Explainable Visualization for Morphing Attack Detection

Henning Myhrvold[1]

Abstract: Detecting morphed face images has become critical for maintaining trust in automated facial image verification systems. Researchers have discovered that deep facial recognition systems intended for generalizability increased their vulnerability to exploitation. Attacks based on altered face images offer a significant security concern. A morphing attack is a method of fooling a biometric facial recognition system into matching two distinct individuals using the same synthetic face image. Deep Neural Networks have demonstrated good performance in detecting morphed images. However, they lack transparency, and it is unclear how they differentiate between authentic and morphed facial photos. As a result, this phenomenon needs careful consideration for safety and security-related applications. This paper will explore layer-wise relevance propagation for determining the most relevant input features. We fine-tune a CNN for face morphing attack detection. LRP is then used to investigate the decision-making processes and see what input pixels the neural network considers visually. This paper shows that CNN only considers a small part of the image, usually around the eyes, nose, and mouth.

Keywords: Face morph; Morph Attack Detection; Layer-wise Relevance Propagation; VGG19; Deep Neural Network; Convolutional Neural Network; APCER; BPCER; Confusion Matrix

1 Introduction

Face recognition systems (FRS) is a technology that enables an individual to be recognized based on their unique biological traits captured from a facial image [Ve21]. Given the strong generalization capabilities of such systems, an adversary can execute targeted attacks against FRS that use morphed face images, as presented by Ferrara et al. [FFM14]. Face morphing is the smooth transformation of two facial pictures into one. Morphing attacks have developed as a serious threat to the enrollment process in recent years, successfully undermining the capabilities of facial recognition systems. As a result, this attack violates the rule of exclusive ownership [Ve21]. Morphing attack detection (MAD) algorithms have succeeded in determining alterations using deep learning in recent years. The MAD algorithms are doing very well, but what affects the algorithm's decision can be somewhat concealed from the user.

With the advancement of deep learning algorithms, biometric-based identification and verification have become a commonly utilized methodology for a variety of secure access control applications [Ve21]. Classification of images has become a critical component of a

[1] Norwegian University of Science and Technology, Department of Information Security and Communication Technology, NTNU Gjøvik Teknologiveien 22, 2815 Gjøvik, Norway hennimy@stud.ntnu.no

wide variety of computer vision applications, with nonlinear methods such as convolutional neural networks (CNNs) serving as the gold standard [La16]. This is due to deep neural networks, and recognition systems generally have an excellent capacity for generalization in their decision-making. Automated facial recognition is a long-standing area of research that has seen a significant advancement with the introduction of deep neural networks [Sc19]. While approaches based on learned features, such as neural networks, can reach a high level of accuracy, they act like black boxes [SHE21a]. As neural networks become more widely used, the topic of how these models' conclusions may be interpreted becomes increasingly important [Ra21]. While precision is necessary for network performance, generality and robustness are equally critical. One aspect of neural networks is that they frequently employ only the data necessary to perform their task and reject additional helpful information. Worst case, a neural network learns to make correct decisions for the wrong reasons [Se20]. Later in this paper, we will see how our model primarily uses a small area around the person's eyes, nose, and mouth to determine if the image is a bona fide or morph.

In the adversarial attack scenario, we suppose that the attacker has access to the neural network like a black box and is free to test it as frequently as they please. Recently, multiple attacks on neural network prediction were published using only minor content perturbations and without knowledge of the network's weights or design [Se18]. Studies demonstrate that even without knowledge of the network's design or weights, the attacker can influence the network's judgment with only minor changes to the content. This poses a threat to neural networks in a variety of applications, particularly those involving security, where the network must be precise and resistant to specific attacks on its decision-making process [Se18].

We use the VGG19 network architecture to detect morphed images in this paper. The VGG19 can be downloaded pre-trained, but additional fine-tuning of the network is required for our purpose. During this re-training phase, the VGG19 network must predict whether specific portions of an image contain indications of morphing or not [Se20]. This model will only focus on reference-free single image-based MAD (S-MAD). The newly founded discipline of explainable artificial intelligence has seen the development of a plethora of methodologies. Layer-wise relevance propagation has established itself as a notable method for enhancing the interpretability of DNNs and is used in this paper. This explanatory approach generally examines the model's interpretability from a black-box perspective. It provides instance-specific explanations that quantify the significance or relevance of each feature in a single input in relation to the model's output [XD20].

The following are the contributions and research questions to be explored further in this paper:

1. Explore previous research on morph attack detection and explainable visualization techniques for neural networks.

2. Can visualization techniques be applied to improve the explainability of S-MAD algorithms?
3. By applying visualization of the S-MAD algorithms, which are the most important features of the input images for different classification results?

2 Background

We will explore what face morphing is and how it is performed during this background segment, as well as what consequences it could have and how to detect different morphing attacks. Finally, we will look at convolutional neural networks and what visualization techniques exist to help with the explainability of MAD neural networks.

2.1 Face Morphing

Since the 1980s, picture morphing has been a focus of image processing research, with a wide variety of application situations, most notably in the film industry [Sc19]. Morphing is a term that refers to a particular effect that converts one image into another [Ve21]. This technique can be used to construct artificial biometric samples that mimic the biometric information of two, or more individuals, as seen in Figure 1. The best results are obtained when frontal photos with a neutral facial expression are used [Sc19]. Such face morphing attacks have implications on the integrity of automatic and manual identity verification procedures, like those conducted at country borders [SHE21a].

