Deep Classifiers and Wavelet Transformation for Fake Image Detection

Stanislaw Osowski[1,2] and Maciej Golgowski[1]

[1] *Military University of Technology, Warsaw, Poland,*
[2] *Warsaw University of Technology, Warsaw, Poland*

https://doi.org/10.26636/jtit.2023.4.1336

Abstract — The paper presents a computer system for detecting deep fake images in videos. The system is based on continuous wavelet transformation combined with a set of classifiers composed of a few convolutional neural networks of diversified architectures. Three different forms of forged images taken from the FaceForensics++ database are considered in numerical experiments. The results of experiments involving the proposed system have shown good performance in comparison to other current approaches to this particular problem.

Keywords — continuous wavelet transform, convolutional neural networks, deep fake, ensemble of classifiers

1. Introduction

The ability to recognize face-manipulated images is currently an interesting research problem, as such images are frequently exploited for malicious purposes. Different deep fake algorithms based on either auto-encoders or generative adversarial networks are capable of replacing faces in target videos with artificially created images. Different programs used for the generation of manipulated images, such as FakeApp, FaceSwap or Face2Face, are nowadays easily available and are commonly used across the Internet. The methods for detecting such manipulated images have become increasingly important for individuals, businesses, and governments alike. Many world-leading institutions and companies, such as Facebook, Microsoft, Amazon, MIT, Berkeley, Oxford University, etc., have joined their efforts to cope with this problem. Consequently, large-scale databases, such as FaceForensics++ [1] or DeeperForensic [2], [3] are available to train and test forged image detection algorithms.

Several detection algorithms have been developed recently which demonstrate good performance in terms of detecting forged images in videos. These are based mainly on different configurations of deep convolutional neural networks (CNN), such as Siamese CNN [4], capsule architecture [5], different architectures of inception CNNs [1], [6], [7] long short-term memory (LSTM) networks [8] or combinations of deep structures and other aspects of signal processing, e.g. CNN combined with biological signals [9] or CNN combined with gated recurrent units (GRU) and spatial transformer networks [10].

This paper proposes a different approach to the problem. It applies continuous wavelet transformation (CWT) to original images in the process of generating input data for an ensemble composed of different types of CNN architectures. The data fed to the ensemble are processed independently by parallel processing CNN units and the final classification decision is made by majority voting. The results of numerical tests performed with the use of the FaceForensics++ database have shown an increased degree of accuracy of the forged image detection process.

This article is organized as follows. In Section 2, we introduce three recently developed deep fake video generation algorithms which have been applied in our investigations. The proposed approach relied upon to detect forged images is described in Section 3. Section 4 presents and discusses the results of deep fake image recognition tests performed with the use of images taken from the FaceForensics++ database. The obtained results are compared with the recent achievements in his field in Section 5. Finally, concluding remarks are given in Section 6.

2. Database Creation

The numerical experiments have been performed using the FaceForencics++ database [11]. It contains 1,000 original video sequences taken from YouTube and the same number of synthetic videos with a few generative models. All face images are presented in the frontal position. The images were subjected to different artificial modifications involving the faces. Three types of manipulation applications were

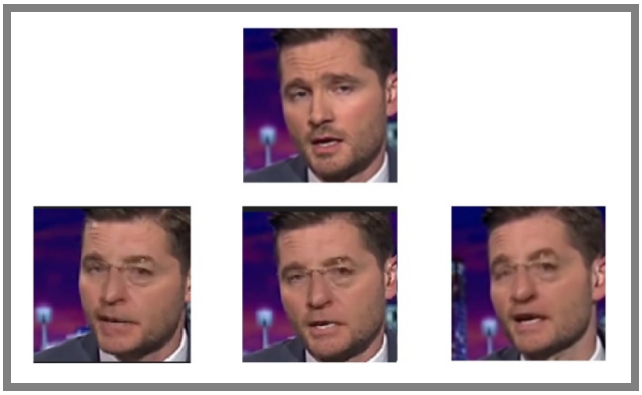

Fig. 1. Examples of manipulated images: upper row – original face, lower row – manipulated faces obtained with the use of the FaceSwap method.

Fig. 2. Examples of manipulated images: upper row – original face, lower row – manipulated faces obtained with the use of the FaceApp method.

used while creating the artificial images: FaceSwap, FakeApp (Deepfakes), and Face2Face [1], [11].

FaceSwap, created by Laan Labs, allows users to swap faces online. The method is based on the graphical transfer of characteristic features of the original region of an image to its newly created counterpart. The colors of the associated images are corrected. Figure 1 presents examples of manipulated images, created based on the genuine picture.

