Spotting tell-tale visual artifacts in face swapping videos: strengths and pitfalls of CNN detectors

Riccardo Ziglio[1,3], Cecilia Pasquini[1], Silvio Ranise[1,2]
[1]*Center for Cybersecurity, Fondazione Bruno Kessler, Trento, Italy*
[2]*Department of Mathematics, University of Trento, Italy*
[3]*DIBRIS, University of Genoa, Italy*
{rziglio, c.pasquini, ranise}@fbk.eu

arXiv:2506.16497v1 [cs.CV] 19 Jun 2025

Abstract

Face swapping manipulations in video streams represents an increasing threat in remote video communications, due to advances in automated and real-time tools. Recent literature proposes to characterize and exploit visual artifacts introduced in video frames by swapping algorithms when dealing with challenging physical scenes, such as face occlusions. This paper investigates the effectiveness of this approach by benchmarking CNN-based data-driven models on two data corpora (including a newly collected one) and analyzing generalization capabilities with respect to different acquisition sources and swapping algorithms. The results confirm excellent performance of general-purpose CNN architectures when operating within the same data source, but a significant difficulty in robustly characterizing occlusion-based visual cues across datasets. This highlights the need for specialized detection strategies to deal with such artifacts.

Index Terms

face swapping, face verification, remote video calls, forensic detection

I. Introduction

The synthesis and manipulation of facial images and videos have achieved increasingly hyper–realistic results in recent years [4], leading to numerous research efforts for the automated identification of non-genuine visual data [25] [27].

The wide majority of the detection techniques proposed in the literature operates in a passive fashion (i.e., assuming no or little a priori information on the media generation and distribution pipeline) [1]. Data-driven approaches based on deep networks are predominant, with an extensive use of general-purpose architectures originally devised for image classification, revisited for these tasks [2] [23] [28]. Practical applications range from the analysis of user-generated content on the web for fact checking purposes to the validation of digital visual evidence in forensic investigations.

A highly relevant scenario where detecting manipulations becomes essential is remote live video communications, where one (or more) subject stands in front of a camera capturing the video scene in real-time and interacts remotely with another subject or with an automated interface. If not properly protected, such scenario might be vulnerable to advanced impersonation attacks enabled by real-time video face manipulation tools, potentially leading to severe security issues. It is the case of video calls, which recently emerged as a channel for perpetrating scams and frauds[1]. In fact, remote identity proofing processes (such as Know-Your-Customer applications) based on face verification [6] [7] [18] are threatened by these new tools [13] [20], in addition to known presentation attack vectors [21]. In this context, face swapping is a particularly powerful technique, as it allows an attacker to modify only the facial area (typically used for verifying the subject's identity), while leaving the rest of the scene under his control.

The detection of face swapping in remote video communications has so far been relatively little explored, with only a few approaches explicitly targeting this scenario [8], [26]. Recently, the work in [16] has proposed to exploit the interactive nature of the setting to create a detection advantage, inspired by biometric challenge-response protocols [12]: the user performs (mostly voluntary) actions denoted as *challenges*, which are devised to interfere with manipulation algorithms and, in case of ongoing impersonation attack, produce visible visual artifacts in the video frame. While the quality levels of manipulation engines are rapidly improving, handling diversified challenge requests in real-time (with no post-processing options) represents a complex task. Face occlusions are a notable case where evident artifacts are likely to be produced, providing precious hints for human observers.

This paves the way for automated detectors exploiting such tell-tale visual cues to identify swapped faces over genuine ones. The authors in [16] provide empirical evidence of a learning-based approach, validated on their proposed dataset. However, a known crucial issue of data-driven solutions for manipulation detection is their ability to generalize in non-aligned testing modes [27] [1], thus when testing data do not come from the very same data source as the training set. Thus, it remains an open question to which extent learned cues can effectively detect occlusion artifacts across data coming from different acquisition settings and manipulation algorithms.

[1]https://edition.cnn.com/2024/02/04/asia/deepfake-cfo-scam-hong-kong-intl-hnk/index.html

This work has been submitted to the IEEE for possible publication. Copyright may be transferred without notice, after which this version may no longer be accessible.

The experimental analysis proposed in this paper investigates this scenario. We introduce a newly collected data corpus of genuine and manipulated videos systematically depicting subjects with face occlusions, as reported in Section II. This, together with the dataset in [16], allows us to perform a cross-dataset analysis (described in Section III) and benchmark the performance of CNN-based data-driven detection models in different experimental settings. The influence of both the data source and the frame content are analyzed, revealing insightful empirical evidence reported in Section IV.

II. Dataset Collection

We denote as FOWS (Face Occlusion With Swapping) the newly collected dataset. Each video clip depicts a human subject in front of a camera who is asked to perform voluntary actions (also indicated as *challenges*) leading to face occlusions. In fact, in [16] those have proved highly effective in producing visual artifacts after face swapping and obtained a high usability score when rated by users among the different considered alternatives (i.e., head movements, facial deformation, facial illumination). In particular, our subjects are asked to place in front of their face either their own hand (*hand occlusion* challenge) or a supplied rectangular object (*object occlusion* challenge) in predefined locations.

