

Accurate and Robust Neural Networks for Security Related Applications Exemplified by Face Morphing Attacks

Clemens Seibold¹, Wojciech Samek¹, Anna Hilsmann¹ and Peter Eisert^{1,2}

¹ Fraunhofer HHI, Einsteinufer 37, 10587 Berlin, Germany

² Humboldt University Berlin, Unter den Linden 6, 10099 Berlin, Germany

Abstract. Artificial neural networks tend to learn only what they need for a task. A manipulation of the training data can counter this phenomenon. In this paper, we study the effect of different alterations of the training data, which limit the amount and position of information that is available for the decision making. We analyze the accuracy and robustness against semantic and black box attacks on the networks that were trained on different training data modifications for the particular example of morphing attacks. A morphing attack is an attack on a biometric facial recognition system where the system is fooled to match two different individuals with the same synthetic face image. Such a synthetic image can be created by aligning and blending images of the two individuals that should be matched with this image.

Keywords: Face image forgery detection, attacks on convolutional neural networks, convolutional neural network analysis.

1 Introduction

Biometric verification systems are nowadays present in many fields of daily life. They are used in the consumer market, e.g. to unlock a mobile phone, as well as for sovereign tasks like automatic border control. One huge advantage and a reason for their success is that every individual carries his or her biometric characteristics always with them and that they are difficult to copy by another individual. Recently, the vulnerability of facial recognition systems against a specific attacking method was made known [1]. This attack, which is called morphing attack, tricks a facial recognition system into matching two different individuals with one forged facial image. The detection of this kind of fraud is substantial to ensure the trustworthiness, especially for security-related tasks like entrance control.

Recently, several attacks on the prediction of neural networks with only small perturbations of the content and without knowledge of the weights or architecture of the network were published. This represents a danger for neural networks in many applications, especially for security related applications where the network should be accurate and robust against specific attacks on their decision

making process. By the example of morphing attack detection, we present in this paper different ways to evaluate the properties of neural networks, e.g. robustness against attacks, and the effect of different modification of the training data on these properties. We train VGG19 [2] networks to detect morphing attacks and differ the amount and location of information that can be used by the network to detect the forgeries. The differently trained networks are analyzed regarding their accuracy and two different kinds of attacks on the decision making. One attack scenario represents a semantic attack where the forger leaves only a few traces in the image so that the network needs to be able to recognize every kind of artifacts to detect a morph. In the other scenario, we assume that the attacker has access to the system as a black box and can test the system as often as he likes. In order to get insights on the decision making of the different networks and to understand the effects of the training data alteration, we use layer-wise relevance propagation [3].

In the next section, we give a summary of the recent studies on face morphing attacks and attacks on neural networks. Our fully automatic morphing pipeline is presented in section 3, followed by details about our self-made and collected face and morphing attack database in section 4. The setup of our experiments and thus the generation of our specifically altered training data is described in section 5. The analysis methods and results are presented in sections 6 and 7.

2 Related Work

The vulnerability of facial recognition systems was first published by Ferrara et al. [1]. They created morphing attacks by manually aligning facial images of different individuals and showed that they can trick commercial facial recognition systems to verify both individual with the forged face image.

Several approaches for morphing attack detection were developed based on different techniques. Raghavendra et al. [4] extracted facial micro-textures using Binarized Statistical Image Features to distinguish between morphing attacks and genuine images. The morphing detection method of Makrushin et al. [5] is based on JPEG double compression. Neubert studies in [6] image degeneration to detect morphing attacks. Morphing detectors based on CNNs were successfully used by Seibold et al. [7] and Raghavendra et al. [8]. Ferrara et al. [9] went even further and proposed a demorphing method, which is capable of extracting one genuine face of a morphing attack, if an image the other face is known.

The vulnerability of neural networks against small perturbations in the image was revealed by Szegedy et al. [10]. They also showed that the required perturbations are not of random nature and depend on the dataset. To get these perturbation they used a gradient-based method and backpropagation. Goodfellow et al. [11] proposed the fast gradient sign attack that requires only the sign of the gradients of the required perturbation to perform the attack. Papernot et al. [12] published a method to estimate the signs of these gradients without having access to the weights or architecture by training a substitute model that behaves similar to the network that should be fooled. To do so, they needed only access

to the final output of the network for generated images. Thus, they provided a framework to use the fast gradient sign attack as a black box attack. Other black box attacks that also try to model the network’s behavior by questioning the network were published by Narodytska and Kasiviswanathan [13] and Dang et al. [14]. Su et al. [15] showed that even a larger modification of only one pixel is sufficient to change the prediction of the network.