The FakeApp method, also known as Deepfakes, applies the autoencoder-decoder technique while creating the falsified image. Two images A and B are first analyzed during the learning process of an autoencoder-decoder system. In the generation of the manipulated image, the decoder of A is replaced with the decoder of B. Thus, characteristic features of image A are transferred to image B. The results of the decoder are associated with the rest of the image using an interpolation algorithm and Poisson edition. Examples of images created with the use of the FakeApp method are presented in Fig. 2. The upper row shows the original face and the lower row presents fake images generated with the use of the FakeApp method [12].

The Face2Face method reconstructs the face by transferring a facial expression from the source video to the target file. In the first step, the original video frame is chosen. The system traces the expression of the face in the subsequent frames. Finally, the blend shape coefficients of these expressions are

Fig. 3. Examples of manipulated images: upper row – original face, lower row – manipulated faces obtained with the use of the Face2Face method.

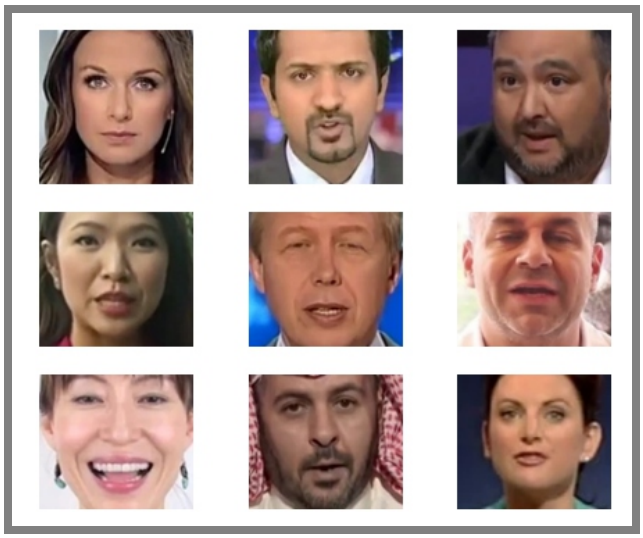

Fig. 4. Examples of face images used in numerical experiments.

combined and applied to the original image, forming the fake image. The results of using the Face2Face method are presented in Fig. 3.

The FaceForensics++ database contains 1,000 original videos and 1,000 manipulated videos, each generated with the use of the three methods referred to above. Consequently, 509,914 video frames are available. The images were compressed using the H.264 codec and the MPEG-4 AVC technique. An average compression ratio classified as c23 was achieved. Due to our limited computation resources, 1,000 video frames representing the original and three types of fake images (4,000 in total, all compressed) have been used in the numerical experiments in this work.

From each film (one original and three manipulated files), we selected, randomly, those frames that represented images shown within a 1-second sequence. Each original face in the frame was accompanied by three manipulated faces.

The next step was to extract the fragments of the frame representing the face image only. This was done using the histogram of oriented gradients (HOG) [13]. In the first step, the gradient values of the image pixels were computed. This was done by applying a 1D centered, point discrete derivative mask in the horizontal and vertical directions. In the second step, the cell histograms were created. Each pixel within the cell brings a weighted vote for an orientation-based histogram channel, based on the values found in the gradient computation. The gradient magnitude itself contributes to the weight of the vote.

The gradient strengths are locally normalized by grouping the cells together into larger, spatially connected blocks. The numerical descriptors generated by HOG are represented by the concatenated vector of the components of the normalized cell histograms from all regions of the block. The blocks are overlapping, which means that the cells may contribute to the final descriptor more than once. The HOG descriptors serve as features relied upon for recognizing objects in the analyzed frame images.

All those steps have resulted in creating a database of original and manipulated images extracted from the frames of

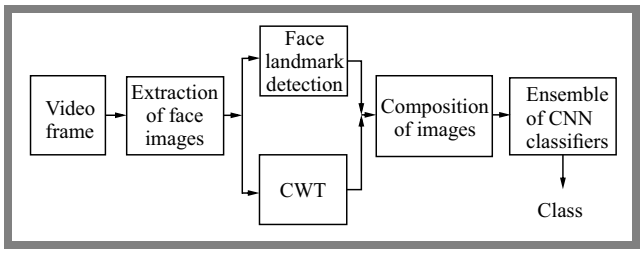

Fig. 5. The proposed system for acquisition and recognition of fake images. Images extracted from video frames are subjected to CWT and face landmark detection. The results of those processes are superimposed and serve as a basis for creating input tensors to the ensemble of CNN classifiers.

a video stream. Figure 4 shows some examples of face images extracted from the videos available in FaceForensics++. Together with the accompanying manipulated images, they create the database on which the numerical experiments were based. Consequently, the database used in the experiments was made up of 1,000 original images combined with 1,000 manipulated images, each created by applying FaceSwap, FaceApp, and Face2Face algorithms.

3. Proposed System for the Detection of Forged Images

The proposed method for detecting deep fake images can be divided into the following steps.

