https://doi.org/10.2352/ISSN.2470-1173.2021.4.MWSF-271

Detecting Deepfakes with Haralick's Texture Properties

Raphael Antonius Frick, Sascha Zmudzinski, Martin Steinebach; Fraunhofer SIT; Darmstadt, Germany

Abstract

In the recent years, the detection of deepfake videos has become a major topic in the field of digital media forensics, as the amount of such videos circulating on the internet has drastically risen. Providers of content, such as Facebook and Amazon, have become aware of this new threat to spreading misinformation on the Internet. In this work, a novel forgery detection method based on the texture analysis known from image classification and segmentation is proposed. In the experimental results, its performance has shown to be comparable to related works.

Introduction

Videos and images nowadays make up a huge part of news articles in order to support or prove an underlying story and are vastly shared on social media platforms. According to a report written by the *Pew Research Center* [1] it was stated, that in 2020 about 53% of the US citizens got their news either frequently or at least sometimes from social media. But also traditional media, such as news channels and newspapers, increasingly refer to *user created content*, as they are in some cases the only source available. In the current days, in which modifying an image or video footage is no longer difficult for even non-professionals to accomplish, it raises the challenge of assessing the credibility of image and video content in terms of its authenticity and integrity.

In this work, special interest lies on multimedia content in which a person's face is featured. Video filters to modify digital videos, especially face filters, have become very popular on social media platforms over the past years. Some of them, such as the freely available *FakeApp*, the mobile application *Zao* or *DeepfaceLab*, are capable to replace parts of a person's face with animated symbols, while others can project a users mimics onto existing 3D face models in real-time.

Providers of *social media platforms* and *web-hosting*, such as *Facebook* and *Amazon*, have acknowledged this new threat to maliciously spreading misinformation on the Internet. At the end of 2019, *Facebook* even hosted a competition, also known as the *DeepFake Detection Challenge (DFDC)*[2], revolving around the detection of deepfakes.

To address this challenge, we introduce a new approach based on the analysis of *Haralick's texture properties* in video and image data.

Structure of this work The remainder of this paper is organized as follows: In the next section, the methods *deepfake* attacks are reviewed and introductions to *Haralick's texture properties* and *focus measures* as a technical background of this work is given. The state-of-the-art in forgery detection techniques are then reviewed. The proposed algorithm is then explained in detail, followed by our experimental results. The paper concludes with a discussion of said results and displays proposals for future work.

Background

Deepfakes

Deepfakes are a special family of face swapping algorithms, in which the face replacement process is automated with the help of deeplearning. Algorithms to create such facial modifications have been in around since 2017 and have gained great popularity in the recent years. The most well-known implementations among them are *faceswap* [3] and *deepfacelab* [4]. Both of them follow a similar processing pipeline in order to create the *deepfakes*, which is explained in the following:

1. **Data Collection:** The training of the *AI-model* requires the collection of data for two datasets: One dataset containing the images or video frames of $person_A$, whose face shall be affected by the deepfake and another dataset containing images or videos featuring $person_B$, that shall be inserted to the video or image.
2. **Faceset Creation:** The datasets of both persons in question are then further processed to generate *facesets*. A face detection algorithm extracts the featuring faces from the data, resizing it to a given target size, while a face-aligning algorithm aligns the found faces using their estimated facial landmark points.
3. **Model Training:** The *AI-model* is most of the time represented by an autoencoder structure, featuring two encoders (one for each person) and a single decoder, which takes an aligned face image of $person_A$ as an input and outputs a newly synthesized face of $person_B$.
4. **Post-Processing:** Since the generated face images are aligned, they need to be transformed using affine warping to match with the position and orientation of $person_A$'s face.

Haralick's Texture Properties

The *textural* properties of digital images describe their characteristics with regards to the variations in the pixel's grey values and the presence (or absence, resp.) of basic patterns. In our approach, the texture in (facial) image regions will be evaluated for identifying malicious tampering caused by *deepfakes*.

For this, we express the image texture in terms of scalar features as presented by *Haralick et al.* [5] already in the 1970s: As a first step, for a greyscale image $f(x,y)$ the so-called "co-occurrence matrix" $H = h(i,j)$ is calculated. It describes the spatial-dependence of the gray value differences for neighboring pixels. It is defined as [6]:

$$\begin{aligned} h(i,j) = \texttt{prob}\big(\, & f(x_1,y_1) = i \,\wedge\, f(x_2,y_2) = j \,\wedge \\ & \wedge\, |(x_1,y_1)-(x_2,y_2)| = d \,\big) \end{aligned}$$

Please note that the term "neighbor" is ambiguous as grey value differences can be measured across different distances d (in pixels) and different angles θ, that is, $H = H_{d,\theta} = h_{d,\theta}(i,j)$.