We developed a recording interface to streamline the video collection from volunteer users. The interface contains a short description and some visual examples of correct/incorrect positions for each of the challenges to be performed. Then, during the actual execution, a *guiding video* displaying visual instructions on where to place the face and the occluding objects is overlaid on the current camera frame, as represented in Figure 1(a). Subjects have been instructed to follow as closely as possible the motion shown in the guiding videos; for each clip, we visually verified the compliance of the user's motion with the displayed instructions at the recording phase.

As pointed out in [16], multiple and diversified occlusions per user help in better detecting potential manipulation attacks. Therefore, we created three different variants of the hand and object occlusion challenges to be performed by each user, where the occluding objects follow different spatial trajectories and overlaps with diverse portions of the face. Thus, six different guiding videos have been developed lasting around 40 seconds, all of them are structured as follows (see Figure 1(a)): they start with a static temporal segment depicting only the subject's face in a prescribed location, followed by a more dynamic segment where the occluding object moves and overlaps the face (lasting around 15 seconds). We recorded videos from 7 volunteers, each of them recording the six challenge variants. All videos were captured with a Logitech C920 HD PRO webcam recording at 1080p resolution and 30 fps. From the dynamic segment of each genuine video, three manipulated versions have been then generated through the recent face-swapping algorithms SimSwap (SS_F) [3], GHOST (GH_F) [11] and FaceDancer (FD_F) [22]. These algorithms are particularly powerful as they perform face swapping from a single image of the target face. We used royalty-free pictures of celebrities as target faces. Examples of genuine and manipulated frames with and without face occlusion are reported in Figure 1(b). In agreement with [16], occlusion-based artifacts are present in all manipulated videos.

In total, the dataset is composed of 168 videos (42 genuine and 126 manipulated), and approximately 70k video frames.

III. Experimental Setup

The dataset and part of the code used used in this study are available at https://st.fbk.eu/complementary/IWBF2025.

A. Datasets

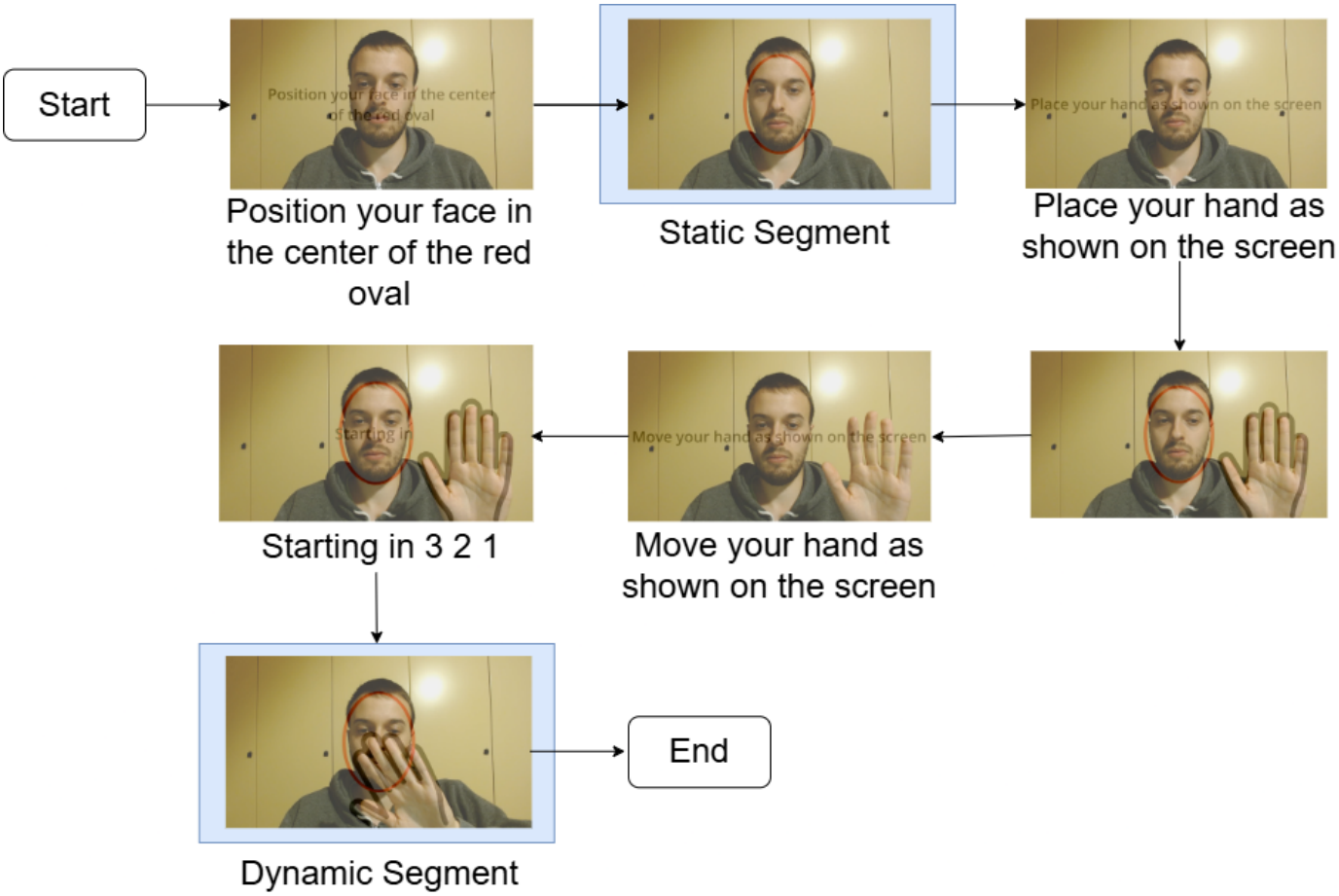
Our experimental analysis involves the previously described FOWS dataset and the one proposed in [16], denoted as GOTCHA dataset. Among the available datasets including face swapping manipulations, this is the only one where subjects systematically perform face occlusion and thus fits the purpose of our study. The dataset involves 47 participants performing several challenges; it includes genuine videos and two corresponding swapped versions, obtained through the algorithms DeepFaceLab (DFL_G) [19] and FSGAN ($FSGAN_G$) [17]. Data are provided as sequences of static frames. In particular, we selected the data related to the hand and object occlusion challenges only; for each user, swapped videos from three different target faces for each challenge are considered among the available ones.

