2023 International Conference on Computing, Networking and Communications (ICNC) | 978-1-6654-5719-4/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICNC57223.2023.10074246

Shallow- and Deep- fake Image Manipulation Localization Using Deep Learning

Junbin Zhang, Hamidreza Tohidypour, Yixiao Wang, and Panos Nasiopoulos
Department of Electrical and Computer Engineering
University of British Columbia
Vancouver, Canada
{zjbthomas, htohidyp, yixiaow, panosn}@ece.ubc.ca

***Abstract*—**Forged image localization is an important research task, as such images may have a tremendous impact of various aspects of society. Images can be manipulated using image editing tools (known as "shallowfakes") or, recently, artificial intelligence techniques ("deepfakes"). While there are many existing works that are designed for manipulated areas localization on either shallow- or deep-fake images, there is no single solution that works for both cases. In this paper, we propose the first solution that can perform the localization task on both shallow- and deep-fake images, with high inference accuracy. The dataset and code are available at: **https://github.com/zjbthomas/ShallowDeepFakesLocalization**.

Keywords*—Image manipulation, Manipulation localization, shallowfakes, deepfakes*

I. Introduction

Images play an important role in our daily life as they carry plenty of information. As such, images that are manipulated in order to create rumors and frauds will negatively affect our life especially when they spread at high speeds and on a large scale throughout the Internet. Even worse, nowadays, tools for manipulating images are easily accessible to everyone. In the past, manipulations were done manually via image editing tools like Photoshop. People could modify any part of an image using these tools that mainly involve signal processing techniques. As such, this conventional editing is also referred to as "shallowfakes" [1]. In recent years, the advent of deep learning and computer graphics provided people a more advanced way for generating fake images. Among all the advanced techniques, "deepfakes" became a heated topic lately, allowing users to swap or modify people's identities (e.g., faces) in images/videos relatively easy. Coupled with the ease of social media access deepfakes managed to have a huge impact, damaging the reputation of many celebrities, executives and politicians. To fight against the unwelcome disruption, researchers have been working on developing deepfake detection algorithms for many years.

In general, approaches for detecting shallowfake images fall into three conventional manipulation types: splicing (copy an object from one image to another), copy-move (copy an object within an image), and inpainting (remove an object by filling it with background pixels), as these manipulations change the meaning of the visual content [1]. Most recent works ([2]-[6]) not only provide a binary detection result for a given image (i.e., determining whether the image is authentic or fake), but also focus on solving the task of forgery region localization – generating a mask to represent what exact parts inside the image have been manipulated. These works aim at finding the discrepancies between the authentic areas and forged areas inside the image under test. Zhou et al. [2] extracted the noise distribution of an input image with Steganalysis Rich Model (SRM) filter [7] and assumed that the noise distribution between authentic and forged areas are different. They feed the original RGB image and the noise distribution into a Faster R-CNN [8] to localize the fake regions. Later, in ManTra-Net proposed by Wu et al. [3], noise distribution features are extracted by both SRM filter and BayarConv [9]. The latter is proved to be more general and robust than the SRM filter [4]. The mask is generated by feeding the features into a network that detect if local features is abnormal to its neighbors. In 2020, both CR-CNN [4] and Spatial Pyramid Attention Network (SPAN) [5] are proposed that use an attention mechanism to identify discrepancies between real and fake areas. More recently, Chen et al. [6] developed MVSS-Net, which not only considers features from the image pixels and the noise distribution extracted by BayarConv, but also identifies edges between real and fake areas for generating more accurate localization masks. However, the above-mentioned networks were only trained on shallowfake images, thus failing to detect deepfake images. For example, the latest MVSS-Net can achieve 67% accuracy on shallowfake images, while its accuracy drops to 47% when tested on the deepfake dataset we constructed (see Section II.C

