Differentiating between Facial Landmark Shifts due to Pose Variation and Morphing

Lian van Kesteren
University of Twente
P.O. Box 217, 7500AE Enschede
The Netherlands
l.m.f.vankesteren@student.utwente.nl

ABSTRACT

Face recognition systems (FRSs) provide accurate biometric authentication, although some security risks have been discovered. In most cases, the FRSs are vulnerable to morphing attacks. In a morphing attack, the pictures of two subjects are morphed into one, resulting in an artificial image of a face that resembles both subjects. Multiple techniques have been developed to detect morphing attacks, however none of the existing algorithms have the necessary accuracy to prevent them. One technique consists of comparing the facial landmarks of the possibly morphed image with the bona fide image. Shifting of these landmarks could indicate the image is morphed. However, the shifts can also be caused by pose variation. This research aims to improve the performance of the aforementioned technique by differentiating landmark shifts due to pose variation and due to morphing to a greater extent. By including pose variation in the data performance can be increased significantly.

Keywords

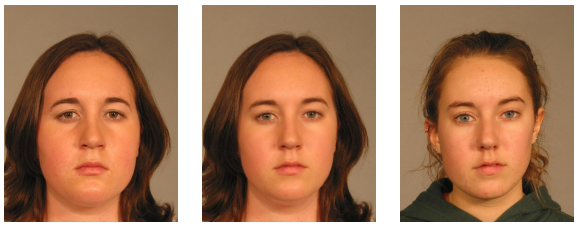
Face recognition systems, Differential morphing attack detection, Facial landmarks, Pose variation

1. INTRODUCTION

Face recognition systems (FRSs) are widely used for security purposes. It is considered an accurate form of biometric authentication. The software detects a face, normalizes the picture and extracts the facial features. If the facial features of two faces are similar enough, they are considered to be the same person. A threshold is necessary to allow for slight differences, as the facial features in two pictures of the same person are not always identical.

However, it was found that this threshold in combination with the applicant providing the picture allows for morphing attacks [3, 4]. In a morphing attack, the pictures of two different subjects are used to create a morphed image of the two, resulting in an artificial image of a face with the combination of the facial features of both subjects. An example of a morphed image created using the FRGC database [14] is seen in figure 1.

The morphed image now resembles both subjects. When an accomplice applies for an identity document using a morphed image, a criminal can use the identity document resulting in the criminal masking their identity with the identity of the accomplice. When the morphed image is of high enough quality, it can mislead FRSs and human experts [10, 18].



Subject 1 *Morph* *Subject 2*

Figure 1. Example of morphed image sliced into background of subject 1.

There exist several techniques for morphing attack detection (MAD). One method is to compare the facial features of the image that is possibly morphed with the facial features of an image of the same subject that is bona fide. The facial landmarks of both images are extracted and the positions are compared. An example of facial landmarks can be seen in figure 2. When the landmarks have shifted, the image could be morphed [1, 19]. Since pose variation and expressions also cause the landmarks to shift, it becomes difficult to detect if an image is morphed.

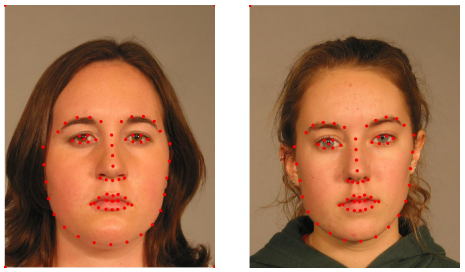


Figure 2. Examples of facial landmarks extracted with the library dlib [9].

The problem is that when the existing different morphing attack detection algorithms were tested on the same database with realistically challenging images, it was found that none of the algorithms came close to the accuracy that is needed [16]. More accurate algorithms need to be developed to successfully prevent morphing attacks.

36th Twente Student Conference on IT February 4th, 2022, Enschede, The Netherlands.

The goal of this research is to improve the performance of detecting morphing attacks using facial landmarks by including pose variation in the training process. We expect that we can increase the performance of this method by better differentiating between shifts of landmarks due to morphing and due to pose variation.

To pursue our goal, the following research questions (RQ) have been defined as the basis of our research:

- **RQ1:** To what extent can landmark shifts due to pose variation be differentiated from landmark shifts due to morphing when using machine learning algorithms?
- **RQ2:** How much can the performance of differential MAD based on facial landmarks be increased by differentiating between landmark shifts due to pose variation and due to morphing?

By the end of this research, we expect to have contributed by improving the performance of differential MAD based on facial landmark shifts.

The remainder of this paper is organized as follows. In § 2 we will discuss the related works. After that, the methodology § 3 is presented that was used to answer the research questions. In section § 4 all aspects of the data that was used are discussed. Following that in § 5 we will mention some specific details of the experiments. Then in § 6 will be the results that were found and a discussion. Finally in § 7, we end with a conclusion.

2. RELATED WORK

Several different techniques have been developed for morphing attack detection (MAD). They can be classified in two different scenarios, Single Image-Based MAD (S-MAD) and Differential Image-Based MAD (D-MAD).

