ForgeryNet: A Versatile Benchmark for Comprehensive Forgery Analysis (Supplementary Material)

Yinan He[1,2*] Bei Gan[1,3*] Siyu Chen[1,3*] Yichun Zhou[1,4*]
Guojun Yin[1,3] Luchuan Song[5†] Lu Sheng[4] Jing Shao[1,3‡] Ziwei Liu[6]
[1]SenseTime Research [2]Beijing University of Posts and Telecommunications
[3]Shanghai AI Laboratory [4]College of Software, Beihang University
[5]University of Science and Technology of China [6]S-Lab, Nanyang Technological University
{heyinan, ganbei, chensiyu, yinguojun, shaojing}@sensetime.com
{buaazyc, lsheng}@buaa.edu.cn slc0826@mail.ustc.edu.cn ziwei.liu@ntu.edu.sg

This supplemental document reports details on our proposed ForgeryNet. Appendix A and B describe the collection and preprocessing of the original data. In Appendix C, we present the 15 forgery approaches of ForgeryNet. In Appendix D and E, we introduce the re-rendering process and perturbations we use to imitate challenges encountered in real-world scenarios. We list specifics of our annotations and the dataset split in Appendix F and G. Finally, we give full details on our two benchmarks in Appendix H and I respectively.

A. Original Data Collection

In contrast to previous facial forgery datasets [25, 40] which only involve original data taken from certain briefing scenarios or TV shows, we choose four face datasets [2, 8, 14, 35] as the original data with diversified face identities, angles, expressions, actions, *etc*., for the sake of building a wild and diverse forgery dataset.

(1) *CREMA-D* [2] is a dataset of $7,442$ video clips from 48 male and 43 female actors with a variety of ethnicities, ages ranging from 20 to 74, and six different emotions.

(2) *RAVDESS* [35] consists of $7,356$ files including both video footages and sound tracks from 24 professional actors with eight emotions, vocalizing two lexically-matched statements in a neutral North American accent.

(3) *VoxCeleb2* [8] is constructed with over one million YouTube videos with utterances of $6,112$ celebrities.

(4) *AVSpeech* [14] is a dataset of 290k YouTube video clips of $3 \sim 10$ seconds long. Note that the speakers talk with no audio background interference, *i.e*. the only audible sound in the soundtrack of a video belongs to a single visible and speaking person.

*Equal contribution.
†Work done during an internship at SenseTime Research.
‡Corresponding author.

B. Original Data Preprocessing

The selected in-the-wild videos vary in length (2 seconds $\sim$ 1 hour), FPS ($20 \sim 30$), semantic annotations, and number of faces appearing in one frame. For further manipulation, we preprocess the original data into a controllable source video set:

(1) *Video-Origin & Image-Origin:* Due to the large amount of videos in VoxCeleb2 and AVSpeech, we respectively pick $43,941$ and $43,584$ videos with length over 6 seconds. The videos are chosen randomly, yet in VoxCeleb2 we guarantee all $6,112$ identities are included in the selected video set. All the selected videos from these two datasets are then truncated into $6 \sim 10$ seconds to enrich length variations, while those from CREMA-D and RAVDESS are retained without cropping due to their short duration ($2 \sim 5$ seconds). The images of `image-origin` are extracted from the aforementioned `video-origin` set with 20 FPS.

(2) *Target Face:* We detect faces from images in `image-origin` by RetinaFace [10] for future manipulation. As shown in Fig. 2 in the main paper, in some scenarios, multiple faces co-occur in a single frame, such as "conversation between two or more people" or "crowd gathering". To determine the target face for forgery, we first use a simple IoU (Intersection-over-Union) based tracking to acquire face tubes each with faces of the same person identity. We select the face which appears most frequently in the video, *i.e*. has the longest face tube.

(3) *Attribute Prediction:* To manipulate facial attributes, the deep models require attribute labels as a conditional input. However, data in `video/image-origin` lack attribute labels due to limited annotations (*e.g*. only "emotions" and "age") of the original datasets. To this end, we predict the attribute labels with Slim-CNN [33, 42], a state-of-the-art

face attribute classification method.

C. Forgery Approach

To guarantee the diversity of forgery approaches in the proposed ForgeryNet, we introduce 15 face forgery approaches [4, 6, 11, 17, 26–29, 36, 37, 44], which are shown in the main paper. We conclude five architecture variants as, 1) *Encoder-Decoder* [1] is used to disentangle the identity from identity-agnostic attributes and then modify/swap the encodings of the target before passing them through the decoder. 2) *Vanilla GAN* [43] consists of a generator and a discriminator which work against each other. After training, the discriminator is discarded and the generator is used to generate content. 3) *Pix2Pix* [29] is a popular improvement on GANs which enables translations from one image domain to another. The generator is an encoder-decoder network with skip connections from encoder to decoder which enable the generator to produce high fidelity imagery by bypassing some compression layers when needed. In addition to the above three variants, which are the basic elements for generating a forgery image, some sequential and dynamic-length data (*e.g*. audio and video) are often handled by 4) *RNN/LSTM* [4], and 5) *Graphics Formation* [13]. The latter represents a simulation of the classical image formation process of computer graphics, that is, reconstructing a 3D face model with 3DMM parameters, which are the output of a classical analysis-by-synthesis algorithm, and then rendering the generated 3D face model into a 2D image.