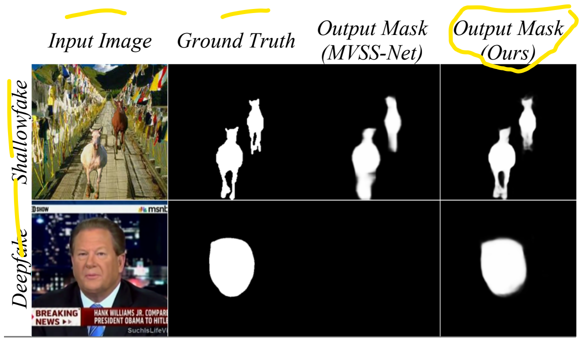

Fig. 1. Masks generated by MVSS-Net and our solution on sample shallow- and deep-fake images. The original MVSS-Net (designed and trained for only shallowfake images) works well on shallowfakes but not deepfakes. Our final solution works on both cases.

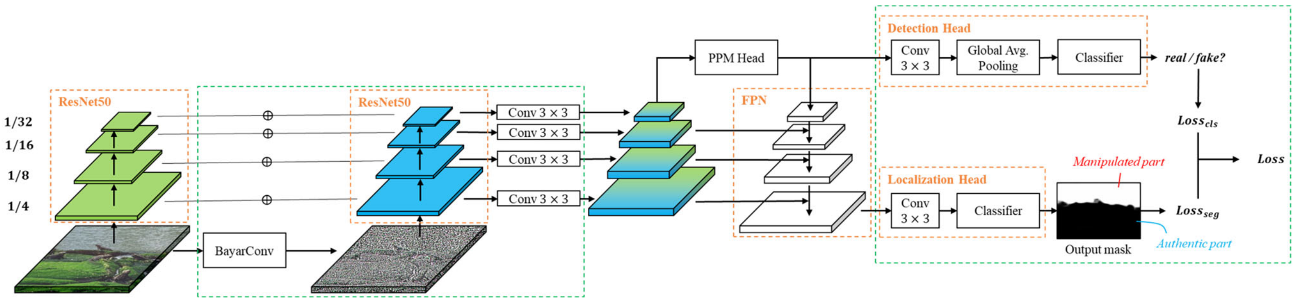

Fig. 2. Structure of our network. "⊕" means we concatenate the feature maps at each layer of the two ResNet50. Each concatenated feature map is then fed into a different convolutional layer to fuse the features. Components enclosed in green rectangles are our modifications on top of the original UPerNet.

for more details). Fig. 1 demonstrates such finding with a sample testing result.

There are also many methods to detect deepfake images/videos proposed in recent years ([10]-[14]). However, most of these works [10]-[13] only solve the task of detection with binary results. As far as we know, only one work, which is developed by Dang et al. [14], outputs manipulated regions for faces. Nevertheless, even in this case one needs to first crop the face areas from the input images when using the proposed solution. As deepfake images/videos also contain large amounts of background pixels besides the faces, the accuracy of their solution decreases when testing the original image (like the one at the second column of Fig. 1).

In this paper we propose an all-in-one solution to achieve localization for both shallow- and deep-fake images without knowing what kind of manipulations is performed. To the best of our knowledge, there exists no such solution today and our proposed approach is of high importance, as forged images may be generated using both shallowfake and deepfake techniques. We built our solution on the following two contributions:

1. We created a new comprehensive deepfake dataset and trained the network with both shallow- and deep-fake content.
2. We modified UPerNet [15], one of the state-of-the-art methods on image semantic segmentation, so the network can perform detection and localization tasks on fake images. Especially, we added BayerConv to the network so features from both the original images and the noise distribution are facilitated in making the final decision of localization.

Performance evaluations showed that our model can achieve high inference accuracy on both shallow- and deep-fake images at the end.

The rest of the paper is organized as follows: in Section II we describe how we design the network and datasets. Section III discusses the experiments we performed and demonstrates our results. In Section IV we draw the conclusions.