3 Forged Facial Image Generation

3.1 Overview

Since we need a lot of morphing attacks to train our networks, we implemented a fully automatic morphing pipeline similar to the face morphing pipelines introduced in [5, 7]. The idea of all of these pipelines is to create a facial image that contains characteristics of two different individuals. They all start with an alignment of two facial images and use additive alpha blending to generate the forgery. Their differences are mostly in the processing of regions that are not part of the inner face. Figure 1 provides an overview about our face morphing pipeline. Our alignment and blending is presented in the following.

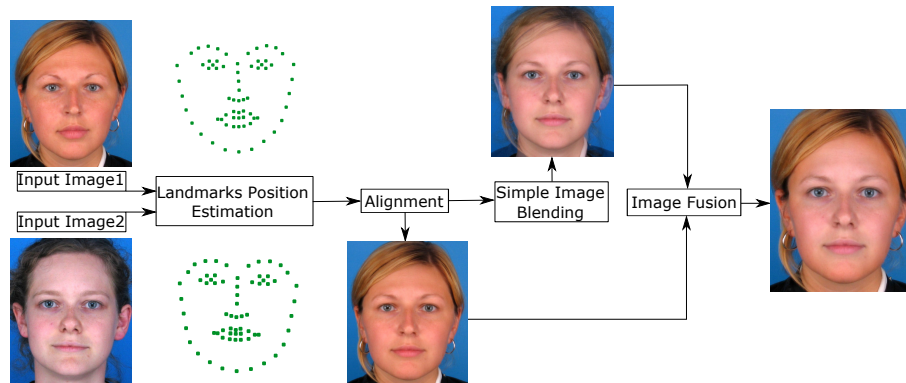


Fig. 1. Morphing Pipeline.

3.2 Image Alignment

We align the facial images based on predefined facial landmarks whose positions we estimate using dLib’s [16] implementation of the facial feature detector described in [17]. The images are warped such that the facial landmarks of both images are at the same position and the regions between landmarks are transformed similar to the nearest facial landmarks. The target position of a landmark is its average position of both input images.

In the following, we describe two different warping methods, whereby we refer to the original image as source image and to its warped version as target image and accordingly to the landmark positions as source positions and target positions. Both approaches use different numbers of facial landmarks. We add some additional landmarks for the triangle warp, whereas the field warping approach only uses a small subset of the estimated landmarks. We used two different kinds of warping methods to increase the variety in data generation and reduce a possible impact of the warping algorithm on our experiments.

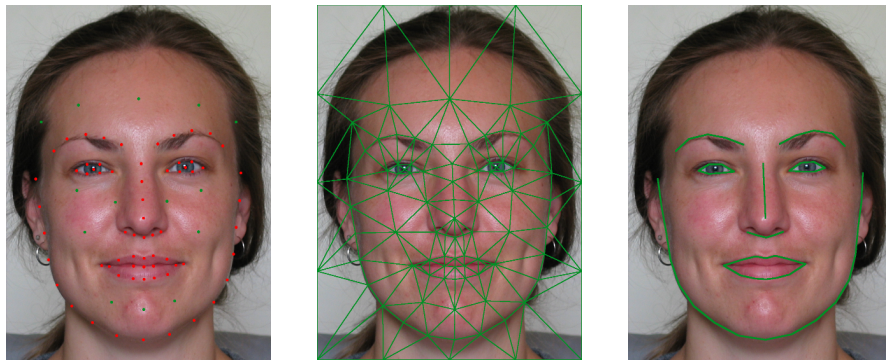


Fig. 2. Basis of our image warping methods: landmarks estimated by dlib in red and additional added landmarks in green (left), triangle mesh for triangle warp (center), lines for field morphing (right).

Triangle Warp We add and fuse some landmarks to get a triangle mesh that includes the whole image and to avoid degenerated triangles. The additional landmarks are at the borders of the images, in the center of each eye, on the cheeks, between mouth and chin and on the forehead, as shown in figure 2. The points on the upper and lower line in the center of the lips, which were estimated by dLib, are fused, since they are very close. After adding and removing the feature points, we apply the Delaunay triangulation on the target point set to obtain a mesh topology. This topology is then applied on the source point set. Now, having a triangle mesh with the same topology on each image, we warp the region of each triangle in the source image to the region of the triangle in the target image using an affine transformation.