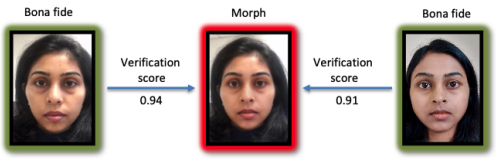

Fig. 1: Face morph illustrated in the middle getting a high similarity score against two bona fide samples from different individuals. Illustration is gathered from this paper [Ve21]

Given that numerous countries, including the USA, France, and Germany, enable residents to provide a picture for passports or national identification cards, a face morphing attack requires no further attacks on other IT infrastructure to execute [SHE21a]. The majority of state-of-the-art morphing algorithms do not treat the image as a grid but instead perform a Delaunay triangulation on the landmarks to determine non-overlapping triangles [Sc19]. Due to the fact that the morphing process alters the pixel positions, some mismatched pixels may result in noise-generating artifacts and ghost-like pictures, giving the photos an unreal aspect [Ve21]. After creating the morphed face image, it can be further processed and manipulated to remove or minimize these unnatural aspects. The image quality may be intentionally increased or decreased to disguise the picture modification. Automatically

created morphs may introduce artifacts, which can be avoided if the attacker creates a single high-quality morph between himself and his partner and manually optimizes the final image [Sc19].

Following the morphing process, some portions of the original facial image, eyes, nostrils, and hair are blended over the morph to conceal artifacts [Sc20]. In general, it is anticipated that mechanically created databases of morphed face photos will have a lower quality than real-world attack scenarios [Sc19].

2.2 Morphing Attack Detection

Noting the limits of human observers, several recent proposals for automatic Morphing Attack Detection (MAD) have been made [Ve21]. The MAD algorithms proposed thus far have been trained and evaluated on datasets with constrained distributions of image features, either with a single morphing tool or with somewhat unrealistic distributions due to the absence of print-scan transformation [Sc20]. For example, the recent NIST FRVT MORPH results indicate that most MAD algorithms submitted lack resilience and performance when applied to unknown and demanding datasets. NIST also evaluates MAD algorithms for various quality levels of morphed face photos in the FRVT MORPH [Sc20].

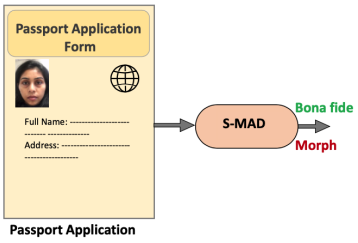

Fig. 2: Illustrating the single image based morph attack detection in passport application scenario [Ve21]

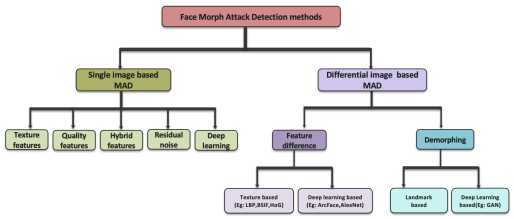

Fig. 3: How different morph attack detection techniques relate to each other [Ve21]

The MAD techniques available are categorized into two broad categories [Ve21]: Single image-based MAD (S-MAD) and Differential image-based MAD (D-MAD).

When trusted live capture from an authentication attempt is obtained in a trustworthy environment, differential-MAD determines if the image is a morph or bona fide by comparing the two [Ve21]. This is outside the scope of this paper, which will focus exclusively on S-MAD, illustrated in Figure 2. The majority of past research has been conducted under this no-reference condition for morphing attack detection [Sc19]. An overview of the different morph attack detection techniques is shown in Figure 3 [Ve21].

Single Image Based MAD is to detect a face morphing attack effectively using a single image supplied to the algorithm. The morphing image might be digital or re-digitized, also

known as print-scan [Ve21]. S-MAD is a difficult task since it is supposed to be robust against differences in picture quality, multiple types of sensors, different types of morph creation tools, and various print-scan procedures [Ve21].

Based on the features used, the existing S-MAD approaches can be further categorized into five sub-types [Ve21]. This paper will mainly discuss deep learning based S-MADs. The other categories listed below will not be discussed in detail [Ve21].

1. Texture features based S-MAD
2. Quality based S-MAD
3. Residual noise-based S-MAD
4. Deep learning-based S-MAD
5. Hybrid approaches for S-MAD

Researchers have successfully used deep learning S-MAD algorithms to categorize bona fide and morphed images. For the most part, deep Convolutional Neural Networks (CNN) are used. To train a CNN from the ground up, a sizable database is required. Therefore, most previously published work utilizes pre-trained networks and transfer learning [Ve21]. Although deep CNNs outperform hand-crafted texture descriptor-based MAD algorithms on both digital and print-scan data, their generalizability and robustness across diverse print-scan datasets are restricted. There is no need to train CNN from scratch since pre-trained CNNs already exhibit high detection capability. As the very nature of deep neural networks is computationally expensive, this approach utilizing CNN has drawbacks such as a high processing cost [Ve21].

2.3 Explainability of deep learning models

Since deep learning is doing an excellent job in detecting morphing attacks, it is helpful to be able to explain what the algorithm uses in its decision-making. As seen with deep learning in MAD algorithms, it is essential to improve the explainability, and visualization techniques are a common approach. First, a new neural network is trained from scratch, or an existing neural network is re-trained for the MAD task. Theoretically, deep neural networks can be trained to identify any artifact [Sc20]. Most approaches for face morph recognition are trained and evaluated on a single database utilizing a single morph generation algorithm [Sc19]. As a result, the training data must have a high degree of variance to avoid overfitting and database-specific artifacts. Deep face recognition networks have demonstrated a high degree of robustness, even when faced with difficult data [Sc20].