Entries on the main diagonal of H indicate rather constant grey values across the image while entries *off the diagonal* indicate rather complex texture patterns.

From H the following texture measures are derived [7, 8]:

- *Angular second moment, ASM*: $\sum_{i,j} h_{i,j}^2$: it describes how uniform the grey values are. A value of 1.0 indicates a constant grey level across the image.
- *Energy*: $\sqrt{(}ASM)$: It is defined as the square root of the ASM texture property.
- *Contrast*: $\sum_{i,j} h_{i,j}^2 (i-j)^2$: it measures the degree of variation of grey levels in the image as entries *off* the diagonal are weighted higher and those *on* the diagonal are even discarded.
- *Dissimilarity*: this quantity $\sum_{i,j} h_{i,j}^2 |i-j|$ is similar to the Contrast measure but uses a different metrics.
- *Homogeneity*: $\sum_{i,j} h_{i,j}^2 / (1+(i-j)^2)$: on the opposite, this quantity weights entries *on* diagonal stronger than *off* the diagonal

While all these metrics appear to be correlating, they still differ by their metrics and have been used in image processing for decades.

In this paper, the *co-occurrence matrix H* generated from image patches of face regions.

Focus Measures

Focus measures have been used in the past for several purposes. They are used in digital photography to measure the focus of the auto-focus in a camera or are used for image segmentation. They can be split into several categories: derivative-based, statistical-based and transform-based [9].

In this work, our focus lies on the focus measures of the first category. These focus measures often utilize image filters for edge detection, such as the *Sobel-filter* (first-order-differentiation) or the *diagonalized Laplacian-filter* (second-order differentiation). Since blurry images tend to have softer-edges, the sharpness of an image can be used as a focus or blurriness indicator. The edge-detection images are often paired with statistical operations, such as the mean or variance value, in order to obtain a discrete measurement value.

Focus measures are used as an additional texture measurement tool in order to support enhance the classification.

Related Works

In this section, an overview of existing approaches to detect *deepfakes* is given.

Face X-ray is an approach developed by *Li et al.* [10] which detects, whether a face has been the output of two images blended together. Since this approach does not need to be trained on manipulated facial images, it is not biased towards the detection of faces generated from a certain facial manipulation algorithm. However, unfortunately it the detection rate is only high on the *FaceForensics++* dataset [11]. The data discrimination ability tends to be drastically reduced when evaluated on the newer *Celeb-DF*[12] and *Facebook DFDC Preview Dataset*[2].

The *deepfake* detection approach created by *Bonetti et al.*[13] utilizes an ensemble classifier consisting of several *ConvNets*. Other approaches based on *ConvNets* are [14] and [15].

Chugh et al. [16] analyze, whether there is a dissonance between the audio track and the visuals. The classification of this method is therefore however restricted to videos featuring audio. A similar work was conducted by *Mittal et al.* [17].

In [18] *Agarwal et. al* proposed an approach to detect *deepfake* videos by exposing inconsistencies in the behavior, facial expressions and head movements, of a subject.

DeepRhythm [19] by *Qi et al.* calculates the heart beat rate from from the *RGB-space*. They showed, that *deepfaked* videos do not produce a coherent sequential heart beat rhythm.

EfficientNets are a family of Convolutional Neural Network (ConvNets) models proposed by Mingxing Tan and Quoc V. Le in 2019 [20]. They have been extensively used during the Facebook deepfake detection challenge. Their effectiveness has been proven in the *ImageNet Classification Challenge*, in which the *EfficientNet-B7* was able to score a Top-1 accuracy of 84.4% and a Top-5 accuracy-score of 97.1%.

Further details on several other approaches can be found in the exhaustive survey on *deepfake* detection schemes in [21].

Proposed Method

In this section, the proposed detection scheme is presented. It utilizes characteristics in the texture of *deepfaked* material to expose an undergone forgery.

Overview on Feature Selection

Many of the proposed classifiers in related works are taking advantage of deeplearned features - features extracted automatically from a deeplearning model. While they usually perform very well in image classification tasks and as such also in the detection of deepfakes, the internals of *neural networks* are usually difficult to interpret by humans. Hence, one of the main motivations was create a classifier that is not reliant on such features.