Beier-Neely field morphing The Beier-Neely field morphing algorithm [18] requires corresponding line segments between the images to define a transformation. Since we only estimate a set of points, we connect some of them with a fixed predefined pattern, as shown in figure 2, to be able to apply the field morphing algorithm. The concept of this morphing method is to transform each pixel relative to the transformation of each line segment. The closer a line segment is to a

pixel, the stronger is its impact on the pixel’s transformation. The pixel position of a target pixel in the source image is calculated by applying the transformation for each line pair individually, adding the weighted displacement vector and normalizing it by the sum of weights. For one line segment Q in the target image with end points Q_s, Q_e with Q_n being the normal of the line segment, a pixel \mathbf{p} can be parametrized with two parameters u, v by

$$\mathbf{p} = u\|Q_e - Q_s\| + Q_s + Q_n v. \quad (1)$$

The pixel position \mathbf{p}' in the source image can now be calculated by

$$\mathbf{p}' = u\|Q'_e - Q'_s\| + Q'_s + Q'_n v, \quad (2)$$

with Q'_e, Q'_s, Q'_n being the end points and normal of the corresponding line segment in the source image. The impact or weight of a line on a pixel is defined with respect to their distance in the target image by

$$w(\mathbf{p}, Q) = \left(\frac{1}{d + \varepsilon} \right)^2, \quad (3)$$

with d being the shortest distance from the pixel \mathbf{p} to the line Q .

3.3 Image Blending

Since most of our facial landmarks are located in the face and not around the head, ears or hair, the alignment procedure takes only the inner part of the face below the forehead into account. A simple blending of the whole image would thus lead to strong visible ghosting artifacts around the silhouette, hairstyle, ears and, if included, shoulders. To avoid this striking forgery indicators, we used the blending method proposed by Seibold et al.[7]. We blend only the inner part of the face and use one aligned input image for the outer part. As a consequence, we have to calculate a smooth and inconspicuous transition from the blended image to the warped original one. The area of this transition is defined as region between two ellipses with the same center. Their width and height are determined relative to the estimated facial landmark positions. As the authors of [7] do, we also separate the images in a high spatial frequency and a low spatial frequency image and calculate for both of them a different transition between the inner and the outer part. In this way, we can avoid cuts through high frequency differences in the images and low frequency differences caused by slightly different skin colors or different illuminations can be handled by a smooth and small color gradient over this zone. The transition for the low spatial frequency part is calculated using Poisson Image Editing [19]. The gradients for the Poission equation in the transition zone are taken from the blended image. For the high spatial frequency part of the images, we calculate a cut through the transition area that does not create high frequencies around the cut. For that, we use the graph cut algorithm with the penalty term used in [7] that prefer cuts through regions with similar gradients. As the authors of [7], we apply this algorithm on images in polar coordinates, so that shorter parts around the inner border do not have lower cost due to less cuts.

4 Training Dataset

4.1 Database

To get a representative database, we acquired several freely-available face databases and captured face images by ourselves. All of these images were manually checked whether they are suitable for passports and images were removed in which the person looked not directly into the camera, had the eyes closed, the mouth opened or other requirements for passports were strongly violated. We gathered about 1,900 images of different individuals. Table 1 provides an overview about the image sources and number of used images per database.

We divide our set of original images into a training set that contains 80% of

Table 1. Face image sources.

Database	BU-4DFE[20]	CFD[21]	FEI[22]	FERET[23]	PUT[24]
#used images	101	597	100	459	100

Database	scFace[25]	Utrecht[26]	own	various
#used images	108	64	261	89

all original images, a testing set (15%) and a validation set (5%). This distribution into training, testing and validation sets hold approximately also for each individual database. Since there is obviously a strong dependency in the data between original images and morphs, we used for the generation of morphing attacks for the training, testing or validation set only images from the same set. To ensure a good quality and variety of the morphs, we made some guidelines for the selection of pairs of original images for morphing attacks:

1. both individuals have the same gender
2. 50% of the morphs use the triangle approach and 50% the field morphing
3. both images are from the same data base
4. each image is approximately used equally frequent to create a morph

4.2 Preprocessing

The generation of a morph requires a lot of image manipulations that change characteristics like gradient distribution, correlation between neighbored pixels and so on. This creates a lot of low-level artifacts that might be used by the network to discriminate a morph from an original image [7]. Since this characteristics depend strongly on the input data, we apply some kinds of noise and blur to change this distributions in the original and forged face images to prevent the network from discriminating on these characteristics and to focus on more semantic high-level models. For each original image and for each morphing attack in a set, we added two different kinds of noise and applied two different kinds of blurring filters. Thereby, each image is added in five different versions to

Table 2. Parameters for data preprocessing.