II. Proposed Methodology

A. Network Design

Image manipulation localization can be treated as a simplified image semantic segmentation task, with only two classes in the output mask (real or fake) [6]. As such, we decided to build our network design on top of one of the state-of-the-art image semantic segmentation networks, UPerNet [15]. UPerNet is a multi-task framework, which is originally designed for recognizing multiple visual concepts of a scene at once. That is to say, the network can output the category of a given scene, as well as what objects are inside the scene and the materials/textures of the objects. Such multi-task property fits our goal in doing detection and localization on fake images. Specifically, the detection task is similar to scene classification, with only two classes (real/fake) as output. For the localization task, we mapped it to material classification, which uses the highest resolution pixel-level information to generate masks. To make the network better understand the discrepancies between the authentic and manipulated areas, similar to previous works, we added BayerConv to the network. The final structure of our network is shown in Fig. 2, where our proposed modifications to the original network are in green rectangles.

Given an input image, we first used BayarConv, which are a set of learnable high-pass filters, to extract the noise distribution of the images. We then utilized two ResNet50 [16] as backbones to extract multi-level feature representations. Feature maps of each level from each ResNet50 are concatenated layer-wise. We adopted four convolutional layers on these concatenated feature maps to achieve better fusion. After such fusion, the highest-level information (i.e., with size $1/32$ of the original image) are fed into a Pyramid Pooling Module (PPM) head [17]. As the detection task is based on the overall image-level information instead of pixel-level information, we adopted a detection head to the output of the PPM head to obtain a binary detection result (real or fake).

The other levels of features are fused into the Feature Pyramid Network (FPN), similar to what the original UPerNet does. In the end, at the highest resolution, we attached a localization head, which outputs masks with manipulated areas highlighted. For more details about the design of UPerNet, ResNet50, PPM, and FPN, we refer readers to [15]-[18].

TABLE I. OUR TRAINING AND TEST DATASETS. FOR DEEPFAKES, NUMBERS REPRESENT THE NUMBER OF FRAMES AFTER EXTRACTION.

Dataset		Training		Test	
		# Real	# Fake	# Real	# Fake
Shallowfakes	CASIAv2	5,992	3,958	749	495
	CASIAv1	0	0	800	920
	Columbia	0	0	183	180
	COVERAGE	0	0	100	100
	NIST16	0	0	0	564
	Subtotal	**5,992**	**3,958**	**1,832**	**2,259**
Deepfakes	Youtube	5,064	0	1,688	0
	Deepfakes	0	1,000	0	600
	Face2Face	0	780	0	468
	FaceSwap	0	890	0	534
	NeuralTextures	0	995	0	597
	Subtotal	**5,064**	**3,665**	**1,688**	**2,199**
Total		**11,056**	**7,623**	**3,520**	**4,458**

B. Loss Functions

Our final loss function consists of two parts. On the image-level, the network should provide correct binary detection results. We used binary cross-entropy (BCE) to compute such a classification loss ($Loss_{clf}$ in Fig. 2). On the pixel-level, similar to MVSS-Net [6], we used Dice loss ($Loss_{seg}$), which measures the overlap between the generated mask and a ground truth, to enhance the correctness of the generated mask. We dropped the edge loss used to train the original MVSS-Net as it is not needed for our network. Our final loss is calculated as:

$$L = \alpha \cdot Loss_{clf} + \beta \cdot Loss_{seg}$$

where we set $\alpha = 0.04$ and $\beta = 0.16$.These coefficients were found through empirical studies to achieve the best trade-off between both losses.

C. Dataset Construction

Our dataset consists of two parts, one with shallowfake images and one with deepfake images. Table I lists the details of the final training and test sets we used.

Shallowfakes. Similar to [6], our training dataset is from CASIAv2 [16]. This dataset includes 7,490 authentic images and 4,948 forged images with slicing and copy-move manipulations, covering different kinds of objects. We split the dataset into training, validation, and test sets at a ratio of 8:1:1. That is to say, at the end we use 5,992 authentic images and 3,958 forged images in training our network.

For testing, besides the above-mentioned 10% images from CASIAv2, we also included all images from CASIAv1 [16], Columbia [17], COVERAGE [18], and NIST16 [19] in our test set to evaluate the generalizability of our model. In the end, our shallowfake test set includes 1,832 real and 2,259 fake images, covering all three types of shallowfake manipulations (i.e., slicing, copy-move, and inpainting).