1) Extraction of face images from the video.
2) Face landmark detection and localization in the image.
3) Continuous wavelet transformation of the image combined with the inclusion of landmarks. The images created in this way serve as a basis for the creation of input attributes used by deep classifiers.
4) Application of the ensemble composed of several deep CNN architectures used for recognizing fake images. The input tensors for the CNN are created by combining images resulting from CWT.

Figure 5 contains a graph presenting the above-mentioned steps taken to acquire and recognize fake images.

3.1. Face Landmark Detection

The face landmark detection algorithm uses descriptors generated by the histogram of oriented gradients (HOG) algorithm [13]. It counts the occurrences of gradient orientations in localized portions of an image and performs computations using a dense grid of uniformly spaced cells. To improve accuracy, it uses overlapping local contrast normalization. The features obtained in this way are used to find the characteristic points of the face characterizing its structure. 68 such points have been used.

3.2. Continuous Wavelet Transformation of Images

To create the input attributes used by the classifier, we subjected the images to 2-dimensional continuous wavelet transformation (CWT). The 2D CWT is a representation of 2D data (the image) with the use of 3 variables: scale (dilation), and position (in 2 dimensions) [14]. Scale is a real-value scalar and position is the 2D vector with real-valued elements. Representing the image as a function $\mathrm{f}(x)$ with x being a two-element vector of real numbers associated with the position of pixels in the 2-dimensional space, the CWT is defined as follows:

$$W_f(a,b) = |a| \int_{-\infty}^{\infty} f(x) \frac{1}{a} \psi\left(\frac{x-b}{a}\right) \mathrm{d}x \,, \tag{1}$$

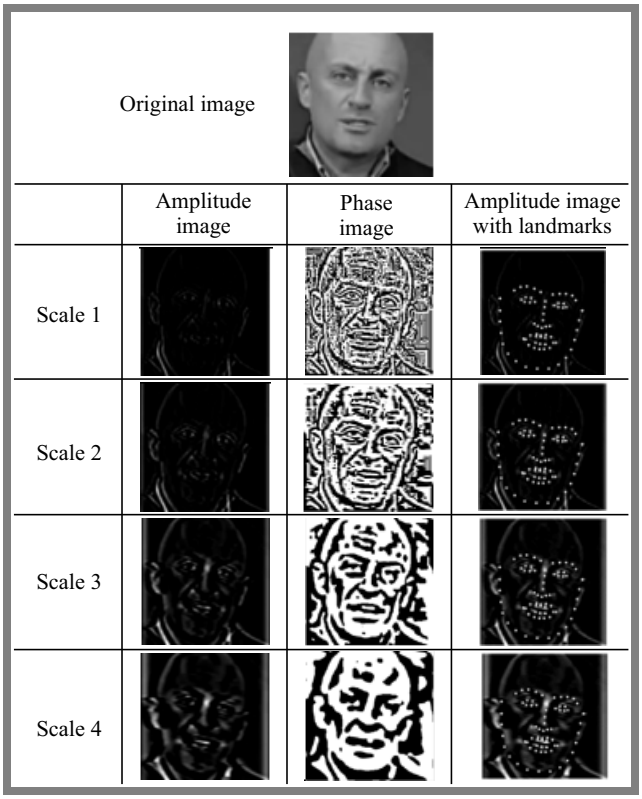

Fig. 6. Examples of CWT-transformed face images: amplitude and phase representations at the application of the Mexican hat as the mother wavelet. The last column presents amplitude images with superimposed landmark points.

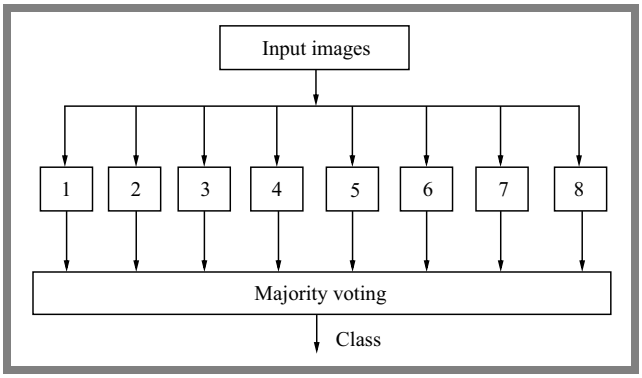

Fig. 7. The ensemble of deep CNN structures used for fake image recognition. The CNN structures are denoted by the following numbers: 1 – alexnet, 2 – mobilenetv2, 3 – resnet50, 4 – efficientnet, 5 – squeezenet, 6 – googlenet, 7 – shufflenet, 8 – inceptionresnetv2. Majority voting is used to find the winning class.

where ψ represents the 2-dimensional wavelet function, a is the continuous scale and b represents the continuous shift. The 2D CWT is a complex space-scale representation of an image.