Kind of Preprocessing	Parameter Range
motion blur	length between 0.5% and 1.0% of image height, random angle
Gaussian blur	sigma between 0.25% and 0.5% of image height
Salt-and-pepper noise	1% of all pixels
Gaussian noise	standard deviation of 0.05 (pixel intensities in [0,1])

one set: one version without any processing and one each after applying motion blur, Gaussian blur, Salt-and-pepper noise or Gaussian noise. The parameters for the different kinds of blur and noise vary for each image as shown in table 2. To remove unnecessary variance in the input data and have a standardized input for the network, we rotate, rescale and crop the images before feeding them into the network. Each image is rotated such that the eyes are at the same height and finally scaled and cropped to a size of 224×224 such that it contains the region between eye brows and mouth and between the outer pars of the eyes. During the training, this region is shifted slightly to augment the training data.

5 Experiment Settings

We start our training in all cases with a VGG19 network [2] that was pretrained on the ILSVRC challenge. Although a naive training with genuine images and morphing attacks leads to a quite good accuracy, its robustness against specific attacks is rather poor, as shown in the following sections. In order to train a network that is accurate and robust, we use different alterations of our training data. Our key idea is to change the amount, location and kind of information that is available to detect a morph. In this way we force the network to consider all regions of the face during the training. For this purpose, we use partial morphs. A partial morph is a forged facial image in which some regions are taken from a morphed image and the rest from an aligned input image. We defined four different regions in the face for the generation of our partial morphs. These regions are the region around the right or left eye containing the eyebrows, around the nose and around the mouth, see also figure 3. For each morphing attack, the regions were automatically estimated relative to the facial feature points estimated by dLib.

In the following, we present the four different kinds of training data manipulation that were used in our experiments. In three cases, we perform a two class classifier training. If an image has at least one forged region, the whole image is assigned to the class morphing attack. In the fourth case, we first do a multiclass training with four different classes. Each class represents one region and if this region is forged, the image belongs to this class. To get a binary decision, we remove the last fully connected layer and train the last layer and with a smaller learning rate the second last layer for the binary class problem using the complex morphs training method described below.

The relative amount of morphs and genuine image is designed such that each



Fig. 3. Regions for partial morphs.

class occurrences with equal frequency. The distribution of the different types of morphs and the amount of genuine images for the different training methods is as follows:

Naive training: 50% genuine images, 50% complete morphs

One region: 50% genuine images, 10% complete morphs and $4 \times 10\%$ with one region morphed

Complex morphs: 50% genuine images, 10% complete morphs, partial morphs with 10% one, two, three and four region morphed (each region is morphed with equal frequency)

Multiclass (Complex morphs): partial morphs with zero, one, two, three or four morphed regions (each 20% and each region is morphed with equal frequency); A two class classifier is obtained by reinitializing the last fully connected layer and training it with the same data as used in the complex morphs training method.

As mentioned above, the aim of our training data modification is to force the network to take all available information for its decision making into account. In case of a naive training it might be sufficient to focus on the eyes to detect morphing attacks, but there are still other relevant information available that were ignored by the network. In case of partial morphs the network has to gather information from all region in the face to make a correct class prediction. One disadvantage of the one region morphed training method is that less artifacts are presented to the network during the training, if we keep the amount of genuine images and forgeries identical. The complex morphs training method overcomes this disadvantage partly, but again the network does not have to gather all available information in one morph to classify it correctly. The multiclass complex morphs training method allows to present more artifacts and keeps the number of images that are in one class identical to the images that are not in this class. This way, we can present more morphing artifacts to the network without creating a class bias and can still force the network to use all available information about morphing artifacts in one image.

6 Analysis Methods

We analyze the differently trained networks based on the accuracy and different measurement for robustness. Especially a network for forgery detection should be robust against all kind of attacks on its decision. Thus, we study two different problem specific attacks, in which the attacker knows nothing about the forgery detection system but tries to manipulate the images as little as possible and as good as possible to leave no suspicious traces, and general attacks on neural networks, where the attacker has access to the system as a black box.

Under the assumptions, that a professional tries to avoid artifacts and leaves only some of them, the detection of partial morphs seem to be a good criteria for robustness.

In another attack scenario, the attacker has access to the system and can test the forged images as often as possible. Thus, he can attack the decision of the network directly by creating adversarial examples that differ only slightly to the created morphing attack but are classified as genuine image by the neural network.

We modeled this attack using the fast gradient sign method [11] and generated the adversarial examples using a substitute model [12]. This model is trained on our validation data set and on adversarial examples that were generated as in [12]. Since we had no success in fooling our networks using the shallow networks in [12], we use a pretrained GoogLeNet [27] instead.