Deepfakes. There is no existing image dataset with ground truth masks of manipulated areas for deepfakes. Therefore, we built our own dataset on top of the famous FaceForensics++

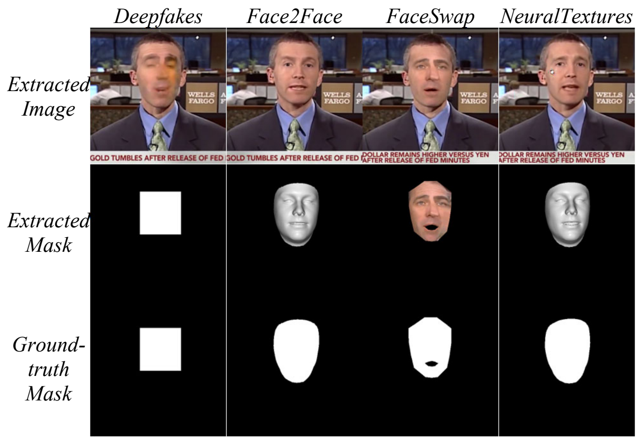

Fig. 3. Masks in our deepfake dataset.

[20], which is, as far as we know, the only dataset that provides masks for most of its deepfake video. FaceForensics++ contains 1,363 authentic videos from Youtube, 3068 videos from Google's DeepFakeDetection dataset, and 5,000 fake videos generated by five automated face manipulation methods (namely, Deepfakes, Face2Face, FaceShifter, FaceSwap, and NeuralTextures).

The next step is to extract image frames from the videos in FaceForensics++. We want the numbers of real and fake frames in the deepfake training and test sets to be close to those of the shallowfake ones. To achieve this, we randomly selected 7 frames per real videos and 2 frames per fake videos generated by Deepfakes, Face2Face, FaceSwap, and NeuralTextures. We excluded Google's DeepFakeDetection, as frames in its forged videos do not align with those in its mask videos, and doing the alignment is out of the scope of this paper. We also excluded 1,000 FaceShifter videos since there were no masks available for them. Due to accessibility issues, we could not download some of the authentic and fake videos. Eventually, we ended up with 8,449 authentic frames and 7,330 forged frames. We further divided these real frames and fake frames into training, validation, and test sets with a ratio of 6:2:2 and 5:2:3.

The contents of mask videos are different for different face manipulation methods. For example, the manipulated areas in Deepfakes are rectangle instead of face-like (we confirmed this by checking the differences between authentic videos and the corresponding manipulated videos). The other three are computer graphics-based methods, so the masks are 3D-looking (see Fig .3, second row). We discard all colour information in the original mask videos and generated binary ground-truth masks at the end.

Some example masks generated for our deepfake dataset are shown in Fig. 3. Readers can check the masks for selected shallowfake datasets from their original papers [16]-[19].

III. EXPERIMENTS AND RESULTS

We compare our network with and without BayerConv against the state-of-the-art MVSS-Net [6]. We trained each

TABLE III. EVALUATION RESULTS OF OUR EXPERIMENTS. IN THE THIRD COLUMN, "S" MEANS TRAINING USING SHALLOWFAKE DATASET, AND "D" MEANS TRAINING USING DEEPFAKE DATASET. WE HIGHLIGHT THE BEST RESULTS IN **BOLD**, AND THE SECOND-BEST RESULTS ARE HIGHLIGHTED USING UNDERLINE.