Overall, from the union of the two datasets we can obtain seven data partitions: five corresponding to the swapping algorithms (denoted as SS_F, GH_F, FD_F, DFL_G, and $FSGAN_G$) and the two sets of genuine frames (OR_F and OR_G), where the subscripts indicate the dataset they belong.

B. Frame category separation

As it will be discussed in Section IV, we perform detection tests separately on two categories of frames: the ones where the face is fully shown and the ones where the subject places the hand/object in front of the face, indicated as **no-occ** and **occ**, respectively. This is functional for our experimental analysis (see Section IV-B), as the sought occlusion-based artifacts in the swapped data appear only in the latter category.

This separation is performed within all data partitions in a semi-automated manner for both datasets. For FOWS, we applied the BlazeFace detector offered by the Google mediapipe framework [10] to locate the face bounding box in all video frames, which was then increased by a factor of 1.3 (as suggested in [23], [24]), to extract the whole user face. Then, 100 frames are



(a)

(b)

Fig. 1: (a) Pipeline of the video recording with the guiding videos for the hand occlusion challenge. (b) Examples of original (left) and manipulated (right) frames.

sampled from the static segment (see Figure 1(a)) for the **no-occ** category; for the **occ** category, we run the BlazeFace detector with strict tolerance on the frames of dynamic segment, and selected 100 frames among the ones where such detector failed due to occlusion.

The latter procedure has also been used on the GOTCHA dataset, which does not have separated static and dynamic segments. Therefore, the separation based on the BlazeFace detector has been applied on the whole temporal sequence. For both datasets, a manual revision of the frames identified for each category was performed to refine the separation.

a) Training/Testing split: For FOWS, data related to 5 users have been used for training, while the remaining 2 for testing. We consider the challenging setting where the two testing subjects are (i) the only woman in the dataset and (ii) the only man with a thick beard. The 100 frames sampled from the dynamic and static segment during category separation yield 600 frames per user in each category, multiplied by 4 (the original version plus the three manipulated counterparts). Thus, for both **occ**

TABLE I: Numerosity of GOTCHA training/testing splits.

	OR_G	DFL_G	$FSGAN_G$	Total
	trn — tst	trn — tst	trn — tst	trn — tst
occ	28395— 19463	13716— 9899	14057— 10019	56168— 39381
no-occ	15188 — 13200	7922 — 6338	7906 — 6407	31016 — 25945

and **no-occ** data, the training and testing splits are composed by 12000 (3000 original/ 9000 manipulated) and 4800 (1200 original/ 3600 manipulated) frames.

In GOTCHA, 30 users are selected for training and the remaining 17 for testing, making sure that no subject appears in both sets either as host or target face. In general, GOTCHA provides more original samples than swapped ones, probably due to the fact that fake videos were saved at lower fps after processing. To compensate for this, we sample original frames for each user and each challenge, to obtain a comparable number of frames. The resulting split is reported in Table I.

C. Detection models

We consider a battery of five CNN architectures, most of them previously used on face video manipulation detection [23]. While we are aware that more sophisticated approaches exist, the analysis in [28] shows that such general-purpose architectures provide performance comparable to handcrafted approaches, in front of a reduced complexity. We therefore consider three baseline models pretrained on ImageNet: MobileNetv2 (*MobNet*), EfficientNetB4 (*EffNetB4*), and XceptionNet (*Xception*) [5]. Moreover, we include two CNNs already specialized in deepfake detection:

- *ICPR2020*: proposed in [2], based on the EfficientNetB4 architecture and trained on the DFDC dataset;
- *NeurIPS2023*: used in [28], based on the Xception architecture and trained on FaceForensics++.

Both models are selected due to their performance in the respective papers. We use the original versions released by the authors [2] [28] and replicate their data preprocessing to ensure fairness in the evaluation.

D. Testing protocol

We adopt a binary classification framework: the frames in OR_F and OR_G are associated to the label 0, the manipulated frames in the other partitions are associated to the label 1. Classification is performed according to the default 0.5 threshold on the CNN output score. All CNN models are retrained on both FOWS and GOTCHA datasets, separately. For the baselines, we use the *focal loss* function [14], a modified version of the Cross Entropy loss that adaptively focuses the model learning on less represented samples, to counter the class imbalance between original and manipulated data in the FOWS dataset. A simple early-stopping procedure (with patience 3) was adopted to control overfitting during training. Models are trained for 10 epochs on FOWS, using the Adam optimizer as in [28] [2], and for 15 epochs on GOTCHA, applying the AdamW optimizer [15]. After some preliminary tests (not reported for the sake of space), we identified the best performing PyTorch data augmentation configuration as follows: random resize crop at (224,224), random horizontal flip, rotation in the range [-5,5] degrees, and color jitter with default values. For the *ICPR2020* and *NeurIPS2023* CNNs, we used the same loss and augmentation configurations as in the original code.