2.3.1 Convolutional neural network

A neural network is made up of node layers, each of which has an input layer, one or multiple hidden layers, and an output layer. Each artificial neuron is connected to the others and has a weight and threshold. If an individual node's output exceeds a specified threshold value, that node is activated, transmitting data to the network's next tier. Otherwise, no data is sent further [Ne20a].

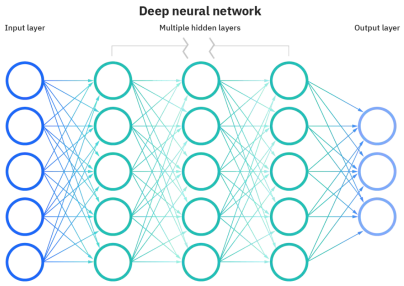

Fig. 4: High-level illustration of a deep neural network [Ne20a]

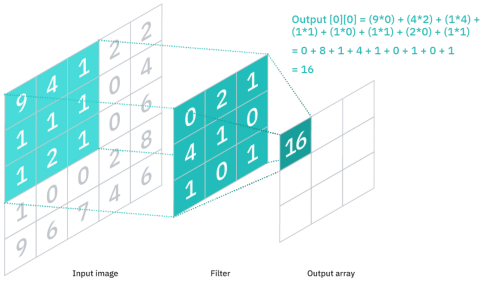

Fig. 5: Showing how the convolution layer could be calculated in a CNN [Ne20b]

A neural network is considered a deep neural network when it has three or more hidden layers, shown in Figure 4. DNN is an umbrella term for multiple types of neural networks, including convolutional neural networks, CNN [Ne20a].

Convolutional neural networks differ from other neural networks in that they function better with picture inputs. They have three distinct layers: convolutional, pooling, and fully connected. Figure 5 is a representation of the convolutional layer and how it can be calculated [Ne20b]. Figure 6 illustrates how all these distinct functions are connected.

The convolutional layer is the fundamental part of a CNN, as it is responsible for the majority of computation. This layer is composed of three components: input data, a filter, and a feature map. Pooling layers, also known as downsampling, is a technique for lowering the dimensionality of an input by reducing the number of factors. Like the convolutional layer, the pooling operation sweeps an unweighted filter across the entire input. Pooling can be classified into two types: maximum pooling and average pooling. Each node in the output layer is connected directly to a node in the preceding layer in the fully-connected layer. This layer performs classification based on the features gathered in the preceding levels. Typically, fully connected layers employ a softmax activation function to classify inputs appropriately, generating a probability between 0 and 1 [Ne20b].

The Visual Geometry Group developed VGG-19, a convolutional neural network architecture. It contains 19 layers with trainable weights, hence the name. VGG-19 employs an architecture with minimal, 3×3 convolution filters. This is the architecture we will use together with LRP in this paper and is illustrated in Figure 7 [Mi21].

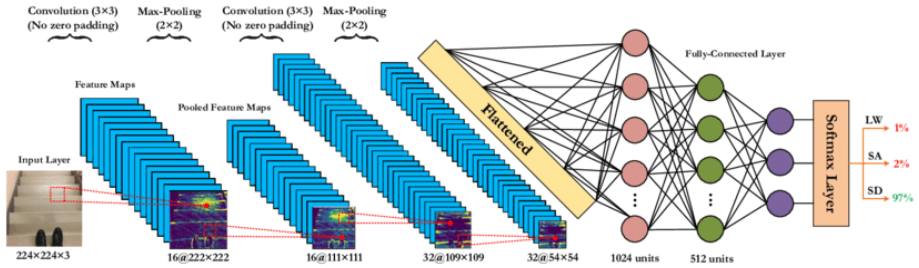

Fig. 6: Showing all the parts of a CNN and how they work together [KS19]

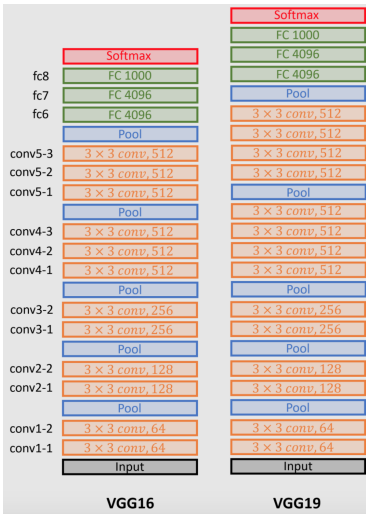

Fig. 7: Showing the VGG16 and VGG19 architectures and how they differ [da18]

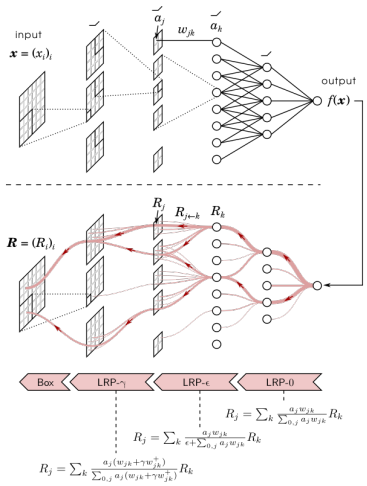

Fig. 8: Showing how LRP is designed visually [he]