D. Re-rendering Process

(1) For the *face mask* condition shown in Fig. 4 (e-1) in the main paper, we first align the landmarks of $\tilde{\mathbf{I}}_t^f$ and $\mathbf{I}_t^f$ to align their masks $\tilde{\mathbf{I}}_t^m$ and $\mathbf{I}_t^m$, and then calculate an optimal transformation to align $\tilde{\mathbf{I}}_t^f$ back to the $\mathbf{I}_t$. Color matching is then operated on the re-aligned face to make $\tilde{\mathbf{I}}_t^f$ more adaptable to $\mathbf{I}_t^f$[1]. The following step is blending, with the objective of making $\tilde{\mathbf{I}}_t^f$ seamlessly fit the target full image $\mathbf{I}_t$. We corrode and blur the smaller mask between $\tilde{\mathbf{I}}_t^m$ and $\mathbf{I}_t^m$, and perform the Poisson blending along the outer contour of $\tilde{\mathbf{I}}_t^f$ to get the full forgery image $\tilde{\mathbf{I}}_t$.
(2) For the *face bounding-box* condition, an easy way is to directly substitute the bounding-box in the original target image $\mathbf{I}_t^b$ with a forgery one $\tilde{\mathbf{I}}_t^b$, and simply perform the Poisson blending along the edge of the bounding-box as shown in Fig. 4 (e-2) in the main paper. However, some GAN-based approaches always induce some unexpected details outside the face region, especially some background clutters with jittery and blurred information. Meanwhile, some graphic-based approaches cannot infer the texture of non-face regions such as hair. To this end, we first calculate the convex hull of the face area through the face landmarks to obtain the face mask $\tilde{\mathbf{I}}_t^m$, and then turn to the re-rendering solution for the *face mask* condition described above, as is illustrated in Fig. 4 (e-3) in the main paper.

Each frame of a video is re-rendered through the aforementioned steps. However, the obtained re-rendered frame sequence often contains frequent jitters due to misalignment and forgery effect. To generate a realistic and smooth video, we apply slight motion blur as well as compression or super-resolution to the frame sequence.

[1] *Identity-remained* forgery do not have this step since it only changes local intrinsic or external attributes. Moreover, some editing even aims at altering colors such as lip or eye color.

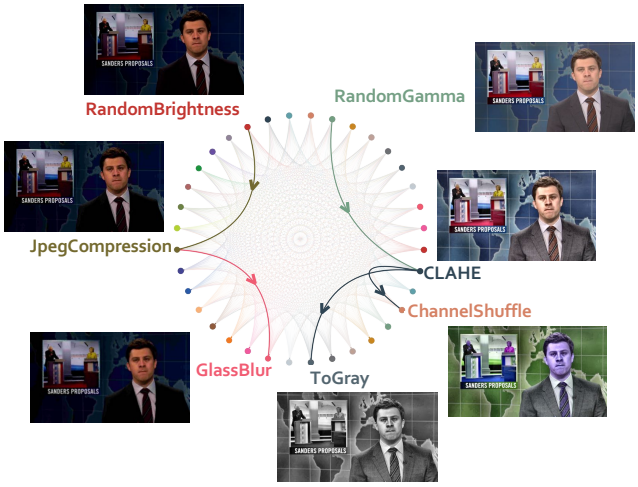

Figure 1: **Perturbations in ForgeryNet.** Different perturbations are marked in different colors. This example shows the effects of one or mixed perturbations. Arrows indicate the mixture order. The image on the left is first added "GlassBlur" followed by "JpegCompression" and at last "RandomBrightness".

E. Perturbation

Fig. 1 presents an overview of perturbations. For example, "GlassBlur" and "JpegCompression" can simulate distortion of information in video capture and storage in the real world. Some color distortions such as "RandomBrightness" and "ChannelShuffle" provide diversity in color distributions to adapt to different color renderings of a video.

Mixed perturbations with $2 \sim 4$ distortions are randomly applied to approximately 98% data, while another 1% are added with a single perturbation. The rest 1% are remained unchanged. Each perturbation has $1 \sim 5$ intensity levels. Types and levels of the applied perturbations are all chosen at random, and are applied at the video level, *i.e*. all frames of a video share the same type of perturbation with the same level. Meanwhile, to avoid severe distribution bias, we guarantee each pair of perturbation types co-occurs at

Table 1: **Summary of the four types of forgery approaches.** In this table, the input, output, architecture, resolution, modification ability, and whether to retrain in inference of each forgery approach are presented. S/T represents the modality of x_s and x_t. v:=video, i:=image, a:=audio, m:= mask, s:=sketch, l:= noise, S:=single identity, M:=multiple identities