The complex results achieved by applying the CWT technique may be represented by module and phase images. The application of different scale values results in slightly different results of CWT. This is seen in Fig. 6, which contains amplitude and phase representations of the original image presented in the first row, for the values of scale changing from 1 to 6. There are visible differences in the resulting images using different scale factors (both in amplitude and phase representations). Combining them together leads to an improved representation of the details of the analyzed image. Their combination might be done in different ways to create the input attributes for the block of CNN classifiers. Moreover, to provide additional information on the structure of the analyzed image, the face landmarks are superimposed on the results of the amplitude images created by CWT. The result of such an operation is illustrated in the last column in Fig. 6. The continuous wavelet transform plays an important role in the system. Wavelet transformation decomposes the image into many levels with different resolution values. Such an approach provides more information on the details in different regions of the image, thus enhancing the differences between the original and forged images. Therefore, the classifiers are supplied with richer information about the analyzed images and are able to generate more accurate results.

3.3. CNN Ensemble System for Classification

By applying the CWT method, space for the generation of input attributes used by the classification system is created. In our solution, we have applied an ensemble of CNN classifiers. CNN classifiers combine two functions within one structure: they generate and select diagnostic features, also performing the final stage of classification [15], [16].

In creating the ensemble, it is important to ensure independent operation of its constituent members. Therefore, we decided to use the transfer learning technique applied to different architectures of CNN. After some introductory experiments, the following CNN pre-trained models have been chosen: alexnet, mobilenetv2, resnet50, efficientnet, squeezenet, googlenet, shufflenet and inceptionresnetv2 [17]–[22]. The classification system defined in this way is presented in Fig. 7. The applied CNN architectures differ significantly in terms of signal processing, types and sizes of filters applied, number of processing layers, etc. All of them rely on cross-entropy in the definition of the loss function, and on the softmax classifier in the output layer. For the purpose of training these CNN networks, the ADAM optimizer with an initial learning rate of 0.001 was used.

The choice of input attributes used by the classification system was made based on the introductory experiments. They have shown that the best results were obtained by applying amplitude representations of CWT-transformed images of different scales. Due to the fact that the pre-trained CNN architectures allow 3 parallel images to be used, in parallel, as input, we have tried different combinations of 3 amplitudes as well as phases of such images, corresponding to different scales. As a result of the experiments, we have found that the information contained in 3 subsequent scales of amplitude CWT leads to the greatest improvement in the operation of the classification system.

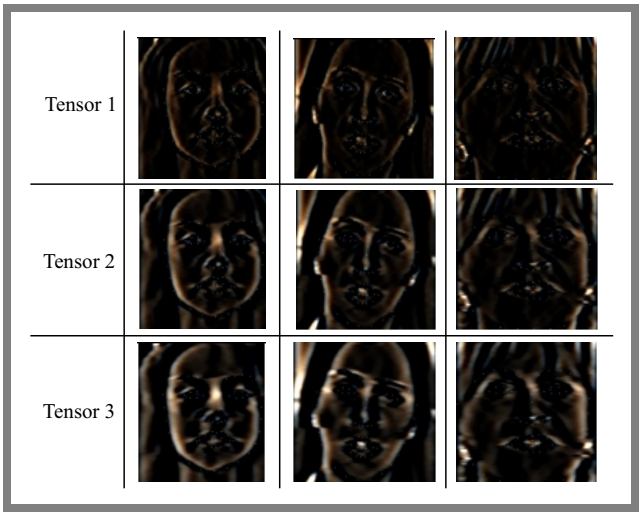

Fig. 8. The exemplary images represent combinations of transformed images as input tensors for the CNN ensemble classification system. The tensors were formed from three succeeding scales of CWT results.

The final experiments have been performed by application of the following input tensors composed of only magnitude representations superimposed with the face landmarks:

– tensor 1: scale 1 + scale 2 + scale 3,
– tensor 2: scale 2 + scale 3 + scale 4,
– tensor 3: scale 5 + scale 6 + scale 7.

Figure 8 shows the exemplary images forming the tensors defined in such a way.

4. Results of Numerical Experiments

In the numerical experiments, we have used the FaceForensics++ database [11] containing 1,000 original face images and 3,000 deep fake images, representing three different methods of manipulation (FaceSwap, FakeApp and Face2Face, each with a population of 1,000). For each pair of data (normal image and one type of a fake image), an individual ensemble comprising 8 CNN architectures, as presented in Fig. 7, was applied. The database was split into randomly chosen learning parts (70% of samples) and testing parts i.e., the remaining samples. The experiments were performed 10 times, with the content of learning and testing samples changing randomly. The paper presents and discusses the statistical results of tests performed on data sets that do not take part in the learning process. The accuracy (Acc.), precision (Prec.), recall (Rec.), and F1 measures will be presented for the three methods of creating fake images taken into consideration.