Transfer learning can be used to fine-tune a pre-trained deep facial recognition network to detect morphs. However, due to the model's significant complexity, a large amount of training data is still required [Sc20]. In the instance of CNNs used to detect face morphing attacks, the CNNs are constantly exposed to face images, and the traces of the morphing stages can be rather subtle and appear in many regions of the face [SHE21a]. A network used to detect forgeries should be particularly resistant to all types of attacks on its decision-making [Se20]. In terms of CNNs for detecting face morphing attacks, we are interested in detecting and highlighting morphed traces [SHE21a]. Deep neural networks can be tricked by subtle modifications of visual content that are invisible to the naked eye [Se20]. It is therefore interesting to get insight into their decision-making process.

2.3.2 Layer-wise Relevance Propagation

The Layer-wise Relevance Propagation (LRP) interpretability approach assigns significance to each pixel in the input image and is illustrated in Figure 8 [SHE21a]. The LRP assigns relevance layer by layer, starting with a single selected neuron representing a single class and ending with the image via the CNN. Each layer's relevance is communicated backward into the preceding one by a set of rules. These criteria are intended to direct attention to the neurons in the prior layer that are required for each neuron in the current layer to fire [SHE21b].

We employ LRP to gain insight into the decision-making process to interpret the MAD algorithm's accuracy, and robustness [Se20]. There is a risk of overfitting with using low-variance datasets. Particularly, the resulting deep classifiers may prefer image areas with artifacts, such as shadows surrounding the iris region [Sc19]. To guarantee that the layer's relevance is transferred into the input image, LRP uniformly distributes relevance over all pixels covered by the size of the convolutional filter [SHE21b]. LRP considers the CNN's overall structure, the classification component, as well as the convolutional layer activations and weights. The relevance is assigned so that regions that considerably contribute to the activation are given a positive value. In contrast, areas that significantly inhibit its activation are assigned a negative value. This enables the production of finer heatmaps and the assignment of a relevance score to each pixel, defining its ability to either contribute to or prevent activation. This heatmap indicates which image regions are critical to the network's decision [SHE21a].

LRP's mathematical foundation is built on a deep Taylor decomposition of a neural network for a particular class. We feed the desired image into the CNN and establish relevance using the resulting activation levels. This initiation technique is intended to prioritize neurons that identify face morphing artifacts. To propagate the significance into the input picture, we begin with this assignment of relevance in the final layer of the feature extractor.

It is possible to utilize LRP with different rules for the relevance back-propagation. We follow the rules currently regarded as best practice for LRP [SHE21b]. These are epsilon-decomposition for the fully connected layers. Illustrated with green color in Figure 7. Alpha-beta - decomposition with a = 2, b = - 1 for the convolutional layers in the blocks 3 to 5. Flat decomposition for convolutions layer blocks 1 to 2, illustrated with an orange color in Figure 7 [SHE21b].

3 Method

This paper is researched by first taking a qualitative approach to the issue of explainability of deep learning networks. A literature study was done to establish a foundation. Academic references were identified using reputable online academic search engines and literary databases. The research approach reviewed several sources, and the collected literature is

primarily from peer-reviewed papers or international standards. After establishing a good foundation on the topic, this paper adds quantitative research by fine-tuning a pre-trained neural network for morph attack detection. In addition, we add LRP functionality for visualizing what input contributed to the network's final decision and look at performance metrics.

The dataset used in this paper to fine-tune the MAD algorithm is a subset of the database presented by Phillips et al. [Zh21]. The data originates from the FRGC-v2 dataset, and morphs were generated by the 'Landmark-I' algorithm mentioned in this paper [Zh21]. The 'Landmark-I' algorithm employs three different face morph generation techniques based on facial landmarks constrained by Delaunay triangulation with blending. The dataset consists of 140 unique participants from the FRGC-v2 collection based on the high-quality facial photos where images similar to passport pictures were chosen. 47 of the 140 data individuals are female, whereas 93 are male. Each subject has a sample size of 7 to 21 images. The images are cropped by utilizing MTCNN presented by Zhang et al. [Zh16].

The database explained above is split into multiple datasets. The MAD algorithm's dataset utilized during fine-tuning consists of 2045 bona fide images and 2499 morphed images. This is split into a training set and a validation set. The training dataset is made up of 694 bona fide and 1190 morphed images, 1884 in total. The validation dataset is made up of 1351 bona fide and 1309 morphed images, 2660 in total. The dataset utilized in the LRP explanation is split into a test set and a test_explain set. Each dataset contains 584 bona fide images and 1311 morphed images.

Due to the high computational cost of training the MAD network and calculating different performance metrics, we have had to compute two different MAD networks with a parameter change. This is due to the computer system used in this paper not being powerful enough to train both a high accuracy network and measure performance metrics at once. As a result, the tasks had to be split into:

1. Compute the MAD network with a high batch size in order to compute a higher number of epochs. This results in a higher accuracy network which LRP uses to create visual explanations of the network's decision-making.
   a) Train the deep learning network to detect morphing attacks by fine-tuning the VGG19 architecture. The batch size equals 32, and the number of epochs equals 24.
   b) Use the model with the best accuracy to visualize the most relevant features for the MAD decision-making process using LRP. The LRP batch size is set to 12.
2. Compute the MAD network with a batch size of 1 to compute detection performance metrics. Since this is much more computationally expensive, the used computer system cannot achieve a high number of epochs.
   a) Train the deep learning network to detect morphing attacks by fine-tuning the VGG19 architecture. Batch size equal to 1, number of epochs equal to 6.

b) Compute a confusion matrix based on true positive, true negative, false positive, and false negative metrics captured during the computing phase. Compute APCER and BPCER of the neural network.