	Method	S/T	CG/GAN	Input	Modification	Resolution	Retraining
Face Reenactment	FirstOrderMotion [44]	v/i	GAN	M/M	pose,expression	256*256	No need
	ATVG-Net [4]	v/i	GAN	M/M	pose,expression	128*128	No need
	Talking-head Video [17]	a/v	CG+GAN	M/S	mouth	256*256	1∼3 portraits
Face Editing	StarGAN2 [6]	i/i	GAN	M/M	attribute transfer	256*256	portraits
	StyleGAN2 [27]	l/i	GAN	M/M	rebuild from latent	1024*1024	portraits
	MaskGAN [28]	m,i/i	GAN	M/M	editing record	512*512	portraits,mask
	SC-FEGAN [26]	s,i/i	GAN	M/M	sketch record	512*512	portraits,sketch
	DiscoFaceGAN [11]	i/i	CG+GAN	M/M	3dmm attributes	1024*1024	portraits
Face Transfer	BlendFace	v/v	CG	M/M	identity, expression	Any	No need
	MMReplacement	i/i	CG	M/M	identity, expression	Any	at least 1 protrait
Face Swap	FSGAN [36]	v/v	GAN	M/M	identity	256*256	No need
	DeepFakes [37]	v/v	GAN	S/S	identity	192*192	2k∼5k portraits
	FaceShifter [29]	i/i	GAN	M/M	identity	256*256	No need

least once. The variety of perturbations improves the diversity and realness of ForgeryNet to better imitate the data distribution in real-world scenarios.

F. ForgeryNet Annotation

Image Forgery Classification. The annotations for this task have been elaborated in Sec. 3.3 in the main paper, where we introduce three types of forgery labels, *i.e*. labels for two-way (real / fake), three-way (real / fake with identity-replaced forgery approaches / fake with identity-remained forgery approaches), and n-way ($n = 16$, real and 15 respective forgery approaches) classification tasks respectively.

Spatial Forgery Localization. Due to the fact that forgery images contain various numbers of faces and each face can be manipulated completely or partially, it is more substantial to specify the manipulated area in addition to the classification labels. We convert the forgery image $\tilde{\mathbf{I}}_t$ and the corresponding real image $\mathbf{I}_t$ into two gray-scale images to calculate their pixel-by-pixel absolute differences. We then normalize the difference map within the face area of the real image $\mathbf{I}_t^f$ to obtain a *forgery distribution* $\tilde{\mathbf{I}}_t^d$. As shown in Fig. 5 (a) in the main paper, stronger response suggests the area is manipulated with heavier intensity. Note that we perform perturbations on the forgery image which cause further modifications in the whole image. The perturbed forgery area distributes all over the whole image rather than merely the face region. In the main paper, compared to Fig. 5 (b) which shows a near-uniform distribution of forgery area both inside and outside the faces, the distribution before perturbation in Fig. 5 (a) shows its advantages in two aspects: 1) the forgery area focuses more on face area, which is consistent with how these deep forgery techniques actually work, and 2) the forgery distribution behaves distinctive among different types of forgery approaches. Take *face reenactment* and *face transfer* as an example, the former has particularly high response on lip and also some medium response around head since the audio- or video-source always drives the lip and pose of the target being manipulated, while the latter replaces both identity-aware and identity-agnostic contents of the target and leads to more even response inside the face. In this paper, we define the *spatial forgery localization* task as "*localizing the face area manipulated by deep forgery approaches*", and thus the forgery distribution before perturbation $\tilde{\mathbf{I}}_t^d$ is taken as the ground-truth annotation.

Video Forgery Classification & Temporal Forgery Localization. As is mentioned in Sec. 3.3 in the main paper, in contrast to all existing datasets, we construct our video forgery dataset with untrimmed forgery videos $\tilde{\mathbf{V}}_t'$, each of which splices real and manipulated segments together. This is based on the consideration that forgery videos in the real world often only involve manipulation on a certain subject at some key frames. Specifically, for each pair of forgery video $\tilde{\mathbf{V}}_t$ and its corresponding real video $\mathbf{V}_t$, we first randomly select $1 \sim 4$ segments from the forgery video $\tilde{\mathbf{V}}_t$, and then fill the rest with the corresponding real segments $\mathbf{V}_t$. Each forgery/real segment in $\tilde{\mathbf{V}}_t'$ has no fewer than 9 frames.

Same as image-forgery, the *Video Forgery Classification* also contains three types of class annotations. We also provide the annotations of each fragment in the untrimmed forgery video and propose a new task, *i.e*. *Temporal Forgery Localization*, to localize the temporal segments which are manipulated.

G. ForgeryNet Split

We first split the identities of the original videos into two subsets, training and evaluation, roughly according to a proportion of 7:3. This guarantees that any person appearing in a training video does not show up in the evaluation set. Note that the AVSpeech dataset does not provide annotations on person identity, so we have to assume that different videos contain different people, and directly split the videos. The evaluation subset is then further divided into validation and test with an approximate ratio of 1:2, and there may be some identity overlaps between the validation and test subsets. The real data for our image set is sampled from the frames extracted with these original videos according to some fixed proportion. Finally, we apply our 15 forgery approaches to generate manipulated data within each subset respectively, *e.g.* the sources and targets for generating validation forgery data must all come from the validation subset of the original videos.

H. Image Forgery Analysis Benchmark

H.1. Metrics

Image Forgery Classification. We detail calculation methods of the metrics listed in Sec. 4.1.1 in the main paper. For k-way classification ($k = 2,\ 3,\ 16$), we use Accuracy (Acc) balanced over classes, *i.e.* we first calculate k accuracy values from the k classes respectively, and then take the uniform average of them as the final balanced accuracy. We also evaluate the standard Area under ROC curve (AUC) for binary classification. In terms of the other settings with more than two classes, we turn to mean Average Precision (mAP) to measure the discrimination ability of the forensics method. More specifically, the AP of some class i is simply the AUC calculated with class i as the sole positive class and all others being negative. After obtaining k APs, we average them to get mAP. Apart from Acc and mAP, we also compute binary metrics for 3-way or n-way classification, and we sum up probabilities predicted for all forgery categories as the final fake confidence.

Spatial Forgery Localization. As is mentioned in Sec. 4.1.2 in the main paper, we choose three metrics for evaluating predicted maps in our spatial localization task: two variants of Intersection over Union (IoU) and L1 distance. Let N denote the number of pixels, and τ be a predefined threshold.

- IoU $= \frac{1}{N}\sum_{i=1}^{N} |\mathbb{I}[\text{pred}_i \geq \tau] - \mathbb{I}[\text{gt}_i \geq \tau]|$ (*e.g.* $\tau = 0.1$) represents the accuracy over all spatial grids.

- $\text{IoU}_{\text{diff}} = \frac{1}{N}\sum_{i=1}^{N} \mathbb{I}[|\text{pred}_i - \text{gt}_i| \leq \tau]$ (e.g. $\tau = 0.05$) indicates whether the predicted value of each pixel is close to the groundtruth.

- L1 distance $\text{Loss}_{l1} = \frac{1}{N}\sum_{i=1}^{N} |\text{pred}_i - \text{gt}_i|$ also implies how close is the predicted map to the groundtruth one.

H.2. Models

Image Forgery Classification. There are in total 11 image-level classification methods.

- **MobileNetV3** [22] is an efficient mobile model, combining the following three layers: depthwise separable convolutions from MobileNetV1 [23], the linear bottleneck and inverted residual structure from MobileNetV2 [41], and lightweight attention modules based on squeeze and excitation from MnasNet [45]. We use both MobileNetV3-Small and MobileNetV3-Large for evaluation.

- **EfficientNet-B0** [46] is the baseline network of the EfficientNet family, which is developed by leveraging a multi-objective neural architecture search based on mobile inverted bottleneck MBConv [41] with squeeze-and-excitation optimization [24] added to it.

- **ResNet-18** [21] is the smallest ResNet architecture with 17 convolutional layers and one fully connected layer for final output.

- **Xception** [7] is a deep convolutional network architecture based on Inception replaced with depthwise separable convolutions. Xception is regarded as our default baseline in further experiments.

- **ResNeSt-101** [48] is a new variant of ResNet. It introduces a modular Split-Attention block that enables attention across different feature-map groups and stacks these blocks ResNet-style to get better performance with similar number of parameters.

- **SAN19-patchwise** [49] takes patchwise self-attention as the basic building block for image recognition. Specifically, we uses SAN19 which roughly corresponds to ResNet-50 to evaluate.

- **ELA-Xception** and **SNRFilters-Xception** differ from Xception in the fact that they do not directly take RGB images as input. More specifically, the input for ELA-Xception is the resulting difference image from Error Level Analysis (ELA) [20]. SNRFilters-Xception, as its name suggests, applies a set of 5×5 high pass kernels [5] to the original input image, and then concatenate the 4 output images along the channel dimension (the number of input channels of the first convolution in Xception is changed to 12 accordingly).

- **Gram-Net** designs Gram Block to leverage global image texture information for fake image detection. The

original paper [34] adds Gram Blocks to the ResNet architecture. Yet in our benchmark, we apply them to our baseline model Xception for the sake of fair comparison.

- **F^3-Net** [38] explores frequency information for face forgery detection by taking advantages of two frequency-aware clues: frequency-aware decomposed image components and local frequency statistics. Note that F^3-Net also uses Xception as the backbone network.

Spatial Forgery Localization. We select 3 representative models for spatial localization.

- **Xception+Regression** uses Xception as the backbone network, and adds an extra deconvolution layer after the final feature map to form a direct regression branch which outputs the spatial forgery map.
- **Xception+UNet** [39] supplements a usual contracting network by successive layers where pooling operations are replaced by upsampling operators. A successive convolutional layer can learn to assemble a precise output based on this information. For fair comparison, UNet also uses Xception as its encoder network.
- **HRNet** [47] starts from a high-resolution convolution stream, gradually adds high-to-low resolution convolution streams, and connects the multi-resolution streams in parallel. We use the HRNet-W48 instantiation.

H.3. Implementation Details

Training. For classification methods, we use the default cross-entropy loss for training. As for localization methods, we also add a segmentation loss in addition to the classification loss. There are two choices for the segmentation loss: (1) binary cross entropy loss with soft targets averaged over all spatial locations; (2) MSE loss with respect to groundtruth targets. We select one of these two losses for each localization model based on validation results.