During the assessment of the attack model, we could determine the following aspects in the *deepfake* creation process that affect the synthesized facial image texture:

1. **Autoencoder Resolution:** The amount of *output neurons* in the *autoencoders* used to generate these faces is limited. Early deepfake models were only able to utilize an output resolution of 64*x*64p. Hence, they were not able to train all facial details with high fidelity and the resulting information loss was causing the faces to become unsharp and to miss several facial details.
2. **Faceset Quality:** If the head poses featured in the *faceset* of $person_B$ do not reflect the ones made by $person_A$, the resulting video will become prone to containing temporal flickering as well as to introducing strong blurring artifacts.
3. **Affine Transformations:** Since all generated faces remain aligned after synthesis, their position, orientation and size need to be adjusted to the target area. For this purpose, affine transformations are utilized, which take use of interpolation, e.g. *bilinear-* or *bicubic interpolation*. Interpolation however can introduce additional blurring to the image data as shown by Figure 1.

These introduced textural artifacts will however not be present in parts of the face that are left unmodified by the *deepfake* algorithm. Hence, it can be expected that the skin in the forehead region will appear "sharper" (in average) than in the cheek region in the analyzed image frame (see real world example in Figure 2).

This is likely to cause inconsistencies in the texture properties (as explained) between manipulated and authentic parts of a face and can therefore be exploited to detect facial manipulations.

Classification Pipeline

In the following sections, we explain the main components of our approach based on the classification pipeline displayed in Figure 3.

Estimating the Regions of Interest (ROI) For a given input video I, each individual video frame is extracted. Then, for each frame we define two *regions of interest (ROI)* in the face of a person as visualized in Figure 3:

- `T-ROI`: This region of interest is hereby represented by the facial area in question of having undergone a modification. In the case of most *deepfaked* faces, this region is located between the eyebrows and the underlip.
- `A-ROI`: This defines the region of the face, which is considered to be *authentic* anyway. Since the forehead is mostly not affected by the *deepfake* algorithm, it has been considered as `A-ROI`.

Defining both `ROIs` in the facial area of the subject is important as other parts of the video frame may be out of focus and thus contain natural blur. The `ROIs` are automatically extracted with the help of a facial landmarks extractor [23]. In our implementation, the same facial landmarks extractor algorithm used by the *faceswap deepfake* algorithm was utilized. Since the implementation only maps its 68 facial landmarks points between the chin and the eyebrows, the forehead area is estimated by taking human face proportions into account.

Image Filtering The blurriness (or sharpness, resp.) of an image can be measured by analyzing the strength of edges using *focus measures*. In this paper, we propose using the *diagonalized Laplacian filter*. The following analysis of facial regions are calculated on the filtered video frames as well as the corresponding *RGB* values of the patches.

Extracting Texture Properties Instead of calculating the texture properties on all three channels in the *RGB colorspace*, only the *Y-channel* containing the luminosity information in the *YCbCr colorspace* is considered. *YCbCr* is widely used in compression standards such as *JPEG* and *H.264/H.265* and allows the classifier to make its decision independently from the distribution of color.

The video channel information is encoded as 8-bit values, and no further quantization was applied. Furthermore, as texture calculations are best performed on a symmetrical matrix $H_{d,\theta}$, the resulting *greylevel co-occurrence matrix* was made symmetrical around the diagonal to match with the matrix proposed by Haralick [5].

We then propose to utilize the six texture properties $p_i(i = 1...6)$ on the *Y-channel* as explained in section , that is *Contrast*, *Dissimilarity*, *Homogeneity*, *ASM*, and as well as the *Energy*.

This was carried out on the co-occurrence matrices H across angles $\theta = 0, \pi/4, \pi/2, (3/4)\pi$. A pixel offset of $d = 1$ was chosen.

Additionally, the difference in the average intensity of the pixel values between both `ROIs` was determined by calculating the mean of the *Y-channel* pixel values for each `ROI` as it helps to spot errors in the color-correction applied during the forgery.

Feature-Vector Composition To form the feature-vector, the ratio r between each texture property p_i extracted from `A-ROI` and `T-ROI` has been computed as displayed in equation 1. The expression "+1" in the denominator is required, as it prevents dividing by 0.

$$r = \frac{p_{(\texttt{A-ROI})}}{p_{(\texttt{T-ROI})} + 1} \tag{1}$$

The resulting feature vector consists of $(5*4+1)*2 = 42$ dimensions (5 texture-properties per angle and the average pixel value intensity) for the luma and filtered channels.

The features are then fed into a *SVM* classifier, which returns the probability of a face being modified by a *deepfake* algorithm.