Beside our accuracy and robustness analysis, we try to get some insights on the networks decision making and whether or how they differ [28]. For thus purpose, we use the LRP-Toolbox [29] to mark regions in the image that were relevant for the network’s decision. Over-simplified described, a region is relevant, if at least one path from a neuron in the last layer that defines the class affiliation to this region exists, and the neurons along this path have a strong activation and large weights. We apply the LRP-toolbox on partial morphs to check if the relevance is mostly in the morphed region and thus the classification was done by detecting morphing artifacts. We use the epsilon decomposition as relevance propagation method for the fully connected layers. The relevance of the later convolutional layers was propagated using the $\alpha - \beta$ - decomposition with an α of 2 and β of -1 and the relevance of the first convolutional layers using flat weight decomposition. We consider only forgeries that were classified as such or with an output at the softmax for the class "morphing attack" was of least 0.1 so that there is relevance to propagate to the image.

7 Results

In the following, we present the accuracy and robustness that is achieved with the different kinds of training methods and discuss the insight we got using relevance propagation.

Accuracy We refer to the relative amount of correctly classified genuine images as true positive rate and to the relative amount of correctly classified morphing attacks as true negative. The best true positive rate was achieved using the naive training method. This might be due to the fact that genuine images and morphing attacks share the same content for partial morphs which were used in the other training methods. All networks were better in classifying morphing attacks than genuine images. By moving the threshold for the classifier, we could counter this bias and got for the naive and the multiclass complex training method the best equal error rates, whereas the EERs for the other two training methods were at least twice that much.

Table 3. Accuracy of different training methods.

	naive	one morphed	complex morphs	multiclass
true positive	95%	90%	93%	92%
true negative	98%	95%	95%	99%
EER	3.1%	7.2%	6.1%	2.8%

Semantic Attacks Table 4 shows the true negative rates of our differently trained networks for different kinds of partial morphs, where the attacker blends only the region around right or left eye, the nose or the mouth. Our naive trained network performs quite poor for this scenario, whereas all other training methods perform significantly better. The detection rate for the same network but for different kinds of partial morphs differ significant for some networks. The naive training method performs worse in relative terms for partial morphs of the nose and mouth, whereas the other three methods perform only worse on the mouth than on partial morphs of other regions.

Table 4. Robustness against partial morphs.

	left eye	right eye	nose	mouth	average
naive	25%	21%	14%	13%	20%
one morphed	81%	89%	79%	71%	80%
complex morphs	78%	74%	73%	54%	70%
multiclass	86%	93%	90%	79%	87%

Black Box Attacks The other attack scenario we study is a black-box attack using adversarial examples. As mentioned above, we use the fast gradient sign method and a GoogLeNet for the substitute model to generate adversarial examples instead of the shallow architectures as in the original paper. Since an attacker would of course try to modify a morphing attack to let it be classified as an genuine image and not the other way around, we focus only on the classification results for morphing attacks. Figure 5 shows the robustness against this

kind of attack. All networks were vulnerable against this attack with a strong modification in the image, but especially for small and for the human eye not visible changes their vulnerability differ quite strong. The network that was trained using the naive method is the most vulnerable, the 2-classes training methods with partial morphs are more robust, but still perform worse than the multi-class training method. Even with a change of the intensities by ± 6 , which leads to a visible noise-like pattern (see figure 4), still about 91% of the adversarial examples are correctly classified.

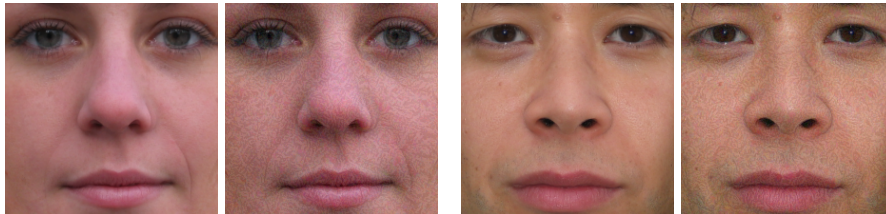


Fig. 4. Adversarial examples created for the naive trained network. The left image is in both cases the original image and the right image the adversarial attack with a change of the intensities of 6. The attack on the right was successful, whereas the adversarial attack on the left was still classified as fraud.