Table 1 shows results for deep fake images created with the use of the FakeApp algorithm. The results are related to the

Tab. 1. Statistical results related to the process of recognizing fake images created with the FakeApp algorithm, with three different methods of defining input tensor applied.

Metrics		Ens. [%]	Mean [%]	Std [%]	Best [%]	Worst [%]
Tensor 1	Acc.	90.16	78.78	5.00	87.43	78.33
	Prec.	91.18	86.42	7.38	90.16	76.53
	Rec.	90.17	86.98	2.38	88.82	84.66
	F1	90.67	86.70	3.59	89.47	84.60
Tensor 2	Acc.	91.33	87.11	8.87	90.22	82.71
	Prec.	92.61	85.65	8.97	91.09	81.89
	Rec.	91.34	79.25	6.55	87.17	78.67
	F1	91.97	82.31	7.57	89.09	80.25
Tensor 3	Acc.	**97.33**	94.39	6.71	96.79	92.23
	Prec.	**97.49**	85.50	7.41	95.10	84.28
	Rec.	**97.33**	94.67	5.33	96.73	93.33
	F1	**97.41**	89.85	6.20	95.91	88.57

Tab. 2. Statistical results related to the process of recognizing fake images created with the FakeSwap algorithm, with three different methods of defining input tensor applied.

Metrics		Ens. [%]	Mean [%]	Std [%]	Best [%]	Worst [%]
Tensor 1	Acc.	84.12	79.35	4.11	82.78	77.02
	Prec.	84.19	77.80	5.94	82.13	74.26
	Rec.	82.04	75.87	5.91	81.17	71.96
	F1	83.10	76.82	5.92	85.03	73.10
Tensor 2	Acc.	87.50	83.21	3.48	85.67	80.22
	Prec.	88.68	84.06	2.54	84.62	80.03
	Rec.	88.94	83.32	3.01	85.45	81.62
	F1	88.81	83.69	2.82	85.03	80.82
Tensor 3	Acc.	**93.16**	85.85	5.99	90.04	81.07
	Prec.	**92.67**	86.51	6.03	89.29	82.30
	Rec.	**93.60**	88.61	5.19	92.24	77.93
	F1	**93.13**	87.54	5.58	90.74	80.05

quality values of the integrated ensemble, the mean of non-integrated classification units, the standard deviation of the results of the individual units. The results of the best and worst members of the ensemble are presented as well. We observed a significant improvement in the integrated ensemble, compared to the mean of its non-integrated members, for all types of tensor representations. The best quality parameters (accuracy, precision, recall, and F1) have been observed for input attributes having the form of tensor 3. For example, the accuracy of the integrated ensemble increased, in that scenario, from the mean of 94.39% to 97.33%. Statistical results corresponding to the remaining methods of creating fake images (FaceSwap and Face2Face) are presented in Tabs. 2 and 3, respectively. In both cases, tensor 3 contained the best configuration of input attributes. It may be noticed that quality-related measures depend, to a considerable degree, on the fake image creation method relied upon. The worst results were achieved while using the Face2Face method. This is strictly related to the manner in which the manipulated images are generated. This method concentrates on a small portion of the face, without changing its general structure. Consequently, the manipulated images contain few changes compared with the originals. Therefore, they are very difficult to recognize in the subsequent frames of the video.

Tab. 3. Statistical results related to the process of recognizing fake images created with the Face2Face algorithm, with three different methods of defining input tensor applied.

Metrics		Ens. [%]	Mean [%]	Std [%]	Best [%]	Worst [%]
Tensor 1	Acc.	71.72	64.56	9.73	69.34	59.50
	Prec.	67.02	66.14	7.65	66.40	60.95
	Rec.	67.51	62.89	4.21	66.98	60.60
	F1	67.26	64.48	5.43	66.70	60.77
Tensor 2	Acc.	75.09	68.33	7.16	73.61	63.04
	Prec.	72.31	68.42	3.42	70.59	66.32
	Rec.	72.83	69.62	2.75	70.63	65.71
	F1	72.57	69.01	3.05	70.61	66.01
Tensor 3	Acc.	**83.50**	78.25	4.62	82.07	76.81
	Prec.	**82.68**	79.05	4.66	81.58	76.95
	Rec.	**81.15**	78.73	6.21	80.54	71.01
	F1	**81.91**	78.89	5.32	81.06	73.86

5. Comparative Analysis of Results

Detection of forged images has attracted a lot of attention in the past and different deep-learning approaches relying on various configurations of classifiers have been proposed in various papers. These include feedforward convolutional neural networks of different architectures (Xception, capsule, spatial transformer) [1], [4]–[7], [9], as well as recurrent LSTM and gated recurrent unit GRU [8], [10] approaches. Most of these papers focus solely on the accuracy of the image recognition process. The results depend on the classification algorithms applied, the type of algorithm used to create the fake image, and the compression ratio used in the image acquisition process, i.e., raw image, moderate compression c23, or high compression c40. The best accuracy values achieved while working on the same FaceForensics++ database are summarized in Tab. 4. Some papers specify the image compression method used, and some do not. Also, the types of image forgery algorithm used are not specified in some papers.