During classification, the classifier outputs a probability score for a face being modified. This probability value can be compared to a threshold T in order predict the target class. The optimal threshold value is obtained during the training phase, by calculating the cut-off point between the feature-class distributions. The cut-off maximizes the accuracy and should therefore be used as the threshold value T.

Evaluation

Implementation Setup

The deepfake detector was written in Python 3.7 and was running on an *Intel Core i5-3750k @ 3.40GHz* utilizing up to 4 concurrent threads with access to *8GB RAM*, as well as an discrete *NVidia GTX 1050Ti* with access to *4GB VRAM*.

For the extraction of the facial region, the face detection and alignment library by [23] was utilized, which achieves a superior detection accuracy at a higher computation cost than the facial landmarks estimator featured in *DLib*. The *Haralick GLCMs* were calculated with the help of *scikit-image* [7]. Furthermore, *FFmpeg*[24] was utilized for the video-frame input and output operations. *Keras* [25] and the underlying *Tensorflow (2.4-nightly)* deeplearning framework were used to train the models based on *EfficientNet*.

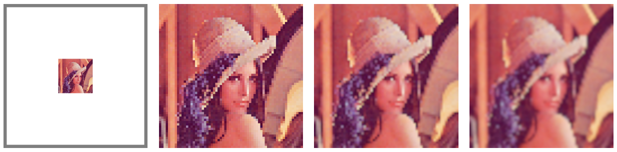


Figure 1: Interpolation techniques have an impact on texture properties. Example: upscaling from $64*64$ px to $256*256$ px, from left to right: original, nearest neighbor, bilinear, bicubic

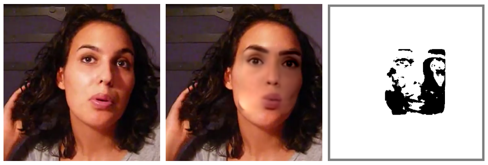


Figure 2: Example: original frame (left); deepfaked frame featuring blurriness and missing shadows (middle); difference mask (right); source [22]

Training & Test Datasets

The evaluation was conducted mainly on three datasets: The *FaceForensics++ dataset*[22], the *Celeb-DF (v2) dataset*[12] and the *Facebook Deepfake Detection Challenge preview dataset*[26].

FaceForensics++ Early 2019, the *Visual Computing Group* at *TUM University*, Munich, released the *FaceForensics++* dataset [22], a video dataset consisting of 1000 authentic and 1000 *deepfake* videos featuring multiple strength of compression. The *deepfake* videos were based on the 1000 authentic videos, which were taken from the *YouTube-8M Dataset* [27].

Celeb-DF (v2): A New Dataset for DeepFake Forensics The *Celeb-DF (v2) Dataset*[12] was released in November 2019 and contains 6229 videos, from which 590 of the videos are pristine and 5639 videos have been deepfaked.

The dataset is focussed on video data containing celebrities. For this purpose, videos featuring people varying in gender, heritage and situation have been gathered from *YouTube* and then modified using a deepfake algorithm.

The Deepfake Detection Challenge (DFDC) Preview Dataset The *Deepfake Detection Challenge Preview Dataset*, short *DFDC Preview Dataset*, was released by *Facebook* in October 2019 as part of the name-giving competition.

The *preview dataset* contains 5,710 videos of which 1,249 are authentic and the rest modified using two different *deepfake* algorithms. 4,277 were modified using one algorithm and the remaining 184 by another algorithm.

SVM Classifier Design

For the classification a *SVM* classifier was chosen. Since *SVMs* work best on normalized data, the values of the feature-vector were scaled to $[0,1]$. For the hyperparameter selection *grid-search* was utilized in conjunction with a *10-fold cross-validation*.

Evaluation Results

FaceForensics++ We set the training/testing partitioning to 1:2, featuring videos of all compression configurations. The performance of our proposed approach and several existing methods, which where evaluated on the *FaceForensics++* dataset, are displayed in Table 1. We also compare the proposed method of this paper with our previous approach based on *compression ghost artefacts*, see [28].

As it can be seen in table 1, our classifier is able to produce similar state-of-the-art results as the ones presented by related works when classifying RAW quality videos. However, with increasing compression strength, the classification rate drastically decreases and the classifier is outperformed by deeplearning based algorithms, such as *MesoNet*[29] and *XCeptionNet*[22]. As compression in image and video codec have similar effects to a low-pass filter, it affects our estimated texture properties depending on the utilized compression strength.

However, the classifier improves upon our previous approach [28] based on *ghost artefacts* by great extent.

Celeb-DF (v2) The *Celeb-DF (v2) dataset* provides predefined train/test-splits, which were utilized during the evaluation of our classifier. The corresponding results of the classification on the test dataset are displayed in Table 2.