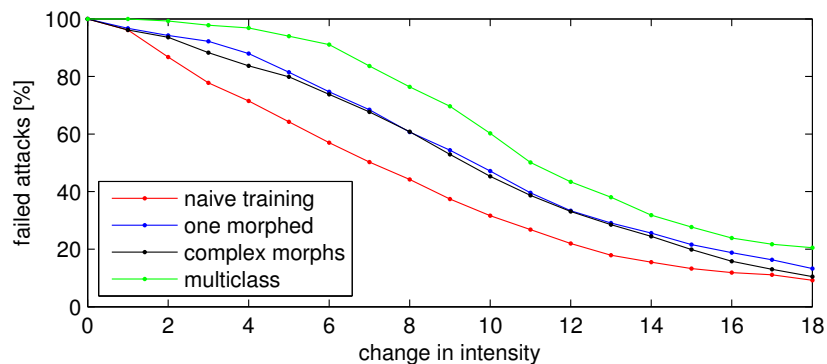


Fig. 5. Robustness against fast gradient sign attacks.

Relevance Distribution Whereas the overall accuracy of our networks differs only less than 5 percent, we got quite different results for the robustness against partial morphs. The robustness also differs strongly for different kinds of partial morphs for the same network. Using relevance propagation, we try to get some

insights about this behavior. We applied the relevance propagation on partial morphs and only if the softmax output of the network for the class morphing attack was at least 0.1, since if there is no or only little relevance in the neuron for the class "morphing attack", there will be no or only noisy relevance for this class in the image. Some neurons in the network are always active and thus lead to relevance that is always present. To overcome this problem, we subtracted the average relevance for each pixel. Finally, we removed all negative relevance, since negative relevance can be interpreted as inhibiting the decision for this class. The relevance for the three networks that were trained directly on the binary decision is mostly in the morphed region as one would expect (see table 5 and figure 6). For these three networks, a strong correlation between the location of the relevance and the detection of the partial morphs can be found. For instance, the naive trained network performs worse on partial morphs of the nose and mouth and the relevance for partial morphs of the nose or mouth are to a large extend outside of the morphed region. The same behavior can be observed for the training method with complex morphs and partial morphs of the mouth.

Table 5. Relevance distributions.

morphed region	relative amount of relevance per region							
	naive				one morphed			
	left eye	right eye	nose	mouth	left eye	right eye	nose	mouth
left eye	0.84	0.00	0.02	0.14	0.96	0.00	0.01	0.04
right eye	0.00	0.91	0.05	0.05	0.00	0.92	0.01	0.07
nose	0.21	0.28	0.47	0.04	0.00	0.01	0.97	0.02
mouth	0.34	0.27	0.04	0.35	0.17	0.12	0.04	0.68
	complex morphs				multiclass			
	left eye	right eye	nose	mouth	left eye	right eye	nose	mouth
left eye	0.98	0.00	0.00	0.02	0.00	0.98	0.00	0.01
right eye	0.00	0.92	0.00	0.08	0.98	0.00	0.02	0.00
nose	0.02	0.03	0.92	0.02	0.01	0.10	0.19	0.70
mouth	0.06	0.00	0.41	0.53	0.11	0.18	0.58	0.13

The relevance distribution indicates that the model for forged eyes of the naive network is not as bad as it seems when looking at the robustness against partial morphs. The bad robustness might be due to the fact that the naive trained network was trained only on complete morphs and genuine images and thus has no knowledge about how to classify an image that contains characteristics of both classes. On the other hand, the relative bad robustness against partial morphs of the nose and mouth and the bad distribution of relevance for these morphs suggests that the naive trained network represents these parts only weakly. This is not surprising, since the eyes might be sufficient to distinguish between morphs and genuine images in most cases.

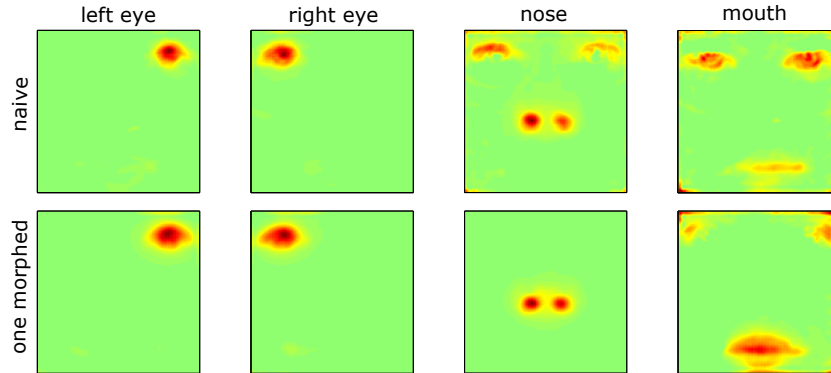


Fig. 6. Average mean adjusted positive relevance distribution for the naive and one morphed training method grouped by the different kinds of partial morphs.