E. Metrics

We applied different metrics for a comprehensive evaluation. Based on the model decisions, we report the *Balanced Accuracy* (B-ACC), defined as the average between sensitivity and specificity, which is typically used in case of imbalanced datasets and provides more representative values with respect to the overall accuracy; we also report the accuracy ACC(p) separately for each partition p defined in Section III-A. Moreover, we compute the *Area Under Curve (AUC)*, obtained by binarizing the network output with different thresholds, and the *Equal Error Rate (EER)*.

IV. Results

We first analyze model performance in a fully aligned setting, i.e., where train and test data belong to the same dataset and frame category. In Table II, results for FOWS and GOTCHA are arranged column-wise, while the top (bottom) row refers to the **occ** (**no-occ**) category.

It can be observed that most models yield excellent results in all cases. For GOTCHA in particular, the distinction is essentially perfect for both the **occ** and **no-occ** categories. We observe that *ICPR2020* and *NeurIPS2023* do not obtain superior results with respect to their baseline versions, *EffNetB4* and *Xception*, on the **no-occ** data. In other words, it emerges that being previously trained on swapped data from other datasets (DFDC and FF++) does not bring an advantage in dealing with swapped data coming from different sources (FOWS and GOTCHA). This holds in particular for *NeurIPS2023*, for which retraining on FOWS (where less retraining data are available with respect to GOTCHA) does not lead to accurate results. This shows a general difficulty of detectors in effectively capturing discriminative cues for data subject to equivalent manipulations (face swapping with no occlusion) but coming from different data sources, as also observed in [28].

TABLE II: Model performance in the fully aligned setting.

Train: FOWS, **occ**. Test: FOWS, **occ**.

Model	B-ACC	ACC(OR_F)	ACC(SS_F)	ACC(GH_F)	ACC(FD_F)	AUC	EER
MobNet	99.56	99.17	99.92	100	99.92	1.0000	0.0033
EffNetB4	97.88	95.75	100	100	100	0.9999	0.0050
Xception	99.10	98.25	99.83	100	100	0.9999	0.0033
ICPR2020	99.05	98.92	99.92	100	97.58	0.9995	0.0100
NeurIPS2023	71.03	46.83	91.42	100	94.25	0.8472	0.2700

Train: GOTCHA, **occ**. Test: GOTCHA, **occ**.

Model	B-ACC	ACC(OR_G)	ACC(DFL_G)	ACC($FSGAN_G$)	AUC	EER
MobNet	99.96	99.91	100	100	1.0000	0.0001
EffNetB4	99.99	100	100	99.95	1.0000	0.0000
Xception	99.98	99.96	100	100	1.0000	0.0001
ICPR2020	99.97	99.96	100	99.96	1.0000	0.0003
NeurIPS2023	99.76	99.51	100	100	1.0000	0.0029

Train: FOWS, **no-occ**. Test: FOWS **no-occ**.

Model	B-ACC	ACC(OR_F)	ACC(SS_F)	ACC(GH_F)	ACC(FD_F)	AUC	EER
MobNet	100	100	100	100	100	1.0000	0.0000
EffNetB4	100	100	100	100	100	1.0000	0.0000
Xception	99.92	100	100	100	100	1.0000	0.0000
ICPR2020	95.72	99.00	99.83	99.83	77.67	0.9965	0.0325
NeurIPS2023	66.51	74.33	51.17	53.08	71.83	0.7368	0.3392

Train: GOTCHA, **no-occ**. Test: GOTCHA, **no-occ**.

Model	B-ACC	ACC(OR_G)	ACC(DFL_G)	ACC($FSGAN_G$)	AUC	EER
MobNet	100	99.76	100	100	1.0000	0.0013
EffNetB4	100	100	100	100	1.0000	0.0000
Xception	100	99.21	100	100	0.9998	0.0020
ICPR2020	100	99.99	100	100	1.0000	0.0000
NeurIPS2023	100	100	100	100	1.0000	0.0000

A. *Cross-dataset tests*

In order to assess their generalization capabilities, detection models are tested in a cross-dataset fashion, thus training on one dataset and testing on the other. Table III reports the results for both categories, where right and left blocks refer to the training dataset. Here, the accuracy metrics show a significant drop in performance for all settings. The B-ACC almost halves for most models, and accuracy values on the different dataset partition reveals poor capabilities of correctly classifying video frames. We observe that the prominent type of misclassification error changes together with the testing dataset: models trained on GOTCHA tend to miss manipulated frames on FOWS, while models trained on FOWS are prone to classifying genuine GOTCHA frames as manipulated. While not reporting here the full results for the sake of space, we stress that the aligned and cross-dataset setting tests have also been replicated in a transfer learning mode (by training the last layer only instead of the whole network) for an extended evaluation. The results obtained are consistent: the median B-ACC over the models in the aligned setting is equal to 90%, while it drops to 65% in the cross-dataset setting.