All models use ImageNet [9] for pre-training. We train both classification and localization models end-to-end using synchronous SGD for optimization. The mini-batch size is set to 128. We adopt a multistep learning rate schedule with 100k iterations in total, and the learning rate is decreased by a factor of 0.5 at steps 20k, 40k, 60k, 70k, 80k and 90k. The base learning rate for each model is selected from the set $\{0.01, 0.02, 0.05\}$ according to validation performance. We use linear warm-up [19] from 0.01 during the first 1k iterations. The weight decay is set to 10^{-4} and we apply Nesterov momentum of 0.9. We use face images cropped with provided square bounding boxes (detected boxes enlarged $1.3\times$) for training. For data augmentation, we with 99% probability randomly select one perturbation from some set of perturbation methods, and apply it to the input image. Apart from random perturbation, for a model with input spatial size $S\times S$, we scale the side length to a random value in range $[S, 8S/7]$, and then randomly crop out a $S \times S$ region. Note that for five Xception-based classification models $S = 299$, for three localization models $S = 256$, and for the other six classification models $S = 224$. We also apply random horizontal flip before feeding the input to the model.

Inference. We only perform single-crop inference, and directly scale the input face image to the input spatial size $S \times S$ of the model.

Table 2: **Ablation study on augmentation (image).** We report accuracy and AUC scores of Protocol 1 binary classification on the validation set with three different levels of augmentation.

	weak aug		normal aug		enhanced aug	
	Acc.	AUC	Acc.	AUC	Acc.	AUC
Xception	66.73	74.75	73.70	82.56	80.78	90.12

H.4. More Experiments

Ablation Study on Augmentation. We experiment on three different levels of augmentation: weak, normal and enhanced. Weak augmentation does not add random perturbation mentioned in Appendix H.3, while normal and enhanced settings include different numbers of common perturbation methods in the perturbation set for augmentation. Results of Xception trained on these types of data augmentation are shown in Tab. 2. It can be seen that exerting appropriate augmentation to the training set significantly improves the performance of an image forgery classification model.

Cross-dataset Experiments. We provide cross-dataset testing results with our ForgeryNet (image forgery binary classfication only) as well as three public deepfake datasets - FF++ (c23) [40], DFDC [12], and DeeperForensics-1.0 (DF1.0) [25] which are only used for testing. For evaluation, we use (1) test set of FF++ (c23); (2) both validation and test set (only the released half) of DFDC; (3) a subset of DF1.0 which corresponds to the test set of FF++; (4) test set of our image benchmark. For video datasets, we extract frames with temporal stride 30 for frame-level testing. We present the numbers in Tab. 3. ForgeryNet shows the best cross-dataset performances on all other test sets, which indicates the strong generality of our dataset.

I. Video Forgery Analysis Benchmark

I.1. Metrics

Video Forgery Classification. The metrics for this task are the same as those for image classification.

Table 3: **Cross-dataset experiments.** We report frame-level AUC scores. Each row corresponds to a model trained with one of the datasets. Underlined values are results of models trained and tested on the same dataset, and the bold ones emphasize best cross-dataset performances.

	DF1.0	FF++	DFDC(val)	DFDC(test)	ForgeryNet
FF++ [40]	85.41	<u>99.43</u>	59.77	62.19	63.80
DFDC [12]	79.60	71.34	<u>90.12</u>	<u>93.50</u>	**68.93**
ForgeryNet	**90.09**	**85.06**	**69.68**	**71.08**	<u>90.09</u>

Temporal Forgery Localization. For the temporal localization task, the goal is to generate proposals which have high temporal overlap with the groundtruth (manipulated segments) as well as high recall. We give specifics on our employed metrics for evaluating predicted segments with respect to the groundtruth ones, which are Average Precision at some tIoU threshold (AP@t, *e.g.* $t = 0.5$), average AP, as well as Average Recall@K (AR@K, *e.g.* $K = 5$). Note that these metrics mostly reference ActivityNet [18] evaluation. In details, we choose 10 equally-spaced tIoU threshold values between 0.5 and 0.95 (inclusive) with a step size of 0.05. Under a certain tIoU threshold value t, we may match our predicted segments with the groundtruth according to the condition that tIoU $\geq t$. Recall@K with tIoU threshold t is defined as the proportion of groundtruth which can be matched with some prediction, after preserving only K predicted segments per video on average. AP@t, on the other hand, is the Area under ROC curve computed with predictions and their associated confidence scores, treating the predictions which are matched to some groundtruth segment with tIoU threshold t as positive. Finally, average AP and AR@K are simply the uniform average of APs and Recall@Ks computed at the 10 tIoU thresholds, respectively. Note that both real and fake videos are included in our evaluation, although the real ones do not contain any forgery segment (Recall is not be affected by real videos, but AP is).

I.2. Models

Video Forgery Classification. We choose four typical models for video classification.

- **TSM** [30] inserts Temporal Shift Modules to 2D CNNs to achieve temporal modeling at zero computation and zero parameters. We follow its default setting with ResNet-50 as the backbone network.
- **SlowFast** [16], featuring its two-pathway design with different input temporal strides, is one of the state-of-the-art architectures for action recognition. We choose its R-50 instantiation (without Non-Local blocks), and set the fast-to-slow ratio $\alpha = 4$.
- **Slow-only** is basically the slow pathway of SlowFast, and we also use the R-50 instantiation. Note that with the same number of input frames, Slow-only is actually heavier than SlowFast since the slow branch of the latter only use $1/\alpha$ of the frames.
- **X3D-M** [15] is one member of the X3D family, a series of efficient video networks obtained by progressive expansion along multiple axes. It is able to achieve performances nearly comparable with SlowFast R-50 on common video benchmarks while having much fewer parameters.

Temporal Forgery Localization. As described in Sec. 6.2 in the main paper, we include a frame-based method, where we use Xception as the frame prediction model. The logic of this method can be briefly stated as the following:

1. For a video with T frames, we run the Xception model to get frame-level scores, and then binarize them with threshold 0.25, acquiring a sequence of T binary predictions (real/fake).
2. We enumerate tolerance value in the set $\{1, 3, 5, 7\}$. For a tolerance value t, we inspect the sequence of T predictions, and selects manipulated segments with at least 5 frames satisfying that the length of consecutive real frames in the middle does not exceed t. The confidence score of a segment is simply the average of its frame-level scores.
3. We combine segments predicted with different tolerance levels, and remove duplicates to form the final predictions.

For two video-based methods (BSN [32] and BMN [31]), we use SlowFast and X3D-M for extracting clip features, forming four different "feature+method" pairs. Note that for these feature extraction models, we use fewer input frames for training than their classification counterparts to increase temporal locality. Accordingly, the fast-to-slow ratio α of SlowFast is decreased to 2.

I.3. Implementation Details

Training. For classification methods and feature extraction models for localization, we use the default cross-entropy loss for training. The frame-based localization method directly uses the Xception model trained with the image binary classification task, and does not need any extra training. BSN and BMN have their own training loss functions and procedures which we do not alter.

All models use Kinetics-400 [3] for pre-training. We train them end-to-end using synchronous SGD for optimization. The mini-batch size is set to 64. We adopt a multistep learning rate schedule with 50k iterations in total, and the learning rate is decreased by a factor of 0.5 at steps 20k, 30k, 40k and 45k. The base learning rate is

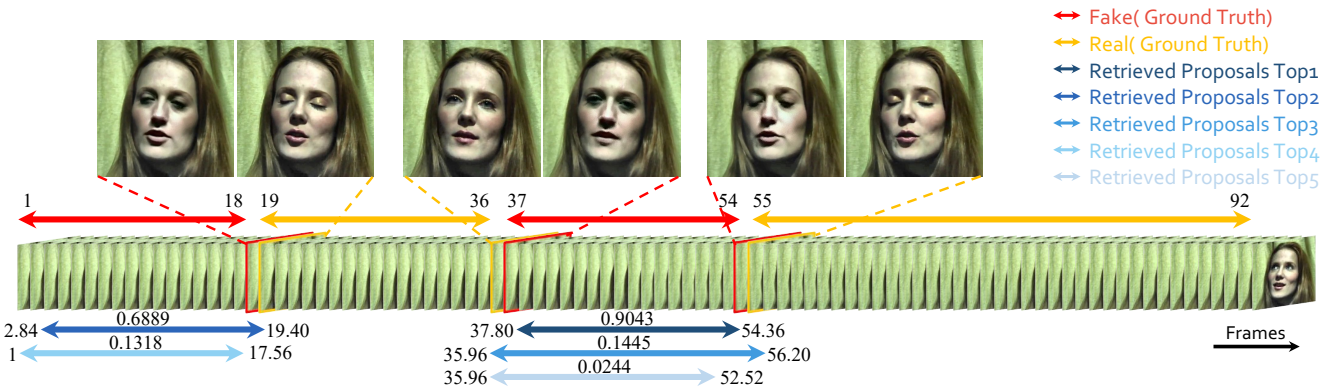

Figure 2: **Example of temporal forgery localization.** We show top-5 predictions of the model SlowFast+BMN. All endpoints of the two manipulated segments are localized with high precision.

set to 0.02. We use linear warm-up from 10^{-3} during the first 500 iterations. All classification models take 16 frames with a temporal stride of 4 as input, yet the feature extraction models (SlowFast and X3D-M) for BSN and BMN use only continuous 8 frames as input for better temporal sensitivity. We use temporal random crop for training, *i.e*. for a model requiring T frames $\times$ stride τ, we randomly sample a segment of length $T \times \tau$ from the video. In some rare cases where the entire video has less than $T \times \tau$ frames, we use loop padding to fill the rest. The input spatial size is fixed to $S = 224$. Other training details are the same as those for image experiments.

For BSN and BMN, since the feature extraction models take 8 frames as input, we extract features with stride 4. We set the temporal scale parameter to 50, and linearly interpolate the extracted features to the 51 endpoints. We only use fake videos for training video-based localization models. We train TEM and PEM in BSN for 20 epochs each. We train BMN for 9 or 18 epochs according to validation performance. The mini-batch size is set to 128. Other hyper-parameters follow the original settings of BSN and BMN.