Relevance Distribution Interpretation for Multiclass Training The relevance distribution for the multiclass training methods seems at the first glance completely misleading, but might point out to a network that compares different structures. A structure in the neural network that explains this kind of relevance distribution can, for instance, rely mainly on a model for genuine images and be connected as follows. Assuming we have two neurons that have a strong activation, if the eyes of the individual in the image looks like a male Caucasian in his late twenties. One neuron represents the left eye and the other one the right eye. If one of the eyes does not fit to the model, e.g., due to a morphing artifact, it has a low activation. If we subtract the activation of one neuron from the activation of the other one, we get either a large negative or positive activation. The large negative activation would be clamped by the rectified linear unit, while the position activation would remain and indicate a morph. As a result, we have a path of strong activation from this neuron, to the area in the input image around the genuine eye and thus relevance in its region. Thus, the absence of relevance in a region can also indicate the importance of this region. For comparisons with feature maps that show a strong activation if a region is morphed, we can still expect that the relevance for a genuine region is different than if this region is forged.

Figure 7 shows the relevance distribution of different partial morphs of the same pair of input images for the multiclass training method. In cases of one morphed region (upper images), there is nearly no relevance in the morphed region, but negative or positive relevance in the other parts of the image. If all but one region was morphed (lower images), there is again nearly no relevance in the forged parts, but a lot of relevance in the original part of the image. Some relevance is also assigned to the left eye if this region is forged, but it is much less than if this region is genuine.

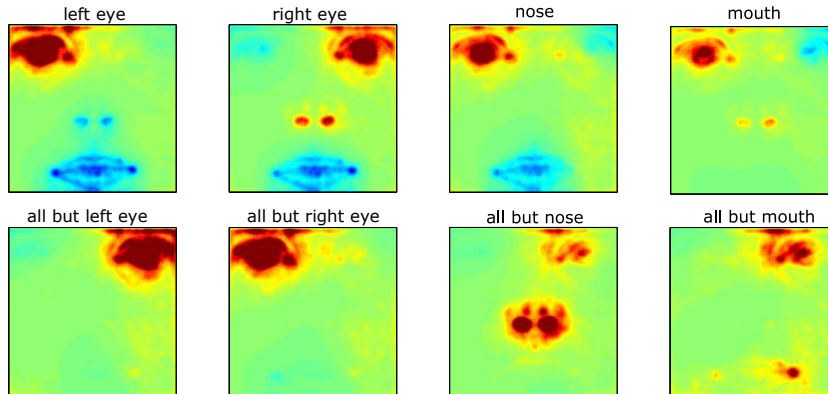


Fig. 7. Relevance distribution for the multiclass training of different kinds of partial morphs based on one input image pair.

8 Conclusion

In this paper, we presented different training methods based on modifications of the training data for the example of face morphing attack detection. We analyzed the resulting networks regarding accuracy, robustness and got insights about differences in their decision making.

We showed that accuracy does not always comes along with robustness against attacks on the network. Using partial morphs, in which only a part of the image is forged, during training, we enhanced the network to detect morphs also if only a part of the image was forged and we enhanced also the robustness against adversarial attacks. Our analysis on the decision making process of the networks indicates that the naive trained network focuses a lot on the region around the eyes and gives less priority to the region around the nose and mouth which leads to a weak distinctive model for these regions. The networks that were trained with partial morphs tend to have better models for all regions of the face, but have a worse accuracy. We overcome this problem by a complex multiclass pre-training by which we got a robust and accurate network. Our analysis of the decision making process of this network indicates that it compared different region among each other to detect morphing attacks, whereas this comparison structure was not visible for the other training methods.

In future work we will combine the complex multiclass pretraining with defence mechanisms against adversarial attacks [30] to further robustify the neural network models.

Acknowledgements. This work was supported by the German Federal Ministry of Education and Research (BMBF) through the Research Program ANANAS under Contract No. FKZ: 16KIS0511, and by the Fraunhofer Society through the MPI-FhG collaboration project “Theory & Practice for Reduced Learning Machines”.