An interesting observation is that the strong performance decrease in terms of accuracy is not always reflected in the AUC and EER metrics: when training on GOTCHA and training on FOWS, models like *MobNet* and *Xception* retain rather good values of AUC and EER for both categories. This indicates that the model distinguishes genuine from manipulated samples in terms of prediction scores, but does not classify them correctly with the default threshold on 0.5, which was instead effective on the training dataset. Figure 2 reports the score distributions (in logarithmic scale) in those cases, showing that there is indeed a separation but approaching zero. While denoting a certain generalization capabilities, this shift on the score distribution represents a relevant issue in practical situations where unseen data (potentially coming from different data sources) would be tested through a pretrained model, as applying the prescribed threshold would lead to strong inaccuracies. The charts in Figure 3 visualize this scenario, showing that no model behaves reliably on all partitions.

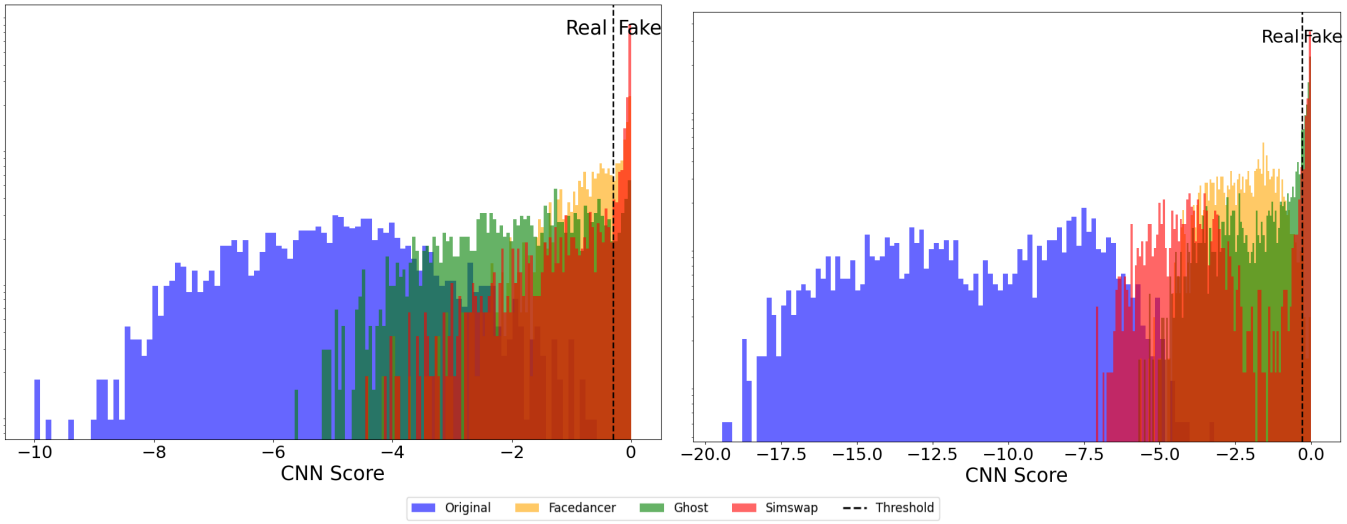


Fig. 2: Histogram of the decision score for *MobNet* (left) and *Xception* (right) trained GOTCHA and tested on FOWS.

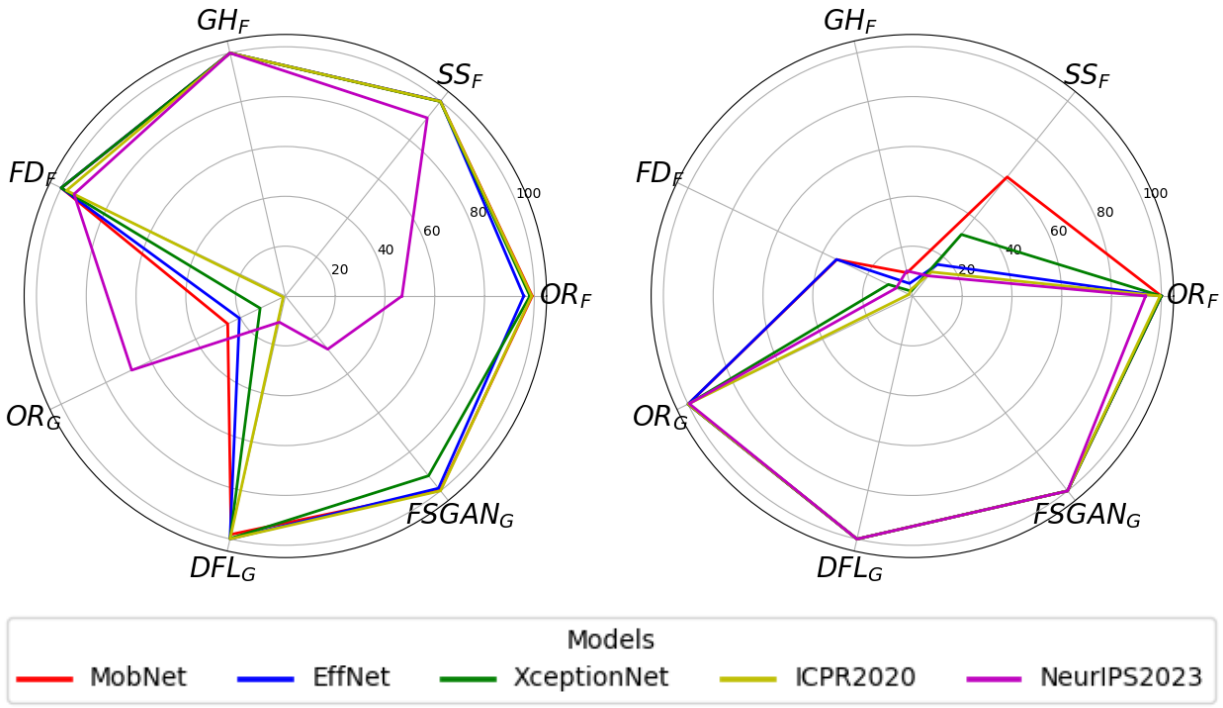


Fig. 3: Visualization of the accuracy on all partitions of the models trained on FOWS (left), and the ones trained on GOTCHA (right).