	Method	Training Set		Test Set											
				Shallowfakes				Deepfakes				Both			
				F1			AUC	F1			AUC	F1			AUC
		S	D	Pixel-level	Image-level	Combined		Pixel-level	Image-level	Combined		Pixel-level	Image-level	Combined	
1	MVSS-Net	√		**0.60**	**0.68**	**0.64**	0.79	0.24	0.51	0.33	0.53	0.42	0.62	0.50	0.68
2			√	0.45	0.00	0.01	0.50	0.90	0.93	0.91	0.98	0.67	0.62	0.64	0.71
3		√	√	0.59	0.64	0.62	0.78	0.90	0.94	0.92	0.97	**0.75**	0.81	**0.78**	0.87
4	Ours w/o BayerConv	√		**0.60**	0.62	0.61	0.83	0.30	0.29	0.29	0.55	0.45	0.47	0.46	0.72
5			√	0.44	0.27	0.33	0.43	**0.92**	**0.95**	**0.94**	**0.99**	0.68	0.69	0.68	0.78
6		√	√	0.59	0.65	0.62	**0.85**	0.87	0.93	0.90	0.98	0.72	0.81	0.76	0.91
7	Ours with BayerConv	√		**0.60**	**0.68**	**0.64**	**0.85**	0.27	0.29	0.28	0.47	0.44	0.51	0.47	0.70
8			√	0.45	0.12	0.19	0.45	**0.92**	**0.95**	**0.94**	**0.99**	0.68	0.66	0.67	0.81
9		√	√	**0.60**	0.66	0.63	0.84	0.91	0.94	0.93	**0.99**	**0.75**	**0.82**	**0.78**	**0.92**

network under three different settings: (1) with only the shallowfake training set; (2) with only the deepfake training set; and (3) with both the shallow- and deep-fake training sets.

For fair comparison, we train each of the above networks for 50 epochs with the batch size set to 32. The learning rate for this stage was set initially to 10^{-4} and was decayed by a factor of 10 when the validation loss did not improve over 5 epochs. For optimization, we used ADAM solvers with default momentum term $\beta_1 = 0.9$ and $\beta_1 = 0.999$. Training was performed using four 32 GB NVIDIA V100 Volta GPUs on a state-of-the-art advanced research computing network [20].

We reported three different F1 values: (1) pixel-level, which represents the accuracy of output masks compared to the ground-truth masks for localization task; (2) image-level, which checks if the output binary results (real/fake) match the ground-truth for detection task; and (3) a combined F1 value, which is the harmonic mean of the pixel-level F1 and image-level F1. We set the detection threshold as 0.5 for these three F1 values. We also reported image-level "Area under the Receiver Operating Characteristic (ROC) Curve" (AUC).

The datasets we used in evaluation are discussed in Section II.C and shown in Table I. We reported performances of the networks on all subsets from shallow- or deep-fake datasets separately, and the combination of the shallow- and deep-fake datasets. All of our evaluation results can be found in Table II.

Effects of training sets. For all three different networks, we notice that using both shallow- and deep-fake training sets in training does not dramatically affect the accuracy on each separate set. Nevertheless, using both training sets in training allows the networks to learn general features that are needed for detecting both shallow- and deep-fake images. For example, when training our solution using both training sets (row 9 in Table II), on shallowfake test set, the combined F1 (0.63) and AUC (0.84) are closed to those (0.64 and 0.85) when training using shallowfake training set only (row 7). Similar trend can be found when comparing data in rows 8 and 9 for the deepfake test set. At the end, when the test set consists of both shallow- and deep-fake images (last four columns), training using both training sets can achieve the best combined F1 (0.78) and AUC (0.92) compared to training using either one training set.

In addition, our results also show that that the networks that are only trained with one training set lack the ability to detect images from another set. This matches the example that is shown in Fig. 1. More precisely, the original MVSS-Net, which is designed and trained only for shallowfake images, cannot identify most forged areas in deepfake images. In fact, in row 1, while MVSS-Net trained with shallowfake training set has high combined F1 (0.64) and AUC (0.78) on the shallowfake test set, its combined F1 and AUC on deepfakes are only 0.33 and 0.53.

Effects of network design. Comparing results in rows 1/2/3 and 4/5/6, we conclude that using a network based on UPerNet can achieve a better result compared to MVSS-Net, regardless of what training set is used during training. For example, when training using both shallow- and deep-fake training sets, our network (even without the BayerConv) achieves an AUC of 0.91 (row 6), with an 5% improvement to MVSS-Net (0.87, row 3).