Inference. We scale the input to $S \times S$ spatially. On the temporal dimension, we use two settings for classification inference (suppose input temporal sampling is $T \times \tau$): (1) single-crop, or to be more specific, temporally center crop $T \times \tau$ frames; (2) multi-crop, *i.e*. crop multiple segments of length $T \times \tau$ to cover the entire video.

For temporal localization, we only keep top 10 predictions per video in terms of confidence score, and for video-based methods, relevant hyper-parameters are the same as training.

I.4. More Experiments

Ablation Study on Augmentation. We conduct similar experiments on augmentation with the same settings as Appendix H.4. As presented in Tab. 4, we observe that our video-level forgery classification method is less affected by augmentation than its image-level counterpart.

Table 4: **Ablation study on augmentation (video).** We report accuracy and AUC scores of Protocol 1 binary classification on the validation set with three different levels of augmentation.

	weak aug		normal aug		enhanced aug	
	Acc.	AUC	Acc.	AUC	Acc.	AUC
SlowFast	84.39	91.61	87.75	93.22	88.78	93.88

Table 5: **Experiemnts on temporal shuffling.** We report accuracy and AUC scores of Protocol 1 binary classification on the validation set with three different levels of temporal shuffling.

	shuffle 16		shuffle 64		shuffle all	
	Acc.	AUC	Acc.	AUC	Acc.	AUC
SlowFast	88.63	94.11	86.24	93.00	85.04	91.74

Temporal Shuffling Experiments. To verify the effect of continuous temporal information for video forgery classification, we train the SlowFast model with different levels of temporal random shuffling to disrupt temporal continuity: shuffle every 16 frames, shuffle every 64 frames, and shuffle all frames. The results in Tab. 5 indicate that temporal disruptions have considerable, but not very major impact on the performance video classification, implying the video model may have leveraged other sources of information than the continuous temporal flow. An interesting finding is that a weak level of random shuffling (shuffle 16) even slightly boosts the AUC score compared to the setting without shuffling recorded in Tab. 4.

I.5. Temporal Localization Analysis

We present an example of temporal forgery localization in Fig. 2. This data sample demonstrates the ability of a

boundary-aware model to locate the transitions between real and fake. All endpoints are accurately pointed out by the BMN model. Note that there exist some highly similar predictions, yet are suppressed by a SoftNMS process.

References

[1] faceswap. https://github.com/deepfakes/faceswap, 2020.

[2] Houwei Cao, David G Cooper, Michael K Keutmann, Ruben C Gur, Ani Nenkova, and Ragini Verma. Crema-d: Crowd-sourced emotional multimodal actors dataset. *IEEE Trans. Affect. Comput.*, 5(4):377–390, 2014.

[3] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017.

[4] Lele Chen, Ross K Maddox, Zhiyao Duan, and Chenliang Xu. Hierarchical cross-modal talking face generation with dynamic pixel-wise loss. In *CVPR*, 2019.

[5] Mo Chen, Vahid Sedighi, Mehdi Boroumand, and Jessica Fridrich. Jpeg-phase-aware convolutional neural network for steganalysis of jpeg images. In *the 5th ACM Workshop*, 2017.

[6] Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. Stargan v2: Diverse image synthesis for multiple domains. In *CVPR*, pages 8188–8197, 2020.

[7] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *CVPR*, 2017.

[8] Joon Son Chung, Arsha Nagrani, and Andrew Zisserman. Voxceleb2: Deep speaker recognition. *arXiv preprint arXiv:1806.05622*, 2018.

[9] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

[10] Jiankang Deng, Jia Guo, Evangelos Ververas, Irene Kotsia, and Stefanos Zafeiriou. Retinaface: Single-shot multi-level face localisation in the wild. In *CVPR*, 2020.

[11] Yu Deng, Jiaolong Yang, Dong Chen, Fang Wen, and Xin Tong. Disentangled and controllable face image generation via 3d imitative-contrastive learning. In *CVPR*, 2020.

[12] 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.

[13] Bernhard Egger, William AP Smith, Ayush Tewari, Stefanie Wuhrer, Michael Zollhoefer, Thabo Beeler, Florian Bernard, Timo Bolkart, Adam Kortylewski, Sami Romdhani, et al. 3d morphable face models—past, present, and future. *TOG*, 39(5):1–38, 2020.

[14] A. Ephrat, I. Mosseri, O. Lang, T. Dekel, K Wilson, A. Hassidim, W. T. Freeman, and M. Rubinstein. Looking to listen at the cocktail party: A speaker-independent audio-visual model for speech separation. *arXiv preprint arXiv:1804.03619*, 2018.

[15] Christoph Feichtenhofer. X3d: Expanding architectures for efficient video recognition. In *CVPR*, 2020.

[16] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *ICCV*, 2019.

[17] Ohad Fried, Ayush Tewari, Michael Zollhöfer, Adam Finkelstein, Eli Shechtman, Dan B Goldman, Kyle Genova, Zeyu Jin, Christian Theobalt, and Maneesh Agrawala. Text-based editing of talking-head video. *TOG*, 38(4):1–14, 2019.