TABLE III: Model performance in the cross-dataset setting.

Train: FOWS, **occ**. Test: GOTCHA, **occ**.

Model	B-ACC	ACC(OR_F)	ACC(DFL_G)	ACC($FSGAN_G$)	AUC	EER
MobNet	62.34	25.72	98.12	99.77	0.7120	0.3135
EffNetB4	59.92	20.52	99.98	98.66	0.7833	0.2560
Xception	53.64	11.27	99.80	92.26	0.6181	0.3968
ICPR2020	50.46	0.94	100	99.96	0.8555	0.2153
NeurIPS2023	43.79	68.55	10.78	27.29	0.3901	0.5703

Train: GOTCHA, **occ**. Test: FOWS, **occ**.

Model	B-ACC	ACC(OR_F)	ACC(SS_F)	ACC(GH_F)	ACC(FD_F)	AUC	EER
MobNet	67.39	100	61.08	9.58	33.67	0.9725	0.0942
EffNetB4	59.20	100	16.17	5.17	33.83	0.9243	0.1675
Xception	57.38	100	31.58	2.00	10.67	0.9405	0.1442
ICPR2020	52.50	99.50	12.33	2.67	1.50	0.7630	0.3075
NeurIPS2023	51.49	93.67	10.33	10.33	7.25	0.4736	0.5375

Train: FOWS, **no-occ**. Test: GOTCHA, **no-occ**.

Model	B-ACC	ACC(OR_F)	ACC(DFL_G)	ACC($FSGAN_G$)	AUC	EER
MobNet	53.48	7.08	99.81	99.95	0.6552	0.3784
EffNetB4	51.98	3.95	100	100	0.8461	0.2113
Xception	54.14	8.704	99.15	100	0.6459	0.3887
ICPR2020	51.23	4.10	99.15	97.58	0.6708	0.3448
NeurIPS2023	31.41	25.57	15.35	59.11	0.2078	0.7363

Train: GOTCHA, **no-occ**. Test: FOWS, **no-occ**.

Model	B-ACC	ACC(OR_G)	ACC(SS_F)	ACC(GH_F)	ACC(FD_F)	AUC	EER
MobNet	55.00	99.83	25.50	4.67	0.25	0.9462	0.1250
EffNetB4	78.00	100	79.08	56.67	32.00	0.9609	0.1258
Xception	64.00	100	42.33	37.75	1.25	0.9982	0.0225
ICPR2020	50.00	100	0.00	0.00	0.00	0.5969	0.4350
NeurIPS2023	47.99	77.33	5.58	27.33	23.00	0.4557	0.5183

B. Cross-category tests

As widely recognized in the deep learning domain, a crucial issue in the training and deployment of deep networks is the interpretability of the discriminative cues they learn. In order to have further insights on our detection scenario, we have performed experiments where models trained on data belonging to a certain category type are tested on the other category of the same dataset. In fact, the **occ** frames contain visual artifacts caused by face occlusion while those are not present in the **no-occ** frames. Table IV reports the results in two insightful settings: the first one explores the gap in terms of EER on **occ** when training on **occ** or **no-occ**; the second one assesses detection performance when a model trained on **occ** is tested on **no-occ** instead of **occ** data.

In both cases, we see a rather limited decrease in terms of EER when moving from an aligned to a non-aligned setting. In particular, it is shown that the **occ** data are well separated by both **occ**- and **no-occ**-trained models, which have not seen occlusion-based artifacts in training (top subtable). Also, a **occ**-trained model performs well also on **no-occ**, thus in absence of occlusion-based artifacts (bottom subtable). This suggests that the decisions of **occ**-trained network are only marginally based on actual occlusion artifacts, while they likely learn shared discriminative cues among the two frame categories.

Lastly, we investigate the discriminative features the network focuses on by applying **GradCAM++** [9]. For the sake of space, we report the results for the *MobNet* and *ICPR2020* models (yielding superior performance in Table III) tested on sample

swapped frames. Looking at the results, we can see that the *MobNet* activations vary over the test data, but they are often not concentrated on the occluded part where the visual artifacts are actually located. On the other hand, *ICPR2020* activations are consistently located in the face center, thus missing other visually relevant areas.

TABLE IV: EER values for different combinations of training/testing frame categories.