The experiments in this paper are based on the code from Frederik Hvilshøj GitHub repository TorchLRP [Hv] as well as the beginner tutorial from pytorch.org authored by Nathan Inkawhich for fine-tuning torchvision models [In]. Unnecessary code has been removed, and some parameters have been changed concerning batch size, epoch number and heatmap settings. The code has also been modified to be able to compute a confusion matrix if the batch size is equal to 1. Ubuntu 20.04.4 LTS was used with Conda for the python environment. The Conda environment was created for reproducibility by utilizing the requirments.yml file from the TorchLRP repository. The MAD network is initialized using the weights of a pre-trained VGG19 network and then fine-tuned on the dataset described above. The code was set to fine-tune the whole model by setting the feature extract parameter to false.

After the fine-tuning is finished, we use LRP to determine the input relevance in the bona fide and morphed images for the network trained with a batch size of 32. By executing the explain.py code, we compute different kinds of explanatory photos. The code calculated a heatmap for alpha2beta1, epsilon, gamma+epsilon, patternnet, and patternattribution explanations. Both the epsilon and gamma explanations are fixed to 1e-1 [Hv]. The bona fide and morphed pictures presented in Figures 12 and 16 were chosen by setting the torch.manual_seed parameter equal to 1.

4 Experiments & Results

Since MAD performance can be visualized as a binary classification problem, the following metrics are widely used to benchmark the MAD algorithms. The performance of the detection algorithms is reported according to metrics defined in ISO/IEC 30107-3 [IS17]. The Attack Presentation Classification Error Rate (APCER) is defined as the proportion of attack samples incorrectly classified as bona fide images [Ve21]. The Bona fide Presentation Classification Error Rate (BPCER) is defined as the proportion of bona fide images incorrectly classified as a morphed image in the system [TB21].

4.1 Results from the MAD network used together with LRP

The neural network training that yielded the best results got a training accuracy of 0.9894 and a validation accuracy of 0.9218. This is from the network trained with a batch size of 32 and is considerably higher than the network trained with a batch size of 1. Figure 9 illustrates the relationship between the training and validation accuracy for all 24 epochs. The training accuracy is consistently high, right below or at 1, while the validation accuracy

ranges between 0.55 and 0.92. Based on the results of this fine-tuned MAD algorithm, we used LRP to visualize what the neural network had used in its decision-making process. The bona fide images are presented first in Figures 12 to 15, and the morphed images are presented in Figures 16 to 19.

4.2 Results from the MAD network used to calculate APCER and BPCER

The training of this neural network with a batch size equal to 1 yielded lower results than the network above. This is reflected in Table 1, which contains the network's performance metrics. The best result for this network was a validation accuracy of 0.55. Figures 10 to 11 show the relationship between APCER, BPCER, and accuracy for the training and validation phases. Setting the batch size to 1 resulted in substantially lower performance.

Tab. 1: Overview of APCER and BPCER and the values used to calculate them

Epoch	Phase	TP	TN	FP	FN	APCER	BPCER	Accuracy
1	Train	513	1064	180	125	0.1447	0.1959	0.8379
1	Validation	1260	62	91	1246	0.5948	0.4972	0.4972
2	Train	447	984	216	205	0.1800	0.3144	0.7727
2	Validation	773	694	578	614	0.4544	0.4427	0.5517
3	Train	216	903	447	286	0.3456	0.5697	0.5946
3	Validation	34	1306	1317	2	0.5020	0.0556	0.5039
4	Train	320	959	373	230	0.2800	0.4182	0.6796
4	Validation	1033	307	318	1001	0.5088	0.4921	0.5039
5	Train	454	1040	239	149	0.1869	0.2471	0.7938
5	Validation	398	1034	953	274	0.4796	0.4077	0.5385
6	Train	483	1044	210	145	0.1675	0.2309	0.8114
6	Validation	627	634	724	674	0.5331	0.5181	0.4742

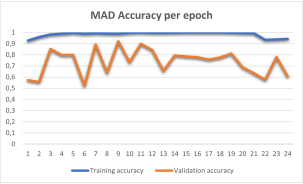

Fig. 9: Training & Validation accuracy, batch size = 32, 24 epochs.

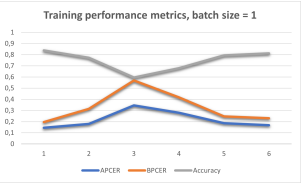

Fig. 10: APCER, BPCER & Accuracy for training, batch size = 1, 6 epochs.

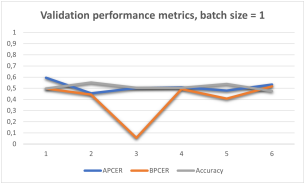

Fig. 11: APCER, BPCER & Accuracy for validation, batch size = 1, 6 epochs.

LRP get a small sub-sample of twelve bona fide and twelve morphed pictures from the overall dataset based on the batch size set in explain.py. Using LRP, we see what went into the decision of the MAD algorithm with the highest accuracy when classifying the images as bona fide or morphs. The red color in the heatmaps represents input pixels that positively affected the decision, while the blue color suppressed the decision.

Fig. 12: Bona fide input images

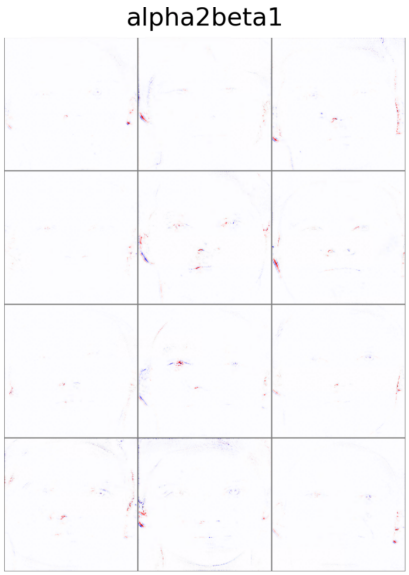

Fig. 13: Visualisation using the bona fide alpha2beta1 explanation

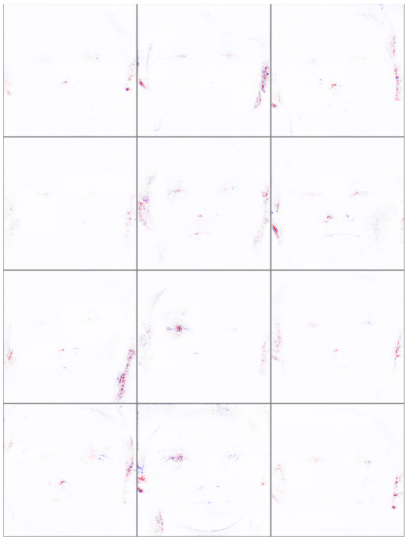

Fig. 14: Visualisation using the gamma + epsilon explanation on bona fide images

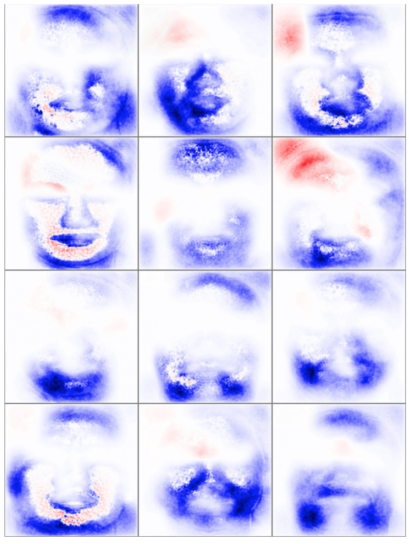

Fig. 15: Visualisation using the patternnet explanation on bona fide images

Fig. 16: Morph input images

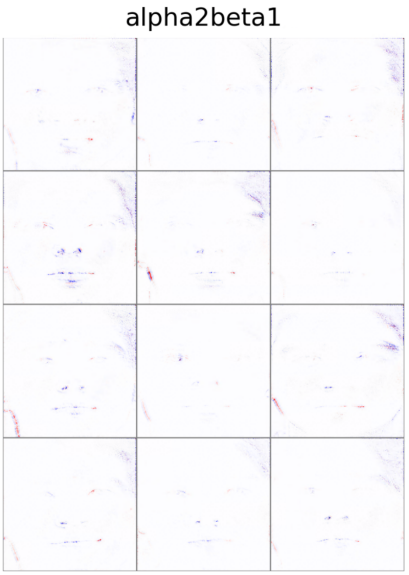

Fig. 17: Visualisation using the alpha2beta1 explanation on morphed images

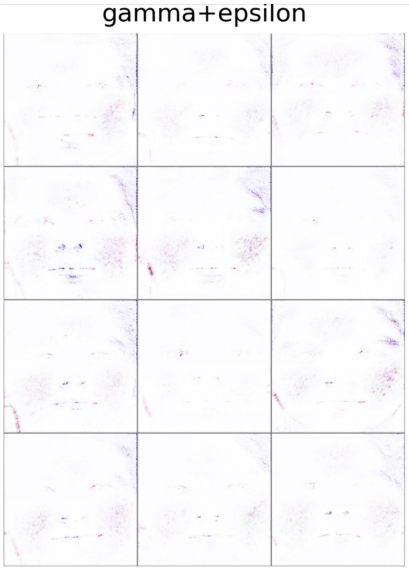

Fig. 18: Visualisation using the gamma + epsilon explanation on morphed images

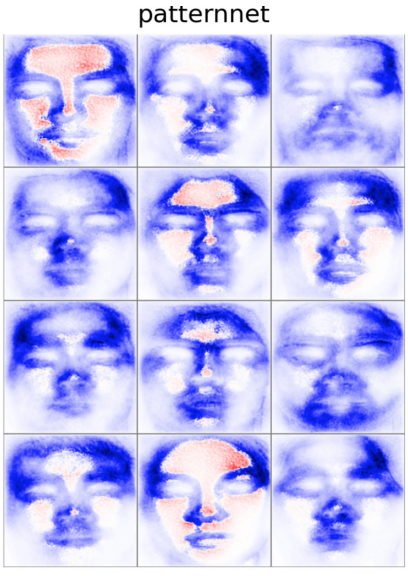

Fig. 19: Visualisation using the patternnet explanation on morphed images

The visual results show that the neural network primarily focuses on the eyes, nose, and mouth. This is best illustrated in the alpha2beta1 explanations. The algorithm also takes into account some of the hair features, as well as the edges of the faces. When examining the epsilon and gamma explanation, the algorithm considers more information about the rest of the face. In these illustrations, the ears and the edges of the face are more visible. The patternnet explanation shows a pattern where most of the image negatively influences its decision, while some areas in the forehead and cheek positively influence its decision.