Besides, feeding the noise features extracted by BayarConv can further improve the inference accuracy. Comparing networks trained with only shallowfake images (rows 4 and 7), using BayerConv achieves an AUC of 0.85 compared to the one without it (0.83). For networks trained with only deepfake images (rows 5 and 8), with and without BayerConv achieve the same level of combined F1 of 0.94 and AUC of 0.99. When training with both test sets (rows 3, 6 and 9), our final network obtains the best combined F1 (0.78) and AUC (0.92) compared to the state-of-the-art networks.

Qualitative visualization. For visualization purposes, we show some samples of output masks of all three networks in Fig. 4. All of these networks were trained with both shallow- and deep-fake training sets.

We noticed that our network tends to generate clear boundaries for the forged areas. For example, in column 1, the forged area in our mask looks more like a dragon. MVSS-Net does not output the shape of the dragon and the output of the network without BayerConv is not as clear as our final solution. On deepfake images like the one in column 3, while MVSS-Net

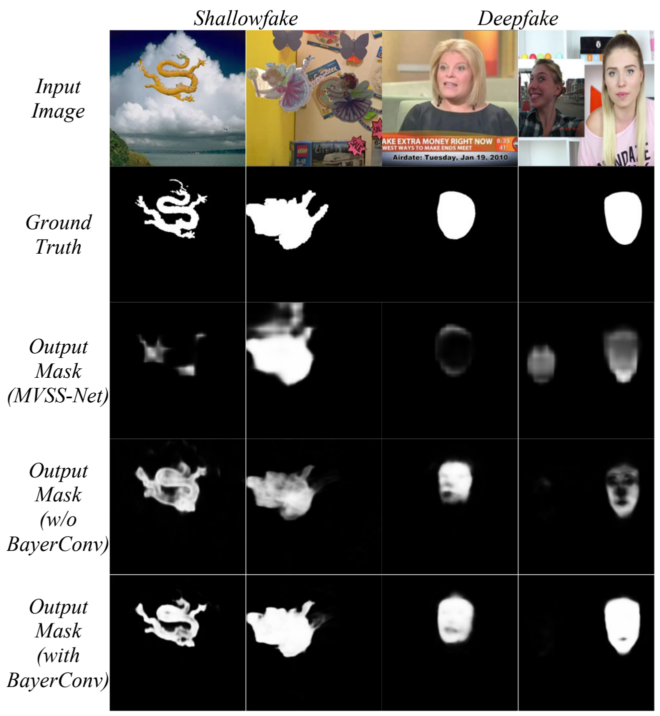

Fig. 4. Samples of network outputs for three networks.

and network without BayerConv can successfully classify the image as fake, their output masks are not clear enough. Our final network provides a mask closer to the ground truth in this case.

In addition, there are less false-positive pixels in our generated masks compared to MVSS-Net. In column 2, MVSS-Net detects incorrect pixels at the top of the manipulated areas, while UPerNet-based solutions suppress such false-positive pixels. On the left-hand side of the image in column 4, MVSS-Net detects the authentic face in the photo of the live news. Our UPerNet-based solutions do not consider such face as fake when generating masks.

IV. Conclustion

In this paper, we designed a novel deep learning network for detecting fake images and localizing manipulated areas in both shallowfake and deepfake images. In order to achieve this, we created a new comprehensive deepfake dataset for training and evaluation purposes. Initially, we based our network design on UPerNet, one of the state-of-the-art image segmentation methods. Afterwards, we added BayerConv to the network to make sure features of both the original images and the noise distribution are facilitated in making the final decision of localization. Performance evaluations showed that our proposed method achieved the accuracy of 82% for real/fake categorization and 75% for localization of our shallowfake and deepfake test images, respectively, outperforming the state-of-the-art shallow-fake detection method retrained on our dataset. Future work involves developing more robust joint shallowfake and deepfake detection methods for image and video content.

Acknowledgment

This work was supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC – PG 11R12450), and TELUS (PG 11R10321). This research was enabled in part by support provided by Compute Canada (www.computecanada.ca).