References

1. Ferrara, M., Franco, A., Maltoni, D.: The magic passport. In: IEEE International Joint Conference on Biometrics. (2014) 1–7
2. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. CoRR **abs/1409.1556** (2014)
3. Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K.R., Samek, W.: On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. PLOS ONE **10**(7) (2015) e0130140
4. Raghavendra, R., Raja, K.B., Busch, C.: Detecting morphed face images. In: IEEE 8th International Conference on Biometrics Theory, Applications and Systems (BTAS). (2016) 1–7
5. Makrushin, A., Neubert, T., Dittmann, J.: Automatic generation and detection of visually faultless facial morphs. In: Proceedings of the 12th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2017) - Volume 6: VISAPP, Porto, Portugal, February 27 - March 1, 2017. (2017) 39–50
6. Neubert, T.: Face morphing detection: An approach based on image degradation analysis. In: Digital Forensics and Watermarking - 16th International Workshop, IWDW 2017, Magdeburg, Germany, August 23-25, 2017, Proceedings. (2017) 93–106
7. Seibold, C., Samek, W., Hilsmann, A., Eisert, P.: Detection of face morphing attacks by deep learning. In: Digital Forensics and Watermarking - 16th International Workshop, IWDW 2017, Magdeburg, Germany, August 23-25, 2017, Proceedings. (2017) 107–120
8. Raghavendra, R., Raja, K.B., Venkatesh, S., Busch, C.: Transferable deep-cnn features for detecting digital and print-scanned morphed face images. In: IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). (2017) 1822–1830
9. Ferrara, M., Franco, A., Maltoni, D.: Face demorphing. IEEE Transactions on Information Forensics and Security **13**(4) (2018) 1008–1017
10. Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., Fergus, R.: Intriguing properties of neural networks. In: International Conference on Learning Representations. (2014)
11. Goodfellow, I., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. In: International Conference on Learning Representations. (2015)
12. Papernot, N., McDaniel, P., Goodfellow, I., Jha, S., Celik, Z.B., Swami, A.: Practical black-box attacks against machine learning. In: Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security. ASIA CCS '17 (2017) 506–519
13. Narodytska, N., Kasiviswanathan, S.P.: Simple black-box adversarial attacks on deep neural networks. In: IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). (2017) 1310–1318
14. Dang, H., Huang, Y., Chang, E.C.: Evading classifiers by morphing in the dark. In: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security. CCS '17 (2017) 119–133
15. Su, J., Vargas, D.V., Sakurai, K.: One pixel attack for fooling deep neural networks. CoRR **abs/1710.08864** (2017)
16. King, D.E.: Dlib-ml: A machine learning toolkit. Journal of Machine Learning Research **10** (2009) 1755–1758

17. Kazemi, V., Sullivan, J.: One millisecond face alignment with an ensemble of regression trees. In: IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). (2014) 1867–1874
18. Beier, T., Neely, S.: Feature-based image metamorphosis. In: Proceedings of the 19th Annual Conference on Computer Graphics and Interactive Techniques. SIGGRAPH '92 (1992) 35–42
19. Pérez, P., Gangnet, M., Blake, A.: Poisson image editing. In: ACM SIGGRAPH 2003 Papers. SIGGRAPH '03 (2003) 313–318
20. Yin, L., Chen, X., Sun, Y., Worm, T., Reale, M.: A high-resolution 3d dynamic facial expression database. In: 2008 8th IEEE International Conference on Automatic Face Gesture Recognition. (2008) 1–6
21. Ma, D., Correll, J., Wittenbrink, B.: The chicago face database: A free stimulus set of faces and norming data. *Behavior Research Methods* **47** (2015)
22. Thomaz, C.E., Giraldi, G.A.: A new ranking method for principal components analysis and its application to face image analysis. *Image and Vision Computing* **28**(6) (2010) 902–913
23. Phillips, P.J.: Color feret database
24. Kasiski, A., Florek, A., Schmidt, A.: The put face database. *Image Processing and Communications* **13** (2008) 59–64
25. Grgic, M., Delac, K., Grgic, S.: Scface — surveillance cameras face database. *Multimedia Tools and Applications* **51**(3) (2011) 863–879
26. Hancock, P.: Utrecht ecvp
27. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: IEEE Computer Vision and Pattern Recognition (CVPR). (2015) 1–9
28. Samek, W., Wiegand, T., Müller, K.R.: Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models. *ITU Journal: ICT Discoveries - Special Issue 1 - The Impact of Artificial Intelligence (AI) on Communication Networks and Services* **1**(1) (2018) 39–48
29. Lapuschkin, S., Binder, A., Montavon, G., Müller, K.R., Samek, W.: The layer-wise relevance propagation toolbox for artificial neural networks. *Journal of Machine Learning Research* **17**(114) (2016) 1–5
30. Srinivasan, V., Marban, A., Müller, K.R., Samek, W., Nakajima, S.: Counterstrike: Defending deep learning architectures against adversarial samples by langevin dynamics with supervised denoising autoencoder. *arXiv:1805.12017* (2018)