The declared accuracy for FakeApp manipulated images subjected to moderate compression (c23) changed from 98.10%

Tab. 4. Comparative results obtained for data taken from the FaceForensics++ database, with different fake image creation methods applied. C23 and c40 notations are used to identify images with average and high compression ratios applied, respectively, while "raw" means uncompressed images.

Paper	Recognition system	Method	Accuracy [%]
[5]	Capsule	FaceForensics++ (raw)	99.13
		FaceForensics++ (c23)	98.00
		FaceForensics++ (c40)	82.00
[23]	Optical net	FaceForensics++ (FakeApp)	98.10
[4]	Siamese CNN	FaceForensics++ (FakeApp)	87.15
		FaceForensics++ (Face2Face)	82.14
		FaceForensics++ (FaceSwap)	92.14
[1]	Xception CNN	FaceForensics++ (raw)	99.26
		FaceForensics++ (c23)	95.73
		FaceForensics++ (c40)	81.00
[6]	Mesoinception v4	FaceForensics++ (raw)	95.23
		FaceForensics++ (c23)	83.10
		FaceForensics++ (c40)	70.47
[10]	CNN + GRU + STN	FaceForensics++ (FakeApp)	96.90
		FaceForensics++ Face2Face)	94.35
		FaceForensics++ (FaceSwap)	96.30
[9]	CNN + biology	FaceForensics++ (FakeApp)	93.75
		FaceForensics++ (Face2Face)	95.25
		FaceForensics++ (FaceSwap)	96.25

[22] to 93.75% [9]. In the case of Face2Face forged images, the declared accuracy changed from 95.25% [9] to 82.14% [4]. The best result for FaceSwap manipulated images equaled 98.00% (as was presented in [5]), while the lowest value of 92.14% was achieved in paper [4].

The compression ratio of the images has a very high impact on the quality level achieved by the recognition system. This is due to the fact that the losses resulting from the compression procedure reduce the differences among images and make the recognition problem more difficult to solve. As some information is lost, small differences existing between the original and the forged images created with the use of the fake algorithm may not be noticed in the course of the direct analysis of the images. The application of CWT introduces more detail to the images at different resolution levels, thus enhancing these differences.

Tab. 5. Comparison of our best AUC values for the three fake image creation methods under consideration with the best results presented in [4].

	FakeApp	FakeSwap	Face2Face
Paper [4]	0.894	0.981	0.807
Our best value	0.954	0.916	0.848

The compression effect is well visible in papers [1] and [5]. The best result declared with no compression at all is 99.26% (with Face2Face and FaceSwap applied) in [1], or 99.13% in [5]. The experiments performed by the same authors at an average compression rate (notation c23) reduced these values to 95.73% [1] or to 98.00% in [5]. At very high compression rates (notation c40), the accuracy dropped to 81.00% in [1] and 82% in [5]. The most stable FakeApp, FaceSwap, and Face2Face results are observed for not-compressed images presented in [9], [10].

Since our data were created by applying moderate compression rates, the comparison of results will concentrate on this type of image presentation. We also include databases other than FaceForensics++. Paper [7] presented the results for a dataset composed of 5,000 short, 10-second clips, with an ensemble based on Xception + EfficientnetB3 + Attention applied. The results depend on the size of the data: accuracy Acc=92.20%, area under ROC curve AUC=0.975 for the basic set or Acc=93.64% and AUC=0.9841 for a data set that was 10 times larger. Paper [8] has applied a Resnet architecture CNN and a recurrent LSTM to detect deep fake videos. The accuracy obtained by the model over the Celeb-DF dataset was: Acc=91% and AUC=0.8880. In paper [4], accuracy depended on the type of the fake algorithm used. The best score of 92.14% was achieved for FaceSwap and the worst results of 82.14% for Face2Face, with the average AUC=0.89. Our best results obtained for the three types of fake algorithms under consideration are as follows: Acc=97.33% and AUC=0.954 for FakeApp, Acc=93.16% and AUC=0.916 for FakeSwap and Acc=83.50% and AUC=0.848 for Face2Face. Such results are comparable with or are among the best in the case of FakeApp manipulated images.

Figure 9 presents three ROC curves for the fake image creation methods under consideration and the AUC values associated therewith. They correspond to the best choice of input data in the form of tensor 3.