In comparison to the classification on the *FaceForensics++ dataset*, the ability to distinguish between authentic and modified videos has drastically decreased, resulting in an *AUC-score* of 0.773. As the quality of deepfakes increase, differing between authentic and deepfaked regions become harder to accomplish. Furthermore, the face of the persons featured in the dataset are not always directly facing the camera, and thus the automatic estimation of the `ROIs` is not always precise.

The increment in classification difficulty can also be observed in the classification results by related works, as only the deeplearning based approach proposed by Tolosana et al. features a high *AUC-score* of 0.836 surpassing the score of our classifier.

DFDC Preview Dataset The *DFDC preview dataset* comes with predefined train/test-splits, that have been used during evaluation. The results of the classification on the test dataset are displayed in Table 2.

The classifier achieves a moderate AUC of 0.7333. This is to be expected, as the videos in the dataset are heavily augmented and as such also feature artificial blur. As some videos were modified using a facial modification algorithm, that is able to exchange the full facial area, the forehead serving as our `A-ROI` is no longer authentic. Classifying these videos will therefore result in misclassifications.

Nevertheless, in comparison the proposed approach is still able to outperform most of the related state-of-the-art *deepfake* detection schemes as can be seen in Table 2. However, the detection scheme by *Tolosana et al.*[30] and *Agarwal et al.*[18] present superior class separability than our proposed work.

EfficientNet-B2

During evaluation, we also wanted to analyze whether it is possible to enhance the classification performance of *ConvNet* based approaches, such as *EfficientNet-B2*. For this purpose, we utilized transfer-learning to train the *EfficientNet* to solve the deepfake detection task. The corresponding baseline results achieved on the *Celeb-DF (v2) dataset* and *Facebook's DFDC preview dataset* can be viewed in Table 2.

To join the classification decisions of both classifiers, the ones made by *our proposed texture-property* based classifier and the ones made by *EfficientNet-B2*, we trained a *meta-classifier* in the form of another *SVM*. It takes the probabilities of both classifiers, weights them internally and outputs a unified decision probability.

As it can be seen in Table 2, the stacked classifier was not able increase the classification performance. The reason for that is, that the *ConvNet* based approach also analyzes the texture using several filters at multiple dimensions. Thus, it is able to extract a lot more information from pixel values than the *GLCM*. Therefore, when combining both classifiers, the *Haralick's texture properties* based classifier does not provide any useful *information gain*.

Conclusion

In this paper, we propose a novel approach to detect *deepfakes* in video data using the analysis of texture properties. The evaluation was conducted on the *FaceForensics++ dataset*, the *DFDC preview dataset* from Facebooks Deepfake Detection Chal-

	Fridrich [31][22]	Cozzolino [11][22]	Bayar & Stamm [32][22]	Rahmouni [33][22]	MesoNet [29][22]	XceptionNet [34][22]	Ghost [28]	Ours (GLCM)
RAW	0.9903	0.9883	0.9928	0.9803	0.9841	0.9959	0.9840	0.9955
HQ	0.7712	0.8178	0.9018	0.8216	0.9526	0.9885	-	0.9179
LQ	0.6558	0.6826	0.8095	0.7325	0.8952	0.9428	-	0.6460

Table 1: Accuracies achieved by our proposed method and related work on the latest FaceForensics++ dataset [35]

	DFDC Preview	Celeb-DF (v2)
Matern et al. [36] [21]	0.662	0.551
Yang et al. [37] [21]	0.559	0.546
Li et al. [38] [21]	0.755	0.646
Afchar et al. [29] [21]	0.753	0.548
Zhou et al. [39] [21]	0.614	0.538
Nguyen et al. [40] [21]	0.536	0.543
Nguyen et al. [41] [21]	0.533	0.575
Tolosana et al. [30]	0.910	0.836
Agarwal et al. [18]	0.930	0.990
Ours (GLCM)	0.733	0.773
Ours (EfficientNet-B2)	0.790	0.920
Ours (Ensemble Classifier)	0.789	0.920

Table 2: AUC-scores achieved on Facebook's DFDC preview dataset and the Celeb-DF (v2) dataset

lenge and the *Celeb-DF (v2) dataset* featuring *deepfake* videos of different quality and diversity.

Based on the results on the *DFDC preview dataset* and the *Celeb-DF (v2) dataset*, the proposed method was able to achieve similar results as related works, although it was not able to detect the forgeries with total high accuracy.