5 Discussion

It should be mentioned that reliable detection of face morphing attacks continues to be a challenge, and numerous open issues exist in the research field of MAD algorithms [Sc19]. One of these issues is the absence of large-scale publicly available datasets with more individual variation as well as a technological variation to reflect the real world [Ve21]. In addition, generating high-quality face morphs automatically continues to be a difficult task. The dataset used in this paper arguably has more artifacts and ghosting in the morphed images than an attacker would achieve manually for a specific purpose. Training neural networks demand considerable amounts of computer resources, making it expensive to train MAD algorithms for professional use cases.

From Table 1, the validation accuracy is relatively low during all epochs. This results in very high and inconsistent APCER and BPCER values. The calculations indicate that the network cannot be used for any meaningful classification of bona fide or morphed images without more optimization and training. This poor performance may be a result of using a small batch size. There is a clear difference in performance between Figure 9, with 32 in batch size, vs. Figure 10 and 11, with a batch size of 1.

We discovered using LRP that our fine-tuned neural network, with the highest accuracy, concentrates on the eyes, nose, and mouth areas while detecting morphed facial images. For the most part, the rest of the image is ignored. While focusing on these features may be adequate to achieve relatively high accuracy, it has limitations. There is a high degree of inconsistency in what the algorithm deems relevant between the different ways of visualizing the input relevance. This could have unforeseen problems, which is especially serious for security-related applications. In critical systems, the algorithm should consider data from all image locations during the classification process to achieve robustness.

While examining the alpha2beta1 visualization technique, the algorithm consistently put relevance to the eye region, while relevance to the nose and mouth is more inconsistent between the images. This aligns with findings from other researchers that also found that in the majority of circumstances, the eyes may be adequate to discern between genuine and morphed facial photos [Se20].

Due to the complexity of the behavior of a CNN's fully connected layers, the relevance scores generated by LRP are not immediately interpretable, demanding additional research

to comprehend the network's overall behavior completely. Seibold et al. [SHE21b] found that LRP commonly assigns high relevance scores to artifact-free regions in morphed face shots, implying that these regions are critical for the decision to classify the images as morphs. This aligns with our results where we see the neural network assign relevance to areas without any visible artifacts and fails to detect other areas with clearly visible artifacts. In Figures 16 and 17, this is especially visible where the hairline of multiple morphed images has artifacts that the algorithm does not detect very well.

A weakness of this paper concerns the fact that due to limited computational resources, we had to train the network with two different methods. Method one, with a large batch size, in order to achieve as high validation accuracy as possible. This was to have the best neural network foundation to use with LRP. Method two with batch size equal to one in order to calculate true positive, true negative, false positive, and false negative. The computation was much more expensive with smaller batch size, resulting in lower accuracy. This results in two different neural networks with different performances, making the calculated performance metrics less relevant than if we only had one MAD network. To justify this weakness, it would be relatively easy to train only one network with more computational resources available. The code is written and may only need some optimizations. It should also be mentioned that the results indicate that larger batch size not only is faster to compute, but also results in higher accuracy.

Using LRP to visualize the decision-making process of convolutional neural networks fine-tuned for morph attack detection has been shown to be inconsistent. Problems with limited high-quality large datasets are an issue and result in poorly trained algorithms. This makes the results of LRP challenging to assess and necessitates further research.

6 Conclusion

Given the strong generalization capabilities of face recognition systems, an adversary can execute targeted attacks that use morphed face images, as presented by Ferrara et al. [FFM14]. Face morphing attacks are a significant and dangerous concern, given that several countries enable residents to provide a picture for passports or national identification cards. Without requiring specialist knowledge, these photos can be forged utilizing readily available tools or websites [Se20]. Advances in deep learning and machine learning approaches have enabled the development of relatively good-quality morphs through various novel techniques. Generalizing morph attack detection is still a long way off, given the fundamental difficulty of gathering massive public databases with a variety of morph production strategies [Ve21]. Robust MAD algorithms must account for the vast diversity of picture post-processing, printing, and scanning technologies. The observed accuracy for detecting face image morphing attacks does not yet represent generalization to datasets containing a range of real-world capture situations [Sc19].

We discovered through LRP that a fine-tuned neural network focuses mainly on the eyes, nose, and mouth to detect morphed images. Though neural network analysis is still in its infancy, we demonstrated how methods such as LRP may be utilized to get valuable insights into a neural network's decision-making process. Future strategies for modifying the training process, particularly the training data, to increase resilience can be developed from this knowledge. Through the experiments, we got insights into how to train a network specifically for detecting face morphs and, more broadly, visual results on what the neural network used in its decision-making. High accuracy does not always imply robustness, implying the necessity for additional quality measures for convolutional neural networks. We presented high training and validation accuracy metrics for our MAD algorithm with a large batch size. LRP showed that most of the input image got ignored, implying a lack of robustness. In addition, our results show inconsistency in the algorithm's ability to detect artifacts and other features of morphed images. Future work could improve the used codebase and try to decrease the computational cost of training the neural network while still being able to calculate essential performance metrics for the MAD algorithm.