	FOWS		GOTCHA	
Test	**occ**			
Train	**occ**	**no-occ**	**occ**	**no-occ**
MobNet	0.00	0.04	0.00	0.00
EffNetB4	0.01	0.01	0.00	0.00
Xception	0.00	0.00	0.00	0.00
ICPR2020	0.01	0.20	0.00	0.00
NeurIPS2023	0.27	0.39	0.00	0.00

	FOWS		GOTCHA	
Test	**occ**	**no-occ**	**occ**	**no-occ**
Train	**occ**			
MobNet	0.00	0.0000	0.00	0.00
EffNetB4	0.01	0.0000	0.00	0.00
Xception	0.00	0.0000	0.00	0.00
ICPR2020	0.01	0.0000	0.00	0.00
NeurIPS2023	0.27	0.2067	0.00	0.00

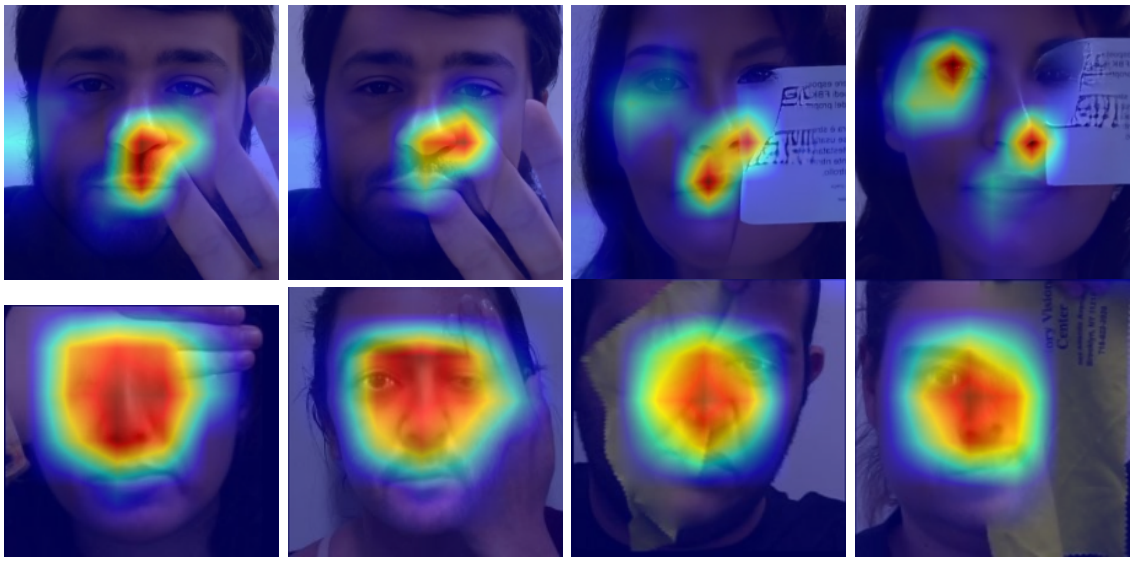


Fig. 4: GradCAM++ activation maps for *MobNet* trained on GOTCHA and tested on FOWS (top row), and for *ICPR2020* trained on FOWS and tested on GOTCHA (bottom row).

V. Discussions and Conclusions

We have performed an experimental analysis exploring the effectiveness of challenge-based detection for face swapping in video communication scenarios [16]. In those systems, the presence of visual artifacts supposedly introduced by swapping algorithms when dealing with face occlusions is leveraged. Although it is clear how this can substantially help human-based detection, it is conjectured in [16] that they can also improve automated detectors in separating original and manipulated videos, although providing evidence for a single dataset. We therefore developed the FOWS dataset, our own video data corpus containing face occlusions in genuine and manipulated videos, so as to extend the empirical evaluation of learning-based automated detectors in a multi-source setting. We trained different baseline and state-of-the-art detectors on both FOWS and GOTCHA datasets and performed a comparative analysis. The prominent finding is an overall lack of model generalization in the cross-dataset setting for all networks, thus confirming a known concern for automated detectors of video manipulation [4] [27]. In fact, models achieve excellent performance when trained and tested on the same dataset (even with limited training data as in FOWS), thus effectively picking up intra-dataset discriminative cues between classes. However, it is shown that those

do not transfer in the inter-dataset mode, despite some cases where the output scores of CNNs are fairly separated but strongly polarized towards zero. In fact, by studying performance when mixing data that contain and do not contain occlusion-based visual artifacts, it emerges that the models learn mostly dataset-specific statistical cues rather than actual occlusion-related cues.

In other words, our analysis reveals that in order to reliably exploit the advantage given by challenge-based visual artifacts, specialized automated approaches are needed. Assessing strategies for cross-dataset generalization and a broader pool of models proposed in the literature will be the subject of future work. Also, techniques for automatically guiding the networks towards the relevant face areas will be explored (such as the addition of an occlusion detection module), so that the detector can be activated only when an occlusion is happening and where related artifacts are located. Moreover, one-class detectors and anomaly detection frameworks also represent a promising direction, as they would reduce the dependency on the swapping algorithms used in training.

ACKNOWLEDGMENT

We thank the authors of [16] for sharing with us their data before the official release.

This work was partially supported by project SERICS (PE00000014) under the MUR National Recovery and Resilience Plan funded by the European Union - NextGenerationEU.

REFERENCES

[1] I. Amerini, M. Barni, S. Battiato, P. Bestagini, G. Boato, T. Sari Bonaventura, V. Bruni, R. Caldelli, F. De Natale, R. De Nicola, L. Guarnera, S. Mandelli, G. L. Marcialis, M. Micheletto, A. Montibeller, G. Orru', A. Ortis, P. Perazzo, G. Puglisi, D. Salvi, S. Tubaro, C. Melis Tonti, M. Villari, and D. Vitulano. Deepfake media forensics: State of the art and challenges ahead. *arXiv:2408.00388*, 2024.