In comparison to older related works however, which based their evaluation on the *FaceForensics++ dataset*, the method of this paper was able to detect low and mid-level compressed videos with high accuracy, exceeding the results of most works. The method was not able to distinguish between authentic and *deepfake* videos very well if they have been compressed using a high value of the *quantization parameter* and thus featured a very high compression (low image quality).

As we cannot assure the authenticity of the forehead region at all time, future work could revolve around being less dependent on the `ROIs` overall as well as incorporating the utilization of temporal features.

Acknowledgment

This work was supported by the German Federal Ministry of Education and Research (BMBF) and/or the and the Hessian Ministry of Higher Education, Research, Science and the Arts within their joint support of "ATHENE – Disinformation on Corona" and "Lernlabor Cybersicherheit".

References

[1] E. Shearer and A. Mitchell. (2021) News use across social media platforms in 2020. [Online]. Available: https://www.journalism.org/wp-content/uploads/sites/8/2021/01/PJ_2021.01.12_News-and-Social-Media_FINAL.pdf

[2] B. Dolhansky, R. Howes, B. Pflaum, N. Baram, and C. C. Ferrer, "The deepfake detection challenge (dfdc) preview dataset," 2019.

[3] deepfakes (Github user). (2019) Github repository 'deepfake/faceswap'. [Online]. Available: https://github.com/deepfakes

[4] I. Perov, D. Gao, N. Chervoniy, K. Liu, S. Marangonda, C. Umé, M. Dpfks, C. S. Facenheim, L. RP, J. Jiang, S. Zhang, P. Wu, B. Zhou, and W. Zhang, "Deepfacelab: A simple, flexible and extensible face swapping framework," 2020.

[5] R. M. Haralick, K. Shanmugam, and I. Dinstein, "Textural features for image classification," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-3, no. 6, pp. 610–621, 1973.

[6] G.-H. Liu and J.-Y. Yang, "Image retrieval based on the texton co-occurrence matrix," *Pattern Recogn.*, vol. 41, no. 12, pp. 3521–3527, Dec. 2008.

[7] S. Van der Walt, J. L. Schönberger, J. Nunez-Iglesias, F. Boulogne, J. D. Warner, N. Yager, E. Gouillart, and T. Yu, "scikit-image: image processing in python," *PeerJ*, vol. 2, p. e453, 2014.

[8] scikit-image development team, "scikit-image – Image Processing Toolbox for SciPy," 2019, module "feature.texture", available at https://scikit-image.org/docs/0.7.0/api/skimage.feature.texture.html (verified July 2020).

[9] J. Pech-Pacheco, G. Cristobal, J. Chamorro-Martinez, and J. Fernandez-Valdivia, "Diatom autofocusing in brightfield microscopy: a comparative study," in *Proceedings 15th International Conference on Pattern Recognition. ICPR-2000.* IEEE Comput. Soc.

[10] L. Li, J. Bao, T. Zhang, H. Yang, D. Chen, F. Wen, and B. Guo, "Face x-ray for more general face forgery detection," 2019.

[11] A. Rössler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Niessner, "Faceforensics: A large-scale video dataset for forgery detection in human faces," *arXiv*, vol. arXiv:1803.09179v1 [cs.CV], 2018.

[12] Y. Li, P. Sun, H. Qi, and S. Lyu, "Celeb-DF: A Large-scale Challenging Dataset for DeepFake Forensics," in *IEEE Conference on Computer Vision and Patten Recognition (CVPR)*, Seattle, WA, United States, 2020.

[13] N. Bonettini, E. D. Cannas, S. Mandelli, L. Bondi, P. Bestagini, and S. Tubaro, "Video face manipulation detection through ensemble of cnns," 2020.

[14] D. M. Montserrat, H. Hao, S. K. Yarlagadda, S. Baireddy, R. Shao, J. Horváth, E. Bartusiak, J. Yang, D. Güera, F. Zhu, and E. J. Delp, "Deepfakes detection with automatic face weighting," 2020.

[15] S. Kaur, P. Kumar, and P. Kumaraguru, "Deepfakes: temporal sequential analysis to detect face-swapped video clips using convolutional long short-term memory," *Journal of Electronic Imaging*, vol. 29, no. 3, pp. 1 – 17, 2020.

[16] K. Chugh, P. Gupta, A. Dhall, and R. Subramanian, "Not made for each other- audio-visual dissonance-based deepfake detection and localization," 2020.

[17] T. Mittal, U. Bhattacharya, R. Chandra, A. Bera, and D. Manocha, "Emotions don't lie: A deepfake detection method using audio-visual affective cues," 2020.

