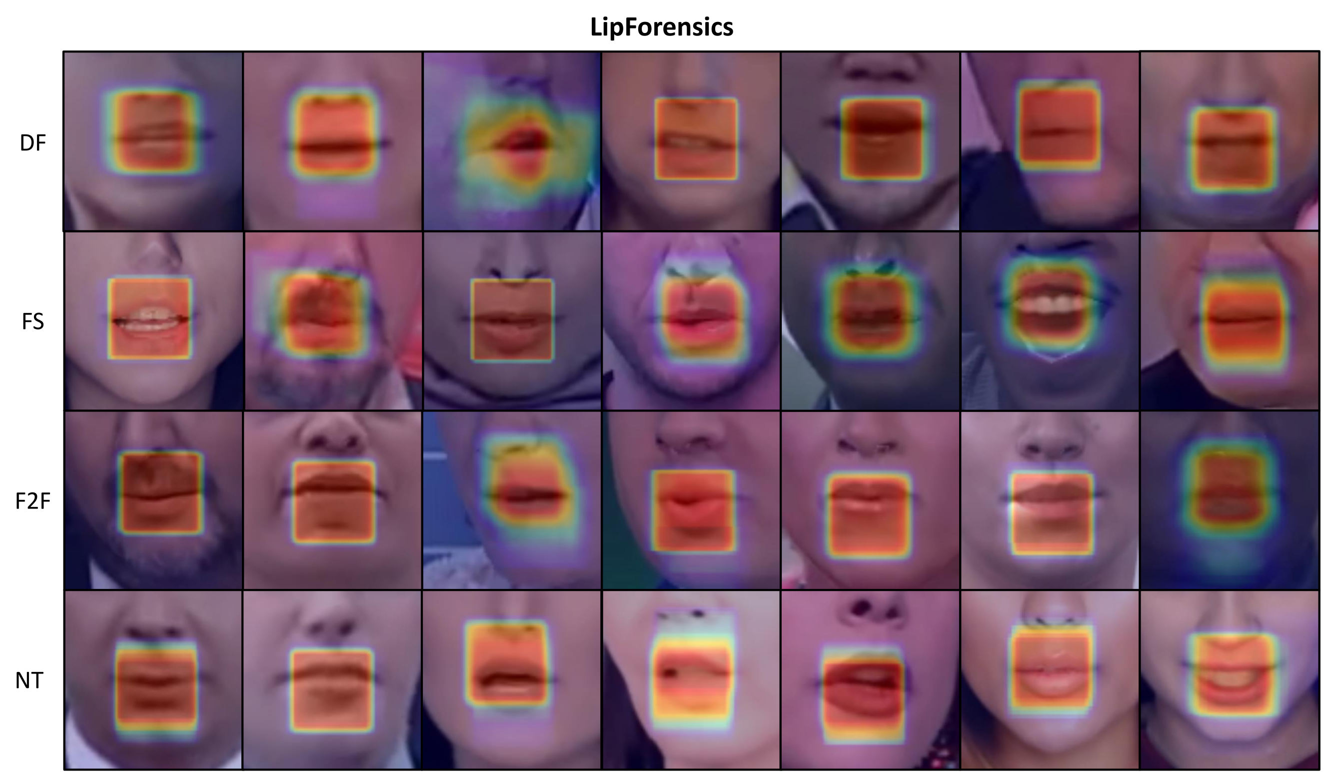

Figure 7. **Visualisation examples for LipForensics.** We show examples for Deepfakes (DF), FaceSwap (FS), Face2Face (F2F), and NeuralTextures (NT).

References

[1] Shruti Agarwal, Hany Farid, Ohad Fried, and Maneesh Agrawala. Detecting deep-fake videos from phoneme-viseme mismatches. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 660–661, 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 *Proceedings of the IEEE International Conference on Computer Vision*, pages 1021–1030, 2017. 2

[3] Lucy Chai, David Bau, Ser-Nam Lim, and Phillip Isola. What makes fake images detectable? understanding properties that generalize. *arXiv preprint arXiv:2008.10588*, 2020. 3

[4] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014. 3

[5] 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. 2

[6] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4690–4699, 2019. 3

[7] Brian Dolhansky, Joanna Bitton, Ben Pflaum, Jikuo Lu, Russ Howes, Menglin Wang, and Cristian Canton Ferrer. The deepfake detection challenge dataset. *arXiv preprint arXiv:2006.07397*, 2020. 1

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015. 1

[9] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018. 3

[10] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017. 3

[11] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015. 1

[12] 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 *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2886–2895. IEEE, 2020. 1, 2

[13] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 2, 3

[14] 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. 1

[15] 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. 2

[16] Yuezun Li and Siwei Lyu. Exposing deepfake videos by detecting face warping artifacts. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2019. 3

[17] 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. 1

[18] 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. 1

[19] H. H. Nguyen, F. Fang, J. Yamagishi, and I. Echizen. Multi-task learning for detecting and segmenting manipulated facial images and videos. In *2019 IEEE 10th International Conference on Biometrics Theory, Applications and Systems (BTAS)*, pages 1–8, 2019. 3

[20] 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 International Conference on Computer Vision*, pages 1–11, 2019. 1, 2, 3, 4

[21] 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. 3

[22] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014. 1

[23] Du Tran, Heng Wang, Lorenzo Torresani, and Matt Feiszli. Video classification with channel-separated convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5552–5561, 2019. 3

[24] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 6450–6459, 2018. 3

[25] 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 Conference on Computer Vision and Pattern Recognition*, volume 7, 2020. 3

[26] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014. 4