Table 5 shows the best values of AUC obtained with the use of our method for different fake algorithms. They are compared with the corresponding values shown in paper [4]. Our results are better for FakeApp and Face2Face and worse for FakeSwap.

6. Conclusions

The paper has presented a new approach to fake image detection. It is based on the application of CWT and an ensemble

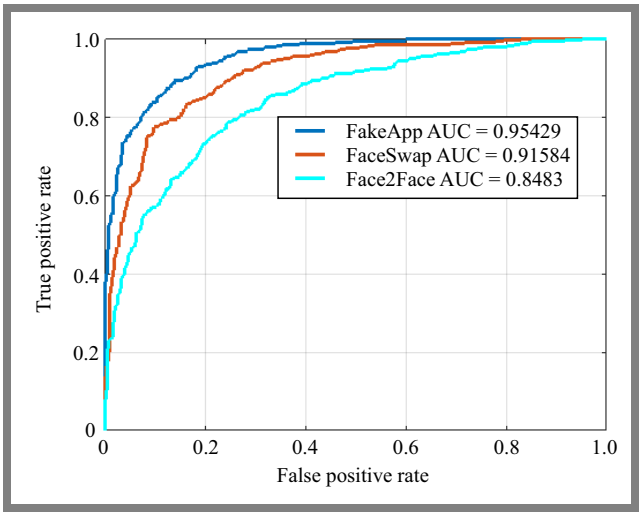

Fig. 9. Receiver operating characteristics (ROC) of the proposed system, corresponding to the best choice of input data (tensor 3).

of CNN architectures. The resulting CWT images constitute a basis for the creation of input tensors to this deep ensemble. Three forms of input tensors to the CNN classifiers have been studied and compared by applying the publicly available benchmark database of FaceForensics++ and using moderately compressed images (c23). In the training of the classification units of the ensemble, the transfer learning approach has been implemented. The results of numerical experiments have shown the superiority of the CNN ensemble over classification units working individually. For example, the best ensemble accuracy score for the FakeApp algorithm equaled 97.33%, while the mean for the individual classifiers (not integrated) was only 94.39%.

The important advantage of CNN classifiers is their relative insensitivity to the initial choice of parameters, such as starting filter weight values, the learning constant used during the adaptation process, random choice of connecting weights to softmax in each iteration (dropout procedure), etc. As a result, the adaptation process is based on the statistics of the learning samples, which are largely insensitive to the details of the learning data. The numerical experiments have shown high repeatability of results in different learning procedure runs, showing good generalization abilities of the system.

The obtained results are presented in the form of quality measures, such as accuracy, precision, recall, F1, and AUC, all estimated for testing data not taking part in learning. The experiments performed show that the method relied upon while creating fake images exerts a considerable impact on the quality of operation of the recognition system. The best results (the values of all quality measures above 97%) have been obtained with fake images created by means of FaceApp (Deepfakes). The remaining fake image creation methods (FaceSwap and Face2Face) have resulted in slightly worse quality values.

The proposed image processing system does not introduce significant limitations in terms of the size of images. However, the smaller the image the better results are expected, since CWT increases the number of details taking part in the recognition process. In the case of very large images, some limitations stemming from the amount of computer memory available might appear. The proposed system is relatively complex and requires a considerable amount of high-quality computation resources, especially with high FPS rates used in the videos processed.

Future investigations will focus on using other fake image creation methods, such as generative adversarial networks or the variational autoencoder. Also, more databases available to the public will be investigated and compared. Increasing the number of CNN units cooperating within the ensemble is an interesting research area as well. The key point is to accelerate the computation process and reduce the time within which the final decision is made by applying more advanced organizational structures of the computation system.

References

[1] A. Rossler *et al.,* "FaceForensics++: Learning to Detect Manipulated Facial Images", in: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Seoul, South Korea, 2019 (https://doi.org/10.48550/arXiv.1901.08971).

[2] L. Jaing, R. Li, W. Wu, C. Qian, and C.C. Loy, "Deeperforensics-1.0: a Large-scale Data Set for Real-world Face Forgery Detection", 2020 (https://doi.org/10.48550/arXiv.2001.03024).

[3] P. Yu, Z. Xia, J. Fei, and Y. Lu, "A Survey on Deepfake Video Detection", *IET-Biometrics*, vol. 10, no. 6, pp. 607–624, 2021 (https://doi.org/10.1049/bme2.12031).

[4] D. Cozzolino, G. Poggi, and L. Verdoliva, "Extracting Camera-based Finger Prints for Video Forensics", in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, Long Beach, USA, 2019.

[5] H.H. Nguyen, J. Yamagishi, and I. Echizen, "Capsule-forensics: Using Capsule Networks to Detect Forged Images and Videos", in: *2019 IEEE International Conference on Acoustics, Speech and Signal Processing*, Brighton, UK, pp. 2307–2311, 2019 (https://doi.org/10.1109/ICASSP.2019.8682602).