[18] S. Agarwal, T. El-Gaaly, H. Farid, and S.-N. Lim, "Detecting deepfake videos from appearance and behavior," 2020.

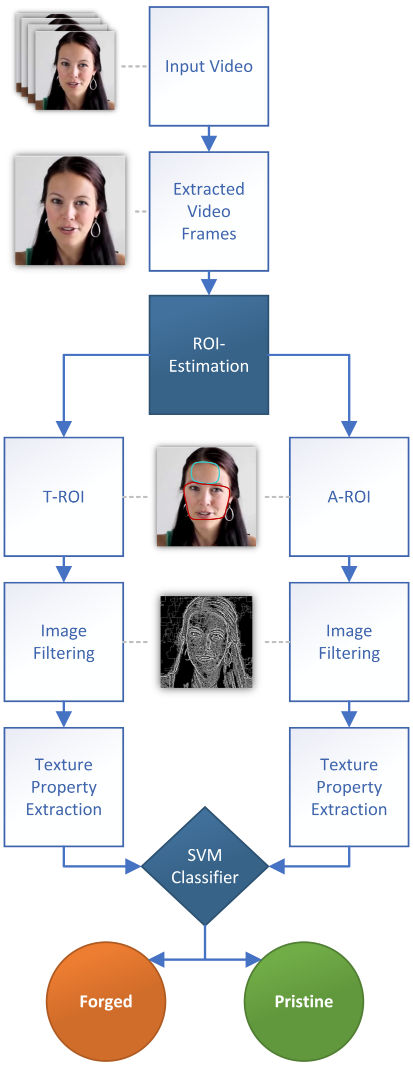


Figure 3: Classification Flowchart

[19] H. Qi, Q. Guo, F. Juefei-Xu, X. Xie, L. Ma, W. Feng, Y. Liu, and J. Zhao, “Deeprhythm: Exposing deepfakes with attentional visual heartbeat rhythms,” 2020.

[20] M. Tan and Q. V. Le, “Efficientnet: Rethinking model scaling for convolutional neural networks,” 2020.

[21] R. Tolosana, R. Vera-Rodriguez, J. Fierrez, A. Morales, and J. Ortega-Garcia, “Deepfakes and beyond: A survey of face manipulation and fake detection,” 2020.

[22] A. Rössler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Niessner, “Faceforensics++: Learning to detect manipulated facial images,” *arXiv*, vol. arXiv:1901.08971v2 [cs.CV], 2019.

[23] A. Bulat and G. Tzimiropoulos, “How far are we from solving the 2d & 3d face alignment problem? (and a dataset of 230,000 3d facial landmarks),” in *International Conference on Computer Vision*, 2017.

[24] FFmpeg. (2019) FFmpeg multimedia framework. [Online]. Available: https://ffmpeg.org/

[25] F. Chollet *et al.*, “Keras,” https://keras.io, 2015.

[26] B. Dolhansky, J. Bitton, B. Pflaum, J. Lu, R. Howes, M. Wang, and C. C. Ferrer, “The deepfake detection challenge dataset,” 2020.

[27] S. Abu-El-Haija, N. Kothari, J. Lee, P. Natsev, G. Toderici, B. Varadarajan, and S. Vijayanarasimhan, “Youtube-8m: A large-scale video classification benchmark,” 2016.

[28] R. A. Frick, S. Zmudzinski, and M. Steinebach, “Detecting ”deepfakes” in h.264 video data using compression ghost artifacts,” in *Electronic Imaging, 3D Measurement and Data Processing 2019*. Society for Imaging Science and Technology, 2020.

[29] D. Afchar, V. Nozick, J. Yamagishi, and I. Echizen, “Mesonet: a compact facial video forgery detection network,” *arXiv*, vol. arXiv:1809.00888v1 [cs.CV], 2018.

[30] R. Tolosana, S. Romero-Tapiador, J. Fierrez, and R. Vera-Rodriguez, “Deepfakes evolution: Analysis of facial regions and fake detection performance,” 2020.

[31] J. Fridrich and J. Kodovsky, “Rich models for steganalysis of digital images,” *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 3, pp. 868–882, jun 2012.

[32] B. Bayar and M. C. Stamm, “A deep learning approach to universal image manipulation detection using a new convolutional layer,” in *IH&MMSec ’16*, 2016.

[33] N. Rahmouni, V. Nozick, J. Yamagishi, and I. Echizen, “Distinguishing computer graphics from natural images using convolution neural networks,” in *2017 IEEE Workshop on Information Forensics and Security (WIFS)*, Dec 2017, pp. 1–6.