[18] Bernard Ghanem, Juan Carlos Niebles, Cees Snoek, Fabian Caba Heilbron, Humam Alwassel, Victor Escorcia, Ranjay Krishna, Shyamal Buch, and Cuong Duc Dao. The activitynet large-scale activity recognition challenge 2018 summary. *arXiv preprint arXiv:1808.03766*, 2018.

[19] Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*, 2017.

[20] Teddy Surya Gunawan, Siti Amalina Mohammad Hanafiah, Mira Kartiwi, Nanang Ismail, Nor Farahidah Za'bah, and Anis Nurashikin Nordin. Development of photo forensics algorithm by detecting photoshop manipulation using error level analysis. *IJEECS*, 7(1):131–137, 2017.

[21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[22] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, et al. Searching for mobilenetv3. In *ICCV*, 2019.

[23] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[24] Shen L. Hu, J. and G. Sun. Squeeze-and-excitation networks. In *CVPR*, 2018.

[25] 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 *CVPR*, 2020.

[26] Youngjoo Jo and Jongyoul Park. Sc-fegan: Face editing generative adversarial network with user's sketch and color. In *ICCV*, 2019.

[27] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of StyleGAN. In *CVPR*, 2020.

[28] Cheng-Han Lee, Ziwei Liu, Lingyun Wu, and Ping Luo. Maskgan: Towards diverse and interactive facial image manipulation. In *CVPR*, 2020.

[29] Lingzhi Li, Jianmin Bao, Hao Yang, Dong Chen, and Fang Wen. Faceshifter: Towards high fidelity and occlusion aware face swapping. *arXiv preprint arXiv:1912.13457*, 2019.

[30] Ji Lin, Chuang Gan, and Song Han. Tsm: Temporal shift module for efficient video understanding. In *ICCV*, 2019.

[31] Tianwei Lin, Xiao Liu, Xin Li, Errui Ding, and Shilei Wen. Bmn: Boundary-matching network for temporal action proposal generation. In *ICCV*, 2019.

[32] Tianwei Lin, Xu Zhao, Haisheng Su, Chongjing Wang, and Ming Yang. Bsn: Boundary sensitive network for temporal action proposal generation. In *ECCV*, 2018.

[33] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *ICCV*, 2015.

[34] Zhengzhe Liu, Xiaojuan Qi, and Philip HS Torr. Global texture enhancement for fake face detection in the wild. In *CVPR*, 2020.

[35] Steven R Livingstone and Frank A Russo. The ryerson audio-visual database of emotional speech and song (ravdess): A dynamic, multimodal set of facial and vocal expressions in north american english. *PLOS ONE*, 13(5):e0196391, 2018.

[36] Yuval Nirkin, Yosi Keller, and Tal Hassner. Fsgan: Subject agnostic face swapping and reenactment. In *ICCV*, 2019.

[37] Ivan Petrov, Daiheng Gao, Nikolay Chervoniy, Kunlin Liu, Sugasa Marangonda, Chris Umé, Jian Jiang, Luis RP, Sheng Zhang, Pingyu Wu, et al. Deepfacelab: A simple, flexible and extensible face swapping framework. *arXiv preprint arXiv:2005.05535*, 2020.

[38] Yuyang Qian, Guojun Yin, Lu Sheng, Zixuan Chen, and Jing Shao. Thinking in frequency: Face forgery detection by mining frequency-aware clues. In *ECCV*, 2020.

[39] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, 2015.

[40] Andreas Rossler, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies, and Matthias Nießner. Faceforensics++: Learning to detect manipulated facial images. In *CVPR*, pages 1–11, 2019.

[41] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *CVPR*, 2018.

[42] Ankit Kumar Sharma and Hassan Foroosh. Slim-cnn: A light-weight cnn for face attribute prediction. In *FG*, pages 329–335. IEEE, 2020.

[43] Yujun Shen, Bolei Zhou, Ping Luo, and Xiaoou Tang. Facefeat-gan: a two-stage approach for identity-preserving face synthesis. *arXiv preprint arXiv:1812.01288*, 2018.

[44] Aliaksandr Siarohin, Stéphane Lathuilière, Sergey Tulyakov, Elisa Ricci, and Nicu Sebe. First order motion model for image animation. In *NeurIPS*, 2019.

[45] Mingxing Tan, Bo Chen, Ruoming Pang, Vijay Vasudevan, Mark Sandler, Andrew Howard, and Quoc V Le. Mnasnet: Platform-aware neural architecture search for mobile. In *CVPR*, 2019.

[46] Mingxing Tan and Quoc V Le. Efficientnet: Rethinking model scaling for convolutional neural networks. *arXiv preprint arXiv:1905.11946*, 2019.

[47] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, et al. Deep high-resolution representation learning for visual recognition. *TPAMI*, 2020.

[48] Hang Zhang, Chongruo Wu, Zhongyue Zhang, Yi Zhu, Zhi Zhang, Haibin Lin, Yue Sun, Tong He, Jonas Mueller, R Manmatha, et al. Resnest: Split-attention networks. *arXiv preprint arXiv:2004.08955*, 2020.

[49] Hengshuang Zhao, Jiaya Jia, and Vladlen Koltun. Exploring self-attention for image recognition. In *CVPR*, 2020.