References

[1] L. Verdoliva, “Media forensics and DeepFakes: an overview,” *IEEE J Sel Top Signal Process*, vol. 14, pp. 910–932, June 2020.

[2] P. Zhou, X. Han, V. I. Morariu, and L. S. Davis, “Learning rich features for image manipulation detection,” in *CVPR*, 2018.

[3] Y. Wu, W. AbdAlmageed, and P. Natarajan, “ManTra-Net: manipulation tracing networks for detection and localization of image forgeries with anomalous features,” in *CVPR*, 2019.

[4] C. Yang, H. Li, F. Lin, B. Jiang, and H. Zhao, “Constrained R-CNN: a general image manipulation detection model,” in *ICME*, 2020.

[5] X. Hu, Z. Zhang, Z. Jiang, S. Chaudhuri, Z. Yang, and R. Nevatia, “SPAN: spatial pyramid attention network for image manipulation localization,” in *ECCV*, 2020.

[6] X. Chen, C. Dong, J. Ji, J. Cao, and X. Li, “Image manipulation detection by multi-view multi-scale supervision,” in *ICCV*, 2021.

[7] J. Fridrich and J. Kodovsky, “Rich models for steganalysis of digital images,” *TIFS*, vol. 3, pp. 868-882, May 2012.

[8] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: towards real-time object detetion with region proposal networks,” in *NIPS*, 2015.

[9] B. Bayar and M. C. Stamm, “Constrained convolutional neural networks: a new approach towards general purpose image manipulation detection,” *IEEE Trans. Inf. Forensics Secur.*, vol. 13, pp. 2691-2706, April 2018.

[10] P. Zhou, X. Han, V. I. Morariu,and L. S. David, “Two-stream neural networks for tampered face detection,” in *CVPRW*, 2017.

[11] D. Afchar, V. Nozick, J. Yamagishi, and I. Echizen, “MesoNet: a compact facial video forgery detetion network,” in *WIFS*, 2018.

[12] E. Sabir, J. Cheng, A. Jaiswal, W. AdbAlmageed, I. Masi, and P. Natarajan, “Recurrent convolutional strategies for face manipulation detection in videos,” in *CVPR*, 2019.

[13] Y. Wang and A. Dantcheva, “A video is worth more than 1000 lies. Comparing 3DCNN approaches for detecting deepfakes.” In *FG*, 2020.

[14] H. Dang, F. Liu, J. Stehouwer, X. Liu, and A. K. Jain, “On the detection of digital face manipulation,” in *CVPR*, 2020.

[15] T. Xiao, Y. Liu, B. Zhou, Y. Jiang, and J. Sun, “Unified perceptual parsing for scene understanding,” in *ECCV*, 2018.

[16] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *CVPR*, 2016.

[17] H. Zhao, J. Shi, X. Qi, X. Wang, J. Jia, “Pyramid scene parsing network,” in *CVPR*, 2017.

[18] T. Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, S. Belongie, “Feature pyramid networks for obbject detection,” in *CVPR*, 2017.

[19] J. Dong, W. Wang, and T. Tan, “CASIA image tampering detection evaluation database,” in *ChinaSIP*, 2013.

[20] Y. Hsu and S. Chang, “Detecting image splicing using geometry invariants and camera characteristics consistency,” in *ICME*, 2006.

[21] B. Wen, Y. Zhu, R. Subramanian, T. Ng, X. Shen, and S. Winkler, “COVERAGE – a novel database for copy-move forgery detection,” in *ICIP*, 2016.

[22] H. Guan, M. Kozak, E. Robertson, Y. Lee, A. N. Yates, A. Delgado, D. Zhou, T. Kheyrkhah, J. Smith, and J. Fiscus, “MFC datasets: large-scale benchmark datasets for media forensic challenge evaluation,” in *WACVW*, 2019.

[23] A. Rössler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Nieeßner, “FaceForensics++: learning to detect manipulated facial images,” in *ICCV*, 2019.

[24] Compute Canada state-of-the-art advanced research computing network. https://www.computecanada.ca (accessed Oct. 1, 2021).