[2] N. Bonettini, E. D. Cannas, S. Mandelli, L. Bondi, P. Bestagini, and S. Tubaro. Video face manipulation detection through ensemble of CNNs. In *International Conference on Pattern Recognition (ICPR)*, 2021. Available at https://github.com/polimi-ispl/icpr2020dfdc/tree/master.

[3] Renwang C., Xuanhong C., Bingbing N., and Yanhao G. SimSwap: An efficient framework for high fidelity face swapping. In *ACM International Conference on Multimedia*, 2020.

[4] J. P. Cardenuto, J. Yang, R. Padilha, R. Wan, D. Moreira, H. Li, S. Wang, F. Andaló, S. Marcel, and A. Rocha. The age of synthetic realities: Challenges and opportunities. *APSIPA Transactions on Signal and Information Processing*, 12(1), 2023.

[5] F. Chollet. Xception: Deep Learning with Depthwise Separable Convolutions . In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[6] ENISA. Remote ID Proofing. ENISA Report, March 2021.

[7] ENISA. Remote Identity Proofing – Attacks & Countermeasures. ENISA Report, January 2022.

[8] C. R. Gerstner and H. Farid. Detecting real-time deep-fake videos using active illumination. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2022.

[9] Jacob Gildenblat and contributors. Pytorch library for cam methods. 2021. Available at https://github.com/jacobgil/pytorch-grad-cam.

[10] Google. Mediapipe face detector python. Available at https://developers.google.com/mediapipe/solutions/vision/face_detector.

[11] A. Groshev, A. Maltseva, D. Chesakov, A. Kuznetsov, and D. Dimitrov. GHOST—a new face swap approach for image and video domains. *IEEE Access*, 2022.

[12] E. Haasnoot, Luuk J. Spreeuwers, and Raymond N. J. Veldhuis. Presentation attack detection and biometric recognition in a challenge-response formalism. *EURASIP Journal on Information Security*, (1):5, 2022.

[13] C. Li, L. Wang, S. Ji, X. Zhang, Z. Xi, S. Guo, and T. Wang. Seeing is Living? Rethinking the Security of Facial Liveness Verification in the Deepfake Era. In *USENIX Security 22*, pages 2673–2690, 2022.

[14] T. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár. Focal loss for dense object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(2):318–327, 2020.

[15] I. Loshchilov and F. Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

[16] G. Mittal, C. Hegde, and N. Memon. Gotcha: Real-time video deepfake detection via challenge-response. In *IEEE European Symposium on Security and Privacy (EuroS&P)*, 2024. Available at https://github.com/mittalgovind/GOTCHA-Deepfakes.

[17] Yuval N., Yosi K., and Tal H. FSGAN: Subject agnostic face swapping and reenactment. *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

[18] C. Pasquini, M. Pernpruner, G. Sciarretta, and S. Ranise. Towards a fine-grained threat model for video-based remote identity proofing. In *Machine Learning and Principles and Practice of Knowledge Discovery in Databases - International Workshops of ECML PKDD*, 2023.

[19] I. Perov, D. Gao, N. Chervoniy, K. Liu, S. Marangonda, C. Umé, Dpfks, C. Shift Facenheim, L. RP, J. Jiang, S. Zhang, P. Wu, B. Zhou, and W. Zhang. DeepFaceLab: Integrated, flexible and extensible face-swapping framework. *arXiv: 2005.05535*, 2021.

[20] M. M. Pic, G. Mahfoudi, A. Trabelsi, and J. Dugelay. *Face Manipulation Detection in Remote Operational Systems*. Springer International Publishing, 2022.

[21] R. Ramachandra and C. Busch. Presentation attack detection methods for face recognition systems: A comprehensive survey. *ACM Computing Surveys*, 50(1), 2017.

[22] F. Rosberg, E. E. Aksoy, F. Alonso-Fernandez, and C. Englund. FaceDancer: Pose- and occlusion-aware high fidelity face swapping. In *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2023.

[23] A. Rössler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Nießner. Faceforensics++: Learning to detect manipulated facial images. In *ICCV*, 2019.

[24] Xianyun S., Beibei D., Caiyong W., Bo P., and Jing D. Visual realism assessment for face-swap videos. *arXiv: 2302.00918*, 2023.

[25] R. Tolosana, R. Vera-Rodríguez, J. Fierrez, A. Morales, and J. Ortega-Garcia. Deepfakes and beyond: A survey of face manipulation and fake detection. *arXiv: 2001.00179*, 2020.

[26] Danial Samadi Vahdati, Tai Duc Nguyen, and Matthew C. Stamm. Defending low-bandwidth talking head videoconferencing systems from real-time puppeteering attacks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2023.

[27] L. Verdoliva. Media forensics and deepfakes: An overview. *IEEE Journal of Selected Topics in Signal Processing*, 14(5):910–932, 2020.

[28] Z. Yan, Y. Zhang, Xi. Yuan, S. Lyu, and B. Wu. Deepfakebench: A comprehensive benchmark of deepfake detection. In *Advances in Neural Information Processing Systems*, volume 36, pages 4534–4565, 2023. Available at https://github.com/SCLBD/DeepfakeBench.