[34] F. Chollet, “Xception: Deep learning with depthwise separable convolutions,” *arXiv*, vol. arXiv:1610.02357v3 [cs.CV], 2016.

[35] Faceforensics++ test set updated july 2020. [Online]. Available: https://github.com/ondyari/FaceForensics/blob/master/dataset/FaceShifter/README.md

[36] F. Matern, C. Riess, and M. Stamminger, “Exploiting visual artifacts to expose deepfakes and face manipulations,” in *2019 IEEE Winter Applications of Computer Vision Workshops (WACVW)*, 2019, pp. 83–92.

[37] X. Yang, Y. Li, and S. Lyu, “Exposing deep fakes using inconsistent head poses,” 2018.

[38] Y. Li and S. Lyu, “Exposing deepfake videos by detecting face warping artifacts,” 2018.

[39] P. Zhou, X. Han, V. I. Morariu, and L. S. Davis, “Two-stream neural networks for tampered face detection,” 2018.

[40] H. H. Nguyen, F. Fang, J. Yamagishi, and I. Echizen, “Multi-task learning for detecting and segmenting manipulated facial images and videos,” 2019.

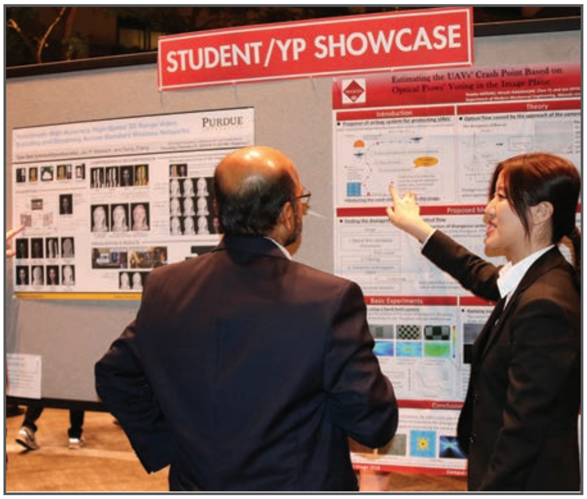
[41] H. H. Nguyen, J. Yamagishi, and I. Echizen, “Use of a capsule network to detect fake images and videos,” 2019.

Author Biography

Raphael Antonius Frick is a student of computer science at Technische Universität Darmstadt, Germany. He works as an assistant researcher at the Media Security and IT Forensics division at Fraunhofer SIT. His research is focused on detecting manipulations in image and video, especially on deepfake detection approaches.

Sascha Zmudzinski is a senior researcher at Fraunhofer SIT in the Media Security and IT Forensics division. He received his PhD at the TU Darmstadt in 2017 for his work on authentication audio watermarking.

Martin Steinebach is the manager of the Media Security and IT Forensics division at Fraunhofer SIT. In 2003 he received his PhD at the Technical University of Darmstadt for this work on digital audio watermarking. In 2016 he became honorary professor at the TU Darmstadt.


STUDENT/YP SHOWCASE
PURDUE

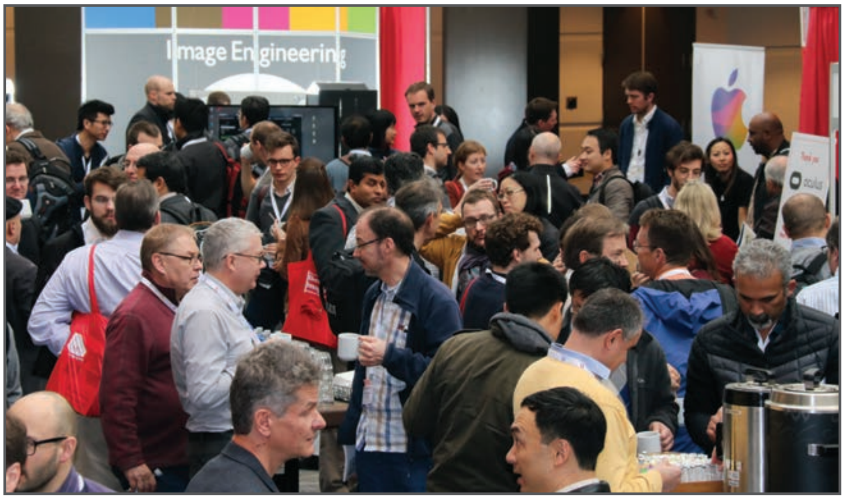


Image Engineering

My first working AR prototype
Today's platforms
Ubiquitous consumer AR
Optimis this is << 20