[6] D. Afchar, V. Nozick, J. Yamagishi, and I. Echizen, "MesoNet: A Compact Facial Video Forgery Detection Network", in: *2018 IEEE International Workshop on Information Forensics and Security (WIFS)*, 2018 (https://doi.org/10.48550/arXiv.1809.00888).

[7] S.H. Silva *et al.*, "Deepfake Forensics Analysis: An Explainable Hierarchical Ensemble of Weakly Supervised Models", *Forensic Science International: Synergy*, vol. 4, art. no. 100217, 2022 (https://doi.org/10.1016/j.fsisyn.2022.100217).

[8] S.S. Shet *et al.*, "Deepfake Detection in Digital Media Forensics", *Global Transitions Proceedings*, vol. 3, no. 1, pp. 74–79, 2022 (https://doi.org/10.1016/j.gltp.2022.04.017).

[9] U.A. Ciftci, I. Demir, and L. Yin, "FakeCatcher: Detection of Synthetic Portrait Videos Using Biological Signals", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 10, no. 10, 2020 (https://doi.org/10.1109/TPAMI.2020.3009287).

[10] E. Sabir *et al.*, "Recurrent Convolutional Strategies for Face Manipulation Detection in Videos", arXiv:1905.00582v3, 2019 (https://doi.org/10.48550/arXiv.1905.00582).

[11] FaceForensics. Database of FaceForensics++ [Online]. Available: https://github.com/ondyari/FaceForensics

[12] M. Massod *et al.*, "Deepfakes Generation and Detection: State-of-the-art, Open Challenges, Countermeasures, and Way Forward", *Applied Intelligence*, vol. 53, pp. 3974–4026, 2022 (https://doi.org/10.1007/s10489-022-03766-z).

[13] N. Dalal and B. Triggs, "Histograms of Oriented Gradients for Human Detection", in: *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, San Diego, USA, 2005 (https://doi.org/10.1109/CVPR.2005.177).

[14] J.J.V. Hernandez, J.I. de la Rosa, G. Rodriguez, and J.L. Flores, "The 2nd Continuous Wavelet Transform: Applications in Fringe

Pattern Processing for Optical Measurement Techniques", in: *Wavelet Theory and Its Applications*, IntechOpen, pp.173–193, 2018 (https://doi.org/10.5772/intechopen.74813).

[15] J. Brownlee, *Deep Learning for Natural Language Processing. Develop Deep Learning Models for Your Natural Language Problems*, Johns Hopkins University Press, Ebook, 372 p., 2018 (ISBN: 9781838550295).

[16] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, MIT Press, Massachusetts, 2016 (ISBN: 9780262035613).

[17] A. Krishevsky, I. Sutskever, and G.E. Hinton "ImageNet Classification with Deep Convolutional Neural Networks", *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017 (https://doi.org/10.1145/3065386).

[18] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition", 2015 (https://doi.org/10.48550/arXiv.1512.03385).

[19] A.G. Howard *et al.*, "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications", 2017 (https://doi.org/10.48550/arXiv.1704.04861).

[20] G. Huang, Z. Liu, L. van der Maaten, and K.Q. Weinberger, "Densely Connected Convolutional Networks", 2018 (https://doi.org/10.48550/arXiv.1608.06993).

[21] F.N. Iandola *et al.*, "SqueezeNet: AlexNet-level Accuracy with 50x Fewer Parameters and <0.5MB Model Size", 2017 (https://doi.org/10.48550/arXiv.1602.07360).

[22] X. Zhang, X. Zhou, M. Lin, and J. Sun, "ShuffleNet: an Extremely Efficient Convolutional Neural Network for Mobile Devices", 2017 (https://doi.org/10.48550/arXiv.1707.01083).

[23] Y. Zhao *et al.*, "Capturing the Persistence of Facial Expression Features for Deep Fake Video Detection", in: *International Conference on Information and Communications Security*, Beijing, China, pp. 630–645, 2019 (https://doi.org/10.1007/978-3-030-41579-2_37).

Stanislaw Osowski, Prof.
Institute of Theory of Electrical Engineering, Measurement, and Information Systems
https://orcid.org/0000-0003-3194-4656
E-mail: stanislaw.osowski@wat.edu.pl
Military University of Technology, Warsaw, Poland
https://www.wat.edu.pl
Warsaw University of Technology, Warsaw, Poland
https://www.pw.edu.pl

Maciej Golgowski, Ph.D.
Electronic Faculty, Institute of Electronic Systems
E-mail: maciej.golgowski@wat.edu.pl
Military University of Technology, Warsaw, Poland
https://www.wat.edu.pl