References

[da18] 013 CNN VGG 16 and VGG 19, Available from: https://datahacker.rs/deep-learning-vgg-16-vs-vgg-19/, Accessed: 12.04.2022.

[FFM14] Ferrara, Matteo; Franco, Annalisa; Maltoni, Davide: The magic passport. In: IEEE International Joint Conference on Biometrics. p. pp. 4, 2014.

[he] How and Why LRP ?, Available from: http://heatmapping.org/, Accessed: 14.04.2022.

[Hv] Implementation of LRP for pytorch, Available from: https://github.com/fhvilshoj/TorchLRP, Accessed: 14.04.2022.

[In] FINETUNING TORCHVISION MODELS, Available from: https://pytorch.org/tutorials/beginner/finetuning_torchvision_models_tutorial.html, Accessed: 14.04.2022.

[IS17] ISO/IEC DIS 30107-3, Information technology - Biometric presentation attack detection - Part 3: Testing and reporting.

[KS19] Khademi, Gholamreza; Simon, Dan: Convolutional Neural Network for Environmentally Aware Locomotion Mode Recognition of Lower-Limb Amputees. p. pp. 5, June 2019.

[La16] Lapuschkin, Sebastian; Binder, Alexander; Montavon, Grégoire; Müller, Klaus-Robert; Samek, Wojciech: The LRP toolbox for artificial neural networks. The Journal of Machine Learning Research, 17(1):pp. 1–4, 2016.

[Mi21] Understanding the Amazon Rainforest with Multi-Label Classification + VGG-19, Inceptionv3, AlexNet Transfer Learning, Available from: https://towardsdatascience.com/understanding-the-amazon-rainforest-with-multi-label-classification-vgg-19-inceptionv3-5084544fb655, Accessed: 09.04.2022.

[Ne20a] Neural Networks, IBM Cloud Education, Available from: https://www.ibm.com/cloud/learn/neural-networks Accessed: 03.04.2022.

[Ne20b] Neural Networks, IBM Cloud Education, Available from: https://www.ibm.com/cloud/learn/ convolutional-neural-networks, Accessed: 01.04.2022.

[Ra21] Raulf, Arne Peter; Däubener, Sina; Hack, Ben Luis; Mosig, Axel; Fischer, Asja: SmoothLRP: Smoothing LRP by Averaging over Stochastic Input Variations. pp. pp. 599–603, October 2021.

[Sc19] Scherhag, Ulrich; Rathgeb, Christian; Merkle, Johannes; Breithaupt, Ralph; Busch, Christoph: Face Recognition Systems Under Morphing Attacks: A Survey. IEEE Access, 7:pp. 23012–23014, 23016, 23019–23026, 2019.

[Sc20] Scherhag, Ulrich; Rathgeb, Christian; Merkle, Johannes; Busch, Christoph: Deep Face Representations for Differential Morphing Attack Detection. IEEE Transactions on Information Forensics and Security, 15:pp. 3625–3637, 2020.

[Se18] Seibold, Clemens; Samek, Wojciech; Hilsmann, Anna; Eisert, Peter: Accurate and Robust Neural Networks for Security Related Applications Exampled by Face Morphing Attacks. CoRR, abs/1806.04265:pp. 1–3, 6–10, 13, 2018.

[Se20] Seibold, Clemens; Samek, Wojciech; Hilsmann, Anna; Eisert, Peter: Accurate and robust neural networks for face morphing attack detection. Journal of Information Security and Applications, 53:pp. 1–6, 8, 10–11, 2020.

[SHE21a] Seibold, Clemens; Hilsmann, Anna; Eisert, Peter: Feature Focus: Towards Explainable and Transparent Deep Face Morphing Attack Detectors. Computers, 10(9):pp. 1–7, 9, 12, 14, 2021.

[SHE21b] Seibold, Clemens; Hilsmann, Anna; Eisert, Peter: Focused LRP: Explainable AI for Face Morphing Attack Detection. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV) Workshops. pp. pp. 88–92, 95, January 2021.

[TB21] Tapia, Juan E.; Busch, Christoph: Single Morphing Attack Detection Using Feature Selection and Visualization Based on Mutual Information. IEEE Access, 9:pp. 1–2, 5–6, 10, 2021.

[Ve21] Venkatesh, Sushma; Ramachandra, Raghavendra; Raja, Kiran; Busch, Christoph: Face Morphing Attack Generation and Detection: A Comprehensive Survey. IEEE Transactions on Technology and Society, 2(3):pp. 128–145, 2021.

[XD20] Xu, Jincheng; Du, Qingfeng: Adversarial attacks on text classification models using layer-wise relevance propagation. International Journal of Intelligent Systems, 35(9):pp. 1397–1399, 2020.

[Zh16] Zhang, Kaipeng; Zhang, Zhanpeng; Li, Zhifeng; Qiao, Yu: Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks. IEEE Signal Processing Letters, 23(10):1499–1503, 2016.

[Zh21] Zhang, Haoyu; Venkatesh, Sushma; Ramachandra, Raghavendra; Raja, Kiran; Damer, Naser; Busch, Christoph: MIPGAN—Generating Strong and High Quality Morphing Attacks Using Identity Prior Driven GAN. IEEE Transactions on Biometrics, Behavior, and Identity Science, 3(3):365–383, 2021.