When a morphing attack is to be detected based on a single image, it is considered S-MAD. The algorithms are based on the properties of the image itself. S-MAD algorithms need to be very robust as the quality of the image varies. Especially in realistic scenarios where the images are often printed and scanned, for example when creating an identity document. This lowers the quality of the image making it more difficult to detect a possible morphing attack.

One approach is to use micro-texture features that are extracted from the colour space to detect morphed images [15]. Another approach by Venkatesh et al. [23] is to use deep neural networks to denoise the image and then determine if the image is morphed or not based on the residual noise. Sherhag et al. [20] proposed an approach based on analyzing the spatial and spectral features extracted from Photo Response Non Uniformity patterns.

Other algorithms use deep learning to extract feature vectors from a suspected image and create a classifier. The learnt classifier is used to determine if the image is morphed. An algorithm proposed by Ferrara et al. [6] uses deep neural networks trained on a large generated print and scan images data set to detect morphed images. A different approach is to use a hybrid of characteristics like the algorithm proposed by Venkatesh et al. [22]. It uses the ensemble of features such as Local Binary Patterns, Histogram of gradients and Binarized Statistical Image Features to provide to a classifier that determines if the image is morphed or not.

The algorithm of Ramachandra et al. [17] is another algorithm based on a hybrid of characteristics. It detects morphed images by extracting the texture features from the scale-space representation of multiple colour spaces of the given image. The algorithm shows impressive results on the Face Recognition Vendor Test [11] for the high-quality morphs and printed and scanned data set, considering it is an S-MAD algorithm. However, for other data sets the performance was significantly reduced and similar to other S-MAD algorithms.

The evaluation of current S-MAD algorithms on multiple data sets concludes that the algorithms are not suitable for detecting morphing attacks in practice due to not meeting the required performance [11, 16]. For example, automated border control systems require a False Accept Rate (FAR) of 0.1% with a False Rejection Rate (FRR) not higher than 5% [7]. For the data sets in Raja et al. [16] none of the algorithms met these requirements and only reached less than 0.8 Bona fide Presentation Classification Match Rate (BPCER, false negative error rate) for an Attack Presentation Classification Match Rate (APCER, false positive error rate) of 0.1.

When the image that is suspected to be morphed is provided together with a bona fide image of the same subject, it is considered D-MAD. One approach is to demorph the possibly morphed image by reverting the methods used to create a morphed image using the bona fide image as a reference [5, 12, 13]. Another approach is to compare the facial features of the possibly morphed image with the bona fide image, as they will be slightly different when the image is morphed [1, 19].

Scherhag et al. [21] show a promising approach to detect whether an image is morphed based on deep face representations. The performance was less than 3% D-EER measured on the FRGCv2 database with morphed images generated using various automated morphing tools and no manual post-processing. The method proposed in Scherhag et al. uses features from a face recognition system that was trained following Deng et al. [2] using an Additive Angular Margin Loss function.

The evaluation of existing D-MAD algorithms on multiple data sets concludes that D-MAD algorithms perform better than S-MAD but still do not meet the required performance needed to prevent morphing attacks in practice [11, 16].

3. METHODOLOGIES

To reach our goal of improving the performance of detecting morphing attacks using facial landmarks by differentiating between landmark shifts due to pose variation and landmark shifts due to morphing, the research questions need to be answered. RQ1 will be split into two methods of differentiating landmark shifts due to pose variation and due to morphing. One method will be optimizing the feature vector set and algorithm selection, the other method will be including pose variation in the data set used for the machine learning algorithm. For each method we will evaluate the performance, thus answering RQ2.

3.1 Feature vectors and algorithm selection

Current morphing attack detection methods based on landmarks use supervised machine learning algorithms to classify if images are morphed. Feature vectors used as input for the supervised machine learning algorithm are based on the shifts of the 68 landmarks extracted using the python library dlib [9]. We expect that we can increase the performance by optimizing the feature vector selection used

by the supervised machine learning algorithm.

To increase the performance of the algorithm by optimizing the feature vectors selection, a base performance is needed. The performance of the algorithm with the landmark shifts as vectors, from all 68 landmarks extracted using dlib, as feature vectors will be the base performance. This performance can then be compared to the performance of the algorithm on the same data but with a different feature set. The altered feature vectors selection increases the performance if the performance is higher than the base performance.

One problem could arise here. As the data used for the machine learning algorithms is the same as the performance is compared, the feature vectors selection could be overfitted to the data. To counter this we will use cross validation to help prevent overfitting.

In order to select feature vectors, we need to analyze which landmarks and the characteristics of their shifts can be used to classify whether images are morphed or not. First we will hypothesize which landmarks will be most affected by the morphing process. Then we will analyze the shifts of these landmarks for morphs and bona fide images and alter the feature vectors used by the machine learning algorithm. Comparing the performance of the machine learning algorithm using this feature set to the performance using the basis feature set will confirm if the feature vectors selection optimized the performance.

As there are multiple supervised machine learning algorithms, we will also compare the performance of different machine learning algorithms. Support vector machine with different kernels and Random forest was chosen. The decision for these algