Deepfake detection by exploiting surface anomalies: the SurFake approach

Andrea Ciamarra[1,4], Roberto Caldelli[3,4], Federico Becattini[2], Lorenzo Seidenari[1], Alberto Del Bimbo[1]

[1]University of Florence, Florence, Italy, [2]University of Siena, Siena, Italy,
[3]CNIT, Florence, Italy, [4]Universitas Mercatorum, Rome, Italy

{andrea.ciamarra, lorenzo.seidenari, alberto.delbimbo}@unifi.it,
federico.becattini@unisi.it, roberto.caldelli@cnit.it

Abstract

The ever-increasing use of synthetically generated content in different sectors of our everyday life, one for all media information, poses a strong need for deepfake detection tools in order to avoid the proliferation of altered messages. The process to identify manipulated content, in particular images and videos, is basically performed by looking for the presence of some inconsistencies and/or anomalies specifically due to the fake generation process. Different techniques exist in the scientific literature that exploit diverse ad-hoc features in order to highlight possible modifications. In this paper, we propose to investigate how deepfake creation can impact on the characteristics that the whole scene had at the time of the acquisition. In particular, when an image (video) is captured the overall geometry of the scene (e.g. surfaces) and the acquisition process (e.g. illumination) determine a univocal environment that is directly represented by the image pixel values; all these intrinsic relations are possibly changed by the deepfake generation process. By resorting to the analysis of the characteristics of the surfaces depicted in the image it is possible to obtain a descriptor usable to train a CNN for deepfake detection: we refer to such an approach as SurFake. Experimental results carried out on the FF++ dataset for different kinds of deepfake forgeries and diverse deep learning models confirm that such a feature can be adopted to discriminate between pristine and altered images; furthermore, experiments witness that it can also be combined with visual data to provide a certain improvement in terms of detection accuracy.

1. Introduction

With the increasing of false information spreading all over the media, nowadays, trust in digital content is potentially compromised by the easiness of creating fabricated facts. Information can further undergo multiple modifications before reaching a potential user. The latest advancements of AI, especially for image and video manipulation, are fostering more and more the possibility to easily change the meaning of the information to convey, due to the fact that media content is likely to be exposed to variations of different nature. Among the possible media manipulation approaches, Deepfakes are a very recent class of methods that can generate synthetic human images. Despite having been used with astonishing results for movie production in Hollywood, Deepfakes can also be easily used for malicious purposes, such as crafting highly realistic fake propaganda. Deepfake creation typically involves the use of deep learning to recreate some person imagery. Specifically deep networks learn how to transfer or reenact facial expression as well as how to generate a proper imitation of a person's voice and inflection. Nowadays, several techniques can create real-looking content easily, by using deep generative models, such as GAN-style architectures [3, 9, 11] and diffusion probabilistic models (DPMs) [15]. Deepfake can be applied to different types of media,

Figure 1. An example of surface anomalies found in fake images. From left to right: the RGB (Red-Green-Blue) face, our proposed GSD (Global Surface Descriptor) feature and the logarithm of the GSD, used here for sake of visualization to highlight the artifacts introduced by the manipulation.

from image to video, but also fake audio can be created, e.g. through text-to-speech [17], by typing a new text, or by voice swapping [28]. Deepfake for image and video is mainly done by tampering with parts of the scene, for instance, the faces of subjects present in the media. Various techniques have been designed to alter faces, some of them regard video applications, e.g. lip syncing [29, 35] where the audio is employed to reconstruct the mouth movements over the video frames. Other general-purpose face manipulation techniques are about the visual alteration of the content, by changing the expressions or moving the face from a source image to the target one.

In this paper, we focus on detecting face manipulations in images. Face manipulations [44] can be basically summarized in two different categories: reenactment and swapping. Facial reenactment deals with manipulating certain facial attributes or reenact faces with deep learning methods while maintaining the identity unchanged. Face swapping [8], instead, aims at replacing the face of a person in the reference image with the same facial shape and features of another target subject by realistically changing the identity. According to this, it is straightforward to understand that deepfake detection is an urgent task, which prevents disinformation and avoids the diffusion of media showing people saying or doing things they never said or did. The fundamental idea behind deepfake detection consists in the fact that a neural network during the process of generating a fake content should leave a sort of trace that is embedded as a fingerprint over the manipulated image (video). Most of the existing approaches for deepfake detection try to recover this hidden pattern to reveal false contents and to do that they generally resort to the analysis of frames at pixel level (RGB raw data). However, not only RGB-level inconsistencies can be detected as anomalies in the visual space, in fact many fine details can be affected by the forgery without compromising the visual perception of the image itself. More precisely, the camera acquisition process is itself a signature that is incorporated into the image. We argue that such information about the acquisition moment could be altered by deepfakes and this could be exploited for the detection task. Such information relates to an ensemble of specific characteristics of the scene, for instance, the external illumination source, including lighting, shadows and reflections, which all impact on the surfaces present in the environment and on the objects at the time of image acquisition. Other relevant details regard the face pose, but even the camera parameters may be somehow embedded into the image and strictly related to the image capturing, e.g. lens distortions and intrinsic noises. In the literature, some works [23, 34] tackled the deepfake detection problem from different angles, by leveraging geometrical aspects of facial landmarks, in which inconsistent or highly synthetic patterns are found in generated fakes by different forgery techniques.

The proposed approach leverages on the analysis of the features of the framed scene that are determined by the overall geometry of the scene itself (e.g. surfaces) and by the original image acquisition process (e.g. illumination, camera orientation). Different from other research works focusing on individuating fakes by detecting specific patterns in terms of depth map or face motion, we specifically deal with the surfaces present within the acquired scene by exploiting the modifications induced to the surface normals and determined by the deepfake alteration. A general visual example of this is provided in Figure 1 where it can be appreciated how a modification, just on the mouth of the woman, can also determine some slight global variations on the other parts of the global surface descriptor (GSD) image. In summary, the main contributions are listed hereafter:

(i) we propose to utilize surface geometry features of the acquired scene to highlight inconsistent patterns revealing fake images;

(ii) we study and evaluate, in which extent, such features can constitute by themselves an effective mean to discriminate between pristine and fake contents;

(iii) we conduct experiments on different kinds of forgeries and network architectures to verify that the new proposed surface-based feature can be advantageously combined with RGB frames to get an improvement in terms of accuracy performance.

The paper is organized as it follows: after this introductory section, Section 2 describes main related works while Section 3 presents the proposed method. Section 4 is dedicated to the experimental results and Section 5 draws conclusions providing possible future works.

2. Related Works

DeepFake Detection is a recent problem raised to recognise real or tampered data, also in vision tasks, which is typically addressed as a binary classification problem. Nowadays visual content is an informative media that can be manipulated, thanks to the usage of recent generative models [3, 9, 11, 15]. In particular, the possibility of manipulating human faces has found a lot of interest both for entertainment and for malicious purposes. Several manipulation techniques can be used, either replacing faces to match the one of another subject or by simply altering facial expressions. Such manipulations are obtained, e.g. via CNNs [19], conditional GANs [27] or based on facial landmark alignment [8]. A plethora of implementations are also available [25]. Existing methods [26] for deepfake detection are designed to directly process entire videos or single frames, so to discover whether faces have been manipulated. Several

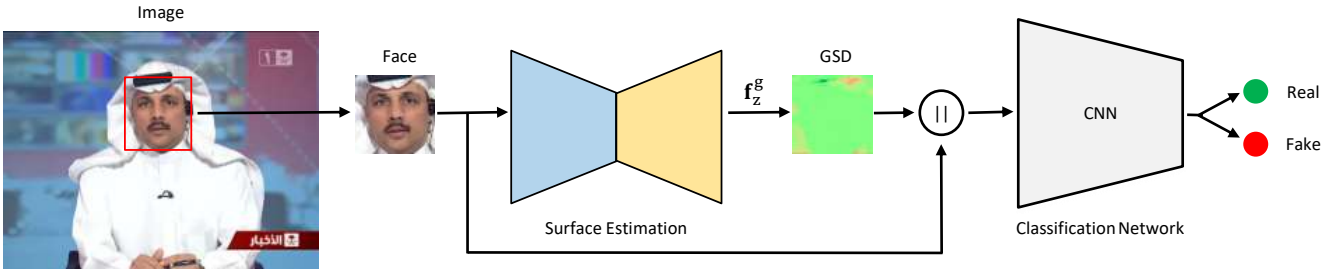


Figure 2. Pipeline of SurFake for deepfake detection. After extracting the face crop from the image, we generate its Global Surface Descriptor (GSD) through UpRightNet [40] and we scale the generated vector values in $[0, 255]$ to obtain an RGB image. Then, we concatenate the face crop and the GSD feature at the last channel and we pass it in input to a classifier. Finally, we train the classifier to distinguish whether the content is real or fake.

works exploit handcrafted features from artifacts and inconsistencies of the fake generation process. Xian et al. [41] proposed to preprocess images to remove low level noise cues of GAN images, and so a forensic model is forced to learn more intrinsic features. This method allows better generalization capability than previous deepfake methods [5, 42]. Amerini et al. [4] leveraged the optical flow in order to look at motion discrepancies, which are found across synthetically generated frames, and finally to classify them as original or deceptive. Li et al. [21] utilized an advanced architecture named HRNet to detect the blending boundary of Deepfakes manipulated images. Guo et al. [13] introduced a CNN model named SCnet to detect Glow-based facial forgery by learning high-level features through a hierarchical convolutional block. Zhao et al. [45] proposed to look at source feature inconsistency within the forged image with the hypothesis that a pristine image should contain the same source features across locations. Instead of learning GAN fingerprints on fakes [43] or visual self-inconsistencies via recorded photo metadata [16], Maiano et al. [24] exploited depth inconsistencies located inside tampered face images to detect manipulations. Other approaches, instead, utilize recurrent networks, e.g. RNNs or LSTMs, to look at visual artifacts within single video frames or temporal inconsistency across frames. Sabir et al. [31] leveraged the use of spatio-temporal features of video streams to detect deepfakes, as temporal coherence is not innate in deepfake generation. Güera et al. [12] extracted frame-level features with a CNN and fed them into an LSTM to create temporal sequence descriptors, which are finally trained to be classified as real or fake. Different methods have also been proposed to exploit visual or behavioral inconsistencies, hardly removable when generating fakes. Based on the fact that a person in deepfakes has a lot less frequent blinking than that in untampered videos, Li et al. [22] proposed to crop eye areas of video frames on which features are extracted and then fed into an LSTM, and so to predict the probability of eye open or close. Caldelli et al. [7] proposed to leverage the optical flow to account for facial motion since artificial parts of the face contain some intrinsic dissimilarities with respect to natural expressions. Becattini et al. [6] found discrepancies in face alignment by looking at the head orientation, i.e. roll, pitch and yaw. Liang et al. [23] extract geometry facial features as peculiarities around the landmark to be discriminated between pristine and manipulated regions (e.g. spatial relationship, appearance, shape). Such features are fed into a CNN-LSTM network, and, finally, a decoder learns to map low-level features to pixel-wise manipulation localizations along with a softmax classifier to detect real and fake. Sun et al. [34] exposed abnormal facial movement patterns and time discontinuities by means of precise geometric features of facial landmarks, by making a proper calibration step, which is performed through a Lucas-Kanade operation to track landmark points and merge the detection and the prediction using a Kalman filter. Differently, we do not make use of segmentation face model [23], facial landmarks, or calibration steps [34]. We consider the Global Surface Descriptor (GSD) of the face, which is a feature describing the geometry of the face. However, in contrast to [6], which leverages head pose estimation with respect to the camera, we use a description of surface orientations at a pixel level by characterizing surface normals in a global up-right reference system that is inherently obtained along with the camera orientation.

3. The proposed method

In this section we will introduce the proposed method, named *SurFake*, which exploits inconsistencies in the features of surfaces belonging to the acquired scene to perform deepfake detection. Our pipeline is organized into three steps as depicted in Figure 2: first, we perform face detection on each video frame using dlib [18] obtaining a face crop with a fixed resolution of 224×224. Secondly, we run a pretrained *UpRightNet* [40] on face crops to extract features (see Section 3.1 for details) in order to get the *Global Surface Descriptor (GSD)*. Finally, the concatenation of the RGB face crop and the GSD feature which constitutes a 6-

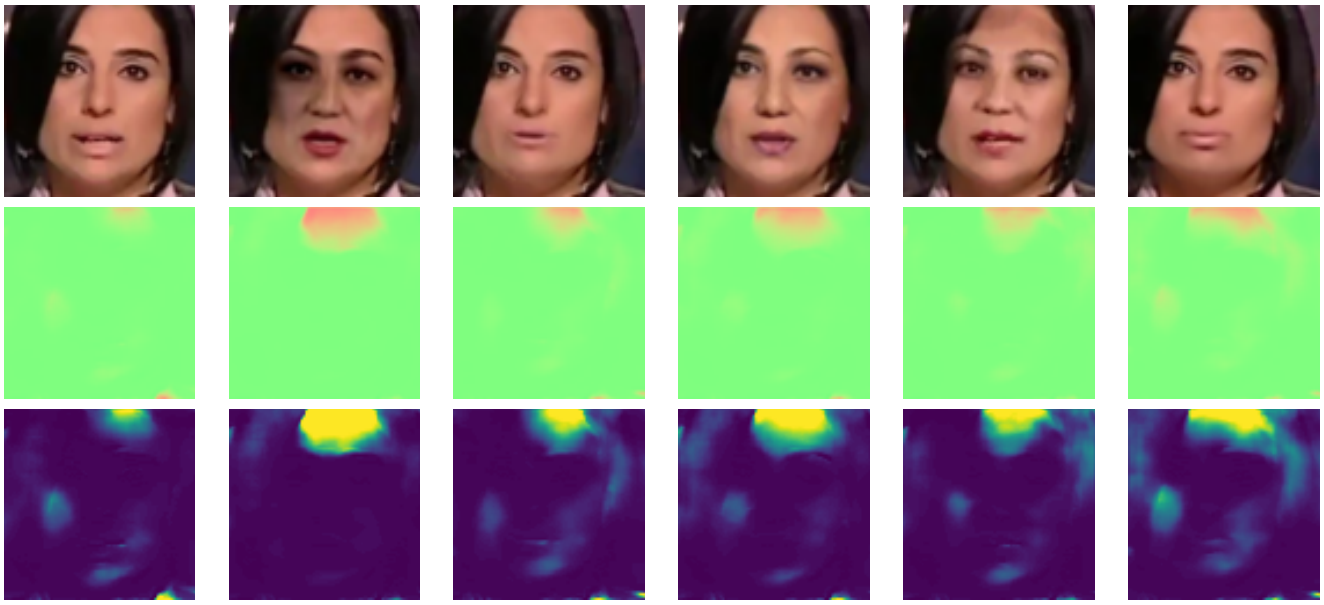


Figure 3. Sample frames (first row) and the corresponding Global Surface Descriptors (second row) and $\log(GSD)$ (third row) for each of the 5 different forgeries in FF++, from left to right: Real, DF, F2F, FSH, FS, NT [30]. The third row highlights how GSD is sensitive to forgeries.

channel tensor is used to train a deep convolutional neural network to perform the binary classification task required to detect deepfakes.

3.1. The Global Surface Descriptor (GSD)

In order to extract subtle scene details, we face deepfake detection from a new perspective. In contrast to looking for inconsistencies in the visual perception domain, we highlight anomalies by looking into the geometrical aspects related to the camera acquisition process. Such aspects permanently mark low-level peculiarities of the image without visually affecting the content. Therefore, tampered images may contain fine-grained distortions which do not stand in the visible space. Typically, deepfakes depict a person in the foreground whose entire face (or some of its parts) has been tampered with. The idea behind our approach is to address forgery detection by focusing on the surface geometry of the face. To do that, we employ a deep learning model named *UpRightNet* [40] to estimate such geometrical characteristics presented by the oval surface of the face but also by other different surfaces such as the chin, the nose, the eye sockets or any headgear. UpRightNet is a neural network that learns to estimate the 2DoF camera orientation, i.e. roll and pitch, from a single RGB image using intermediate representations, called surface frames, estimated from both the local camera and the global up-right coordinate systems. Let us suppose to predict the per-pixel surface normals of an indoor image in the camera perspective. Surface normals on the ground and other horizontal surfaces point in the same direction as the camera up vector, instead, walls and other vertical surfaces are perpendicular to the up vector. Thus, camera orientation can be estimated as finding the vector which is most parallel to the ground normals and most perpendicular to the wall normals. UpRightNet solved the camera orientation problem by computing the rotation that best aligns the two estimated representations of the surface frames. A surface geometry frame $\mathbf{F}(i)$ is estimated from each pixel i, as a 3×3 matrix of mutually orthogonal unit vectors, that is normals, tangents and bitangents respectively: $\mathbf{F}(i) = [\mathbf{n}(i), \mathbf{t}(i), \mathbf{b}(i)]$ with $\mathbf{n}(i), \mathbf{t}(i), \mathbf{b}(i) \in \mathbb{R}^3$. UpRightNet estimates two surface frames, one in the local camera coordinate system, $\mathbf{F}^c(i)$, and one in the global up-right coordinate system, $\mathbf{F}^g(i)$. In order to predict roll and pitch of the camera, UpRightNet aligns the up-vector in the two representations, by using the z-component of $\mathbf{F}^g(i)$, i.e. $\mathbf{f}_z^g(i) \in \mathbb{R}^3$. Such alignment is computed by learning weights to solve a constrained least squared problem using ground-truth camera orientations. Due to the fact that the feature $\mathbf{f}_z^g(i)$ substantially provides a 3-channel global description of the surfaces belonging to the acquisition scene, we have considered that it could be a good candidate to possibly give evidence of a manipulation. Such a feature, denominated *Global Surface Descriptor (GSD)*, will be analysed more in depth within the next sub-section.

3.2. Analysis of Deep Geometric Representations

This section presents how UpRightNet features can be useful in the context of deepfake detection; in Figure 3

we depict an example of a pristine face crop along with various face manipulations methods implemented in FaceForensics++ [30] (first row). We also show, in the second row the corresponding GSD feature extracted by UpRightNet after passing the face image in input. Finally, in the last row, we enhance the colorization of GSD in order to visually highlight how the GSD feature could be useful to detect anomalies by depicting the logarithm of the feature. In fact, upon a visual inspection, the GSD features may appear similar to each other for different faces, even appearing uniform regardless of the content. This is due to the fact that the faces are typically framed frontally in the global upright coordinate system. Similarly to global representations estimated for indoor images, the light green pixels stand for surfaces whose normals are perpendicular to the up-ward vector, e.g. the walls in a room, just like most of the face pixels too. There are also other parts of the image which are sometimes colored with shades of red and dark green, that encode surfaces parallel to the ground of a room. In our face domain, pixels with normals parallel to the ground are located at the top and at the bottom of the image. Therefore, the geometry estimated for face images is quite consistent with indoor environments. However, we would like to demonstrate how this geometry, i.e. our proposed GSD feature, is useful in our task. To this aim, we here highlight anomalies, by calculating the logarithm of each image pixel on the first of the 3 channels and we plot the results in the third row in Figure 3, for each face manipulations.
First of all, we generally observe that some artifacts are added or exaggerated at the top of the image for all manipulations, which explains that any alteration produces unavoidable patterns (see the yellow color and the surrounding parts located at the top of the images in the third row, i.e. the forehead). In the face swapping approach the alterations are mainly reported around the outer facial landmarks for FS and FSH. Interestingly, the latter is also a more recent learning-based face replacing method than the well-known DF, where the face looks almost flat. As soon as facial expressions are tampered (i.e. by performing either F2F or NT) and small parts are faked, e.g. mouth or eyes, other relevant regions are also compromised, e.g. cheeks and hair. Although the GSD feature may look scarsely informative at once, we argue that such subtle details can come up more visible for a neural network trained to detect these synthetic patterns, in the sense of anomalies in the geometric estimation of the face.

4. Experimental results

In this section, we will present the experimental results carried out in order to verify the effectiveness of the presented approach and, in particular, of the GSD feature.

4.1. Implementation Details

Dataset We conduct experiments on FaceForensics++ (FF++) [30], one of the most widely used datasets for deepfake detection. It has collected 1000 original real videos from the internet and for each video 5 different forged versions are generated. This dataset comprises two types of face manipulation techniques: face swapping, in which the face identity in the source image is replaced with the target one, and face reenactment, in which the facial expression in the source image is altered from the one in a target image, while maintaining the identity. FaceForensics++ includes three swapping methods, DeepFakes (DF) [1], FaceShifter (FSH) [20] and FaceSwap (FS) [2], and two reenactment methods, i.e. Face2Face (F2F) [38] and NeuralTextures (NT) [37]. In particular, two of these manipulations are computer graphics-based approaches (Face2Face and FaceSwap) while the other three are learning-based approaches (DeepFakes, FaceShifter and NeuralTextures). Specifically, NeuralTextures operates by only altering the mouth region, i.e. eye parts are unchanged while FaceShifter is a recent learning-based approach. It generates high fidelity identity preserving face swap results being able, differently from the other two face swapping methods, to deal with facial occlusions using a double synthesis stage.

Overall, there are 1000 forged videos for each face manipulation, for a total of 5000. The image resolutions vary across videos, from 272×480 up to 1920×1080. FaceForensics++ provides raw videos, and two versions compressed using the H.264 codec, i.e. light compression (c23), which is nearly lossless, and heavy compression (c40). For all our experiments we choose the c23 videos, which is indicated from the dataset authors as HQ (high quality).

Data preparation To reduce the computational burden and exclude redundancies, we sample one frame out of ten in each video sequence. For all our experiments we follow the 72:14:14 data split, respectively for train, validation and test sets, as indicated in [30], i.e. 720, 140 and 140 videos. Following [30], 224×224 face crops are obtained by first extracting a 1.3-factor enlarged crop centered at the detected face in the input image and then scaling it to the fixed resolution. We consider face crops of such a resolution as it is a standard image dimension that can be processed into most of the existing architectures. Since UpRightNet gets input images of 288×384 and generates outputs at the same resolution, we adapt face crops to such dimension in order to extract the Global Surface Descriptor (GSD) and then we rescale to 224×224, as done in [40]. Among the UpRightNet pretrained weights on InteriorNet and ScanNet (i.e. two indoor image datasets), we chose the former one since this model estimates a more coherent representation

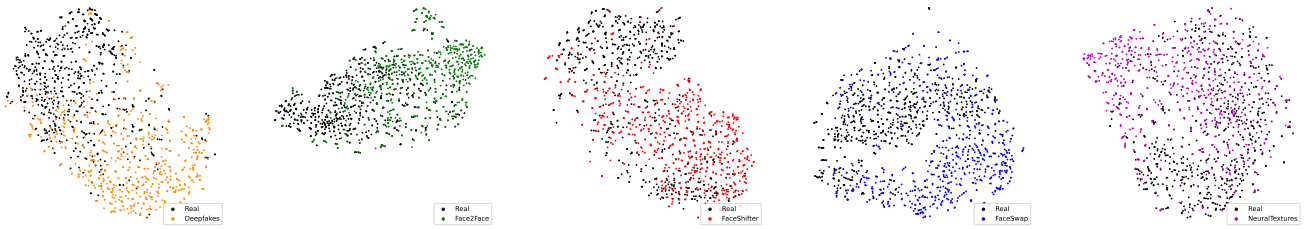


Figure 4. T-SNE [39] plots of the GSD feature activations for real and fake samples of the test set for each of the different forgeries (MobilNetV2 architecture). Only a reduced number of samples is plotted for the sake of visibility.

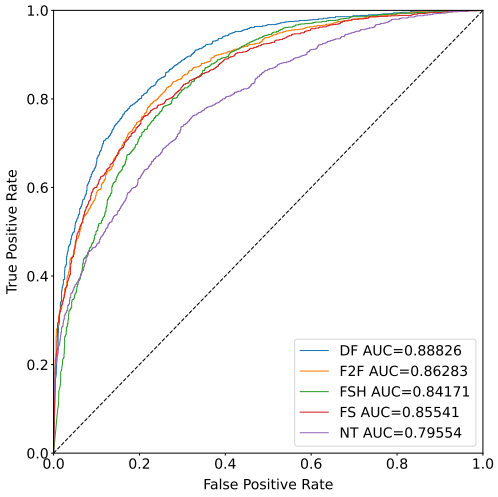


Figure 5. ROC Curve of GSD features for real and fake using MobileNetV2 as classifier. We also reported the Area Under Curve (AUC) for each forgery.

of GSD than the latter, in accordance with the true geometry in the face domain in terms of normals perpendicular or parallel to the global upright coordinate system (see Section 3.1).

Architectures In order to classify images as real or fake we train 4 different well-known and standard architectures: ResNet50 [14], MobileNetV2 [32], EfficientNet-B0 [36] and Xception [10] with pretrained weights on ImageNet. Since we are using a neural network pretrained on classical RGB images from ImageNet, we modify the first convolutional layer, which gets the input, to handle a different type of data, i.e. the 3-band GSD feature and also a different number of input channels, as our approach is employing a total of 6, i.e. the first 3 channels for the RGB concatenated to the second 3 additional ones from GSD. Besides, to make the training more stable and to allow the model to converge faster, we make a proper weight initialization. By following [24], we calculate the average of the three original input channels from the pretrained model and we replace this initialization for each and every channel of GSD. We chose this initialization for all our experiments.

	FF++ forgeries					
Architectures	DF	F2F	FSH	FS	NT	Avg
ResNet50	0.766	0.725	0.735	0.690	0.674	0.718
MobileNetV2	0.800	0.773	0.756	0.764	0.713	0.761
EfficientNet-B0	0.802	0.773	0.761	0.754	0.707	0.759
Xception	0.796	0.759	0.759	0.747	0.726	0.757

Table 1. Performance in terms of accuracy for the GSD feature on the test set with respect to the different network architectures.

Because each architecture is pretrained on ImageNet, we scale values in $[0, 1]$ and then we normalize with mean $[0.485, 0.456, 0.406]$ and standard deviation $[0.229, 0.224, 0.225]$ for ResNet50, MobileNetV2 and EfficientNet-B0 respectively. For Xception, we use the Pytorch implementation and the pretrained ImageNet weights from [33]. Since Xception accepts inputs at 299×299, we upscale our patches to fit that resolution, we scale values in $[0, 1]$ and we normalize with mean and standard deviation both set to $[0.5, 0.5, 0.5]$. Note that we apply scaling and normalization on the input values for RGB and GSD separately.

Training setting We implement SurFake in Pytorch. We model the deepfake detection as a binary classification problem and we train each classification network on an NVIDIA TITAN GTX. Specifically, we use a standard cross entropy loss with two classes, real and fake, for 30 epochs and batch size 32. We utilize SGD as optimizer with momentum 0.9, weight decay 0.0001 and learning rate 0.001.

4.2. Analysis of the proposed GSD feature performance

In this section, experimental results to evaluate the effectiveness of the proposed GSD feature will be presented. To better understand the capacity to provide distinctiveness between real and deepfake images, we have considered the activations obtained at the final layer of SurFake, just before getting the output decision step. Therefore, we train MobileNetV2 to detect each face manipulation by using only

Architectures	FF++ forgeries											
	DF		F2F		FSH		FS		NT		Average	
	RGB	RGB+GSD	RGB	RGB+GSD	RGB	RGB+GSD	RGB	RGB+GSD	RGB	RGB+GSD	RGB	RGB+GSD
ResNet50	0.981	**0.984**	0.988	**0.989**	**0.980**	0.971	0.985	**0.986**	0.938	**0.947**	0.974	**0.975**
MobileNetV2	0.987	**0.992**	**0.990**	0.989	0.985	**0.989**	**0.990**	**0.990**	0.958	**0.966**	0.982	**0.985**
EfficientNet-B0	0.989	**0.992**	0.983	**0.986**	**0.982**	**0.982**	0.984	**0.985**	0.955	**0.958**	0.978	**0.981**
Xception	**0.976**	0.975	**0.979**	**0.979**	**0.976**	**0.976**	0.977	**0.978**	**0.939**	**0.939**	**0.969**	**0.969**

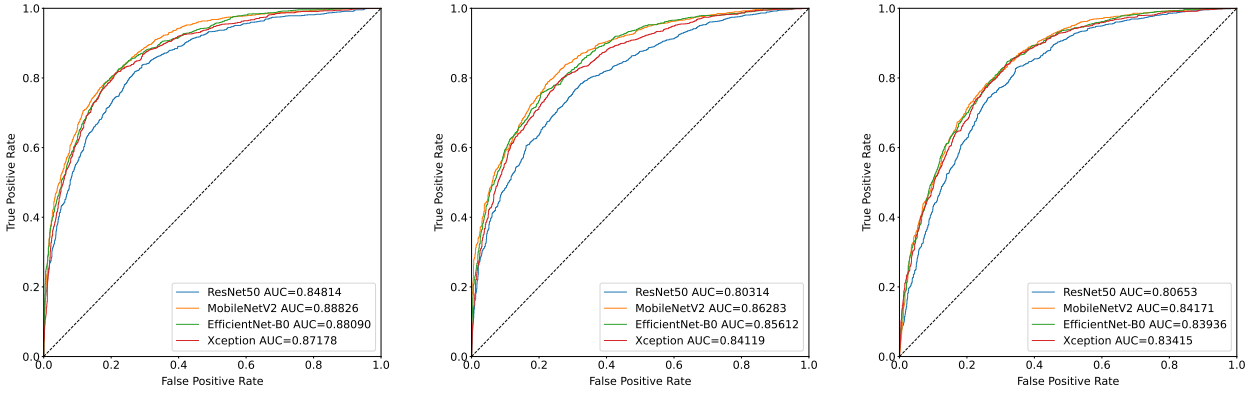
Table 2. Performance in terms of accuracy on the test set for the different architectures with respect to the FF++ forgeries for RGB and RGB+GSD cases.

our GSD feature as input without RGB. MobileNetV2 contains the initial fully convolution layer with 32 filters, followed by 19 residual bottleneck layers and ends with a linear layer with 1280 dimensional feature. We then plot these activations, by resorting to T-SNE [39]. As it can be seen in Figure 4 for all the 5 manipulation techniques of the FF++ dataset, it is possible to appreciate a certain separation between real samples (black dots) and fake ones (colored dots) which is coherent for all the different cases. Even though our proposed GSD features between real and fake are often uniform (see Figure 3), subtle differences can be perceived using a neural network, while, instead, being almost invisible to the human eye. The depicted T-SNE in Figure 4 clearly demonstrates how relevant these GSD patterns are (similar representations can be obtained with other network architectures). Although we are processing patches that describe surfaces of faces rather than canonical RGB face images, a significant amount of test samples have been correctly separated in the projection space. We also plot the ROC curves in the test set for all the face manipulations detected using MobileNetV2, in Figure 5. We deduce that in all the forgery techniques the AUC (Area Under Curve) is around 0.85, which is quite interesting considering the scarce visible information carried on this feature.

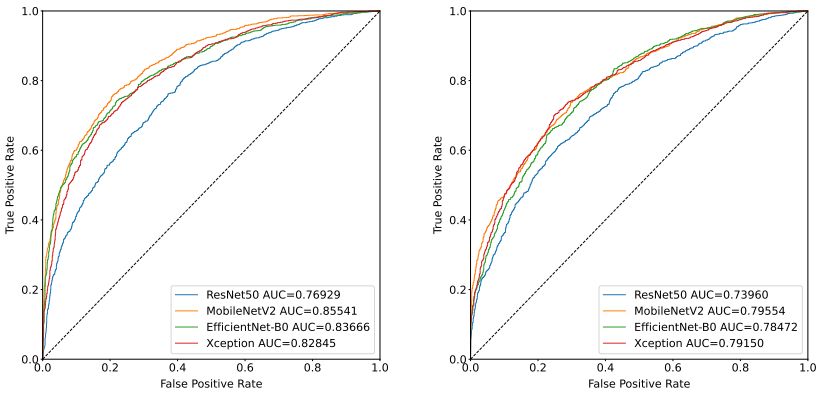
Similarly, we have tried to make a quantitative evaluation of such a phenomenon and we have computed the accuracy values for all the five different distortions. To do that, we evaluate our approach, still using GSD features as unique input to the classification network, and we report our results for all the 4 architectures. As listed in Table 1, we can notice that in each case and, above all, coherently for different kinds of network architectures, an average accuracy around 0.75 can be globally achieved. We observe that GSD exhibits well distinctiveness in most of the manipulations for all the architectures with high accuracy, e.g. DeepFakes (DF), Face2Face (F2F) and FaceShifter (FSH) are well-detected. Performance on NeuralTextures (NT) are lower than the other manipulations and possibly this is due to the fact that NT deals with facial reenactment performed just around the mouth region [30]. Additionally, we report the ROC curves for each forgery of all the architectures in Figure 6, which highlights the efficacy of the GSD features in most cases, as the average Area Under Curve is above 0.80, except for NT which gets an average of 0.77. We can also observe that EfficientNet-B0 and MobileNetV2 report higher AUC in most of the forgeries.

4.3. Composing GSD with RGB frames

Hereafter, we will present the results obtained by composing the 3-channel GSD with the RGB frames that are usually adopted as primary source of information in most deepfake detection methods. This has been done in order to understand if the proposed GSD feature is able to provide an improvement in deepfake detection, thanks to the fact that it takes into account geometrical components related to the acquisition scene. In this case the diverse network architectures have been trained by receiving as input a 6-channel tensor composed of 3 RGB bands concatenated with the 3 GSD channels. The achieved performances in terms of detection accuracy are listed in Table 2. As it can be seen, by looking at the last column of the table, a general increment is registered on average. Due to the fact that accuracy values are already quite high such improvement is rather limited but, what is interesting is that it is consistent for all the five forgeries and coherent for all the different network architectures that we considered. In particular, if we look at the NT case that usually appears to be more difficult to treat, it is possible to appreciate that an overall trend of increment is achieved for all the four networks. It is worth to point out that for the Xception model a substantial similar behavior is registered and the GSD feature does not seem to bring a relevant advancement. Since Xception gets input images of 299×299 but our original patches are of 224×224, both RGB and GSD have to be upscaled and fit them to the bigger resolution. That potentially adds interpolation artifacts. Indeed, ResNet50, that can directly get as inputs the original patches (as well as the other selected architectures), even though it has a different architecture from Xception but with comparable number of trainable parameters and performance reported in ImageNet, obtains an overall improvement when RGB is concatenated along with GSD. EfficientNet-B0, still with a different architecture but with similar performance and with one fifth of the trainable parameters than ResNet50 and Xception, got more improvement, with an average accuracy across all forgery manipulations of 0.981 (i.e. $+0.3\%$ using the percentage notation). Overall, we notice that our approach employing either EfficientNet-B0 or MobileNetV2 gets more



(a) DeepFakes (DF) (b) Face2Face (F2F) (c) FaceShifter (FSH)



(d) FaceSwap (FS) (e) NeuralTextures (NT)

Figure 6. Performance in terms of ROC curves on test set of the GSD features for each forgery in all the 4 architectures. We also report the Area Under Curve (AUC) in the legends.

performance improvement. This is also confirmed, by looking at the ROC curves in Figure 6, where our approach is only trained with the GSD features. These two architectures show a superior trend of the ROC curves with respect to others, and with the corresponding Area Under Curve values highest among the manipulations. Such behavior demonstrates that our proposed GSD feature introduced in a classification network can benefit the detection of deepfakes.

5. Conclusions

In this paper we proposed a novel deepfake approach named SurFake able to detect face manipulations at frame level. To do that, we introduced the use of Global Surface Descriptor (GSD) as feature that accounts for the camera acquisition process which marks permanently the image. In particular, we exploited the characteristics of the surfaces in which pixels belonging to horizontal or vertical areas of the image have a proper direction and intensity with respect to a global coordinate system. We tested SurFake with 4 different architectures on FaceForensics++, which contains 5 different face manipulations. We demonstrated that our proposed GSD features alone allow a classifier to reach around 75% of accuracy on average; furthermore, we tested our proposed pipeline by using RGB frames and GSD together as input and we got an overall improvement, though limited, for all the diverse face manipulations.

As future works, we consider to deal with larger cropped patches in order to possibly improve the effectiveness of GSD and also to make some data augmentations, e.g. random crop. We will investigate other geometric information estimated by UpRightNet methodology, e.g. the local surface geometry which is directly tied to the local coordinate system. Finally, we will carry out further experiments on other deepfake datasets.

Acknowledgements This work was partially supported by the H2020 project AI4Media (GA n. 951911) and by the project SERICS (PE00000014) under the NRRP MUR program funded by the EU - NGEU.

References

[1] DeepFakes github. https://github.com/deepfakes/faceswap. 5

[2] FaceSwap. https://github.com/MarekKowalski/FaceSwap/. 5

[3] Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2stylegan: How to embed images into the stylegan latent space? In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4432–4441, 2019. 1, 2

[4] Irene Amerini, Leonardo Galteri, Roberto Caldelli, and Alberto Del Bimbo. Deepfake video detection through optical flow based CNN. In *Proceedings of the IEEE/CVF international conference on computer vision workshops*, 2019. 3

[5] Belhassen Bayar and Matthew C Stamm. A deep learning approach to universal image manipulation detection using a new convolutional layer. In *Proceedings of the 4th ACM workshop on information hiding and multimedia security*, pages 5–10, 2016. 3

[6] Federico Becattini, Carmen Bisogni, Vincenzo Loia, Chiara Pero, and Fei Hao. Head pose estimation patterns as deepfake detectors. *ACM Trans. on Multimedia Computing, Commun. and Appl.*, 2023. 3

[7] Roberto Caldelli, Leonardo Galteri, Irene Amerini, and Alberto Del Bimbo. Optical flow based CNN for detection of unlearnt deepfake manipulations. *Pattern Recognition Letters*, 146:31–37, 2021. 3

[8] Dongyue Chen, Qiusheng Chen, Jianjun Wu, Xiaosheng Yu, and Jia Tong. Face swapping: realistic image synthesis based on facial landmarks alignment. *Mathematical Problems in Engineering*, 2019. 2

[9] Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8789–8797, 2018. 1, 2

[10] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1251–1258, 2017. 6

[11] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014. 1, 2

[12] David Güera and Edward J Delp. Deepfake video detection using recurrent neural networks. In *15th IEEE international conference on advanced video and signal based surveillance (AVSS)*, pages 1–6, 2018. 3

[13] Zhiqing Guo, Lipin Hu, Ming Xia, and Gaobo Yang. Blind detection of glow-based facial forgery. *Multimedia Tools and Applications*, 80:7687–7710, 2021. 3

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proc. of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 6

[15] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 1, 2

[16] Minyoung Huh, Andrew Liu, Andrew Owens, and Alexei A Efros. Fighting fake news: Image splice detection via learned self-consistency. In *Proceedings of the European conference on computer vision (ECCV)*, pages 101–117, 2018. 3

[17] Jan Kietzmann, Linda W Lee, Ian P McCarthy, and Tim C Kietzmann. Deepfakes: Trick or treat? *Business Horizons*, 63(2):135–146, 2020. 2

[18] Davis E King. Dlib-ml: A machine learning toolkit. *The Journal of Machine Learning Research*, 10:1755–1758, 2009. 3

[19] Iryna Korshunova, Wenzhe Shi, Joni Dambre, and Lucas Theis. Fast face-swap using convolutional neural networks. In *Proceedings of the IEEE international conference on computer vision*, pages 3677–3685, 2017. 2

[20] Lingzhi Li, Jianmin Bao, Hao Yang, Dong Chen, and Fang Wen. Advancing high fidelity identity swapping for forgery detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5074–5083, 2020. 5

[21] Lingzhi Li, Jianmin Bao, Ting Zhang, Hao Yang, Dong Chen, Fang Wen, and Baining Guo. Face x-ray for more general face forgery detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5001–5010, 2020. 3

[22] Yuezun Li, Ming-Ching Chang, and Siwei Lyu. In ictu oculi: Exposing ai created fake videos by detecting eye blinking. In *IEEE Internat. workshop on information forensics and security*, pages 1–7, 2018. 3

[23] Peifeng Liang, Gang Liu, Zenggang Xiong, Honghui Fan, Hongjin Zhu, and Xuemin Zhang. A facial geometry based detection model for face manipulation using cnn-lstm architecture. *Information Sciences*, 633:370–383, 2023. 2, 3

[24] Luca Maiano, Lorenzo Papa, Ketbjano Vocaj, and Irene Amerini. Depthfake: a depth-based strategy for detecting deepfake videos. *arXiv preprint arXiv:2208.11074*, 2022. 3, 6

[25] Momina Masood, Mariam Nawaz, Khalid Mahmood Malik, Ali Javed, Aun Irtaza, and Hafiz Malik. Deepfakes generation and detection: State-of-the-art, open challenges, countermeasures, and way forward. *Applied intelligence*, 53(4):3974–4026, 2023. 2

[26] Thanh Thi Nguyen, Quoc Viet Hung Nguyen, Dung Tien Nguyen, Duc Thanh Nguyen, Thien Huynh-The, Saeid Nahavandi, Thanh Tam Nguyen, Quoc-Viet Pham, and Cuong M Nguyen. Deep learning for deepfakes creation and detection: A survey. *Computer Vision and Image Understanding*, 223:103525, 2022. 2

[27] Kyle Olszewski, Zimo Li, Chao Yang, Yi Zhou, Ronald Yu, Zeng Huang, Sitao Xiang, Shunsuke Saito, Pushmeet Kohli, and Hao Li. Realistic dynamic facial textures from a single image using gans. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5429–5438, 2017. 2

[28] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016. 2

[29] KR Prajwal, Rudrabha Mukhopadhyay, Vinay P Namboodiri, and CV Jawahar. A lip sync expert is all you need for speech to lip generation in the wild. In *Proceedings of the 28th ACM international conference on multimedia*, pages 484–492, 2020. 2

[30] Andreas Rossler, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies, and Matthias Nießner. Faceforensics++: Learning to detect manipulated facial images. In *Proc. of the IEEE/CVF international conference on computer vision*, pages 1–11, 2019. 4, 5, 7

[31] Ekraam Sabir, Jiaxin Cheng, Ayush Jaiswal, Wael AbdAlmageed, Iacopo Masi, and Prem Natarajan. Recurrent convolutional strategies for face manipulation detection in videos. *Interfaces (GUI)*, pages 80–87, 2019. 3

[32] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proc. of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018. 6

[33] Trevor Standley. A pytorch implementation of Xception. https://github.com/tstandley/Xception-PyTorch. Accessed: 2023-09-01. 6

[34] Zekun Sun, Yujie Han, Zeyu Hua, Na Ruan, and Weijia Jia. Improving the efficiency and robustness of deepfakes detection through precise geometric features. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3609–3618, 2021. 2, 3

[35] Supasorn Suwajanakorn, Steven M Seitz, and Ira Kemelmacher-Shlizerman. Synthesizing obama: learning lip sync from audio. *ACM Transactions on Graphics (ToG)*, 36(4):1–13, 2017. 2

[36] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR, 2019. 6

[37] Justus Thies, Michael Zollhöfer, and Matthias Nießner. Deferred neural rendering: Image synthesis using neural textures. *Acm Transactions on Graphics (TOG)*, 38(4):1–12, 2019. 5

[38] Justus Thies, Michael Zollhofer, Marc Stamminger, Christian Theobalt, and Matthias Nießner. Face2face: Real-time face capture and reenactment of rgb videos. In *Proc. of the IEEE conference on computer vision and pattern recognition*, pages 2387–2395, 2016. 5

[39] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008. 6, 7

[40] Wenqi Xian, Zhengqi Li, Matthew Fisher, Jonathan Eisenmann, Eli Shechtman, and Noah Snavely. Uprightnet: geometry-aware camera orientation estimation from single images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9974–9983, 2019. 3, 4, 5

[41] Xinsheng Xuan, Bo Peng, Wei Wang, and Jing Dong. On the generalization of gan image forensics. In *Chinese conference on biometric recognition*, pages 134–141. Springer, 2019. 3

[42] Pengpeng Yang, Rongrong Ni, and Yao Zhao. Recapture image forensics based on laplacian convolutional neural networks. In *Digital Forensics and Watermarking: 15th International Workshop, IWDW 2016, Beijing, China, September 17-19, 2016, Revised Selected Papers 15*, pages 119–128. Springer, 2017. 3

[43] Ning Yu, Larry S Davis, and Mario Fritz. Attributing fake images to gans: Learning and analyzing gan fingerprints. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 7556–7566, 2019. 3

[44] Tao Zhang. Deepfake generation and detection, a survey. *Multimedia Tools and Applications*, 81(5):6259–6276, 2022. 2

[45] Tianchen Zhao, Xiang Xu, Mingze Xu, Hui Ding, Yuanjun Xiong, and Wei Xia. Learning self-consistency for deepfake detection. In *Proc. of the IEEE/CVF international conference on computer vision*, pages 15023–15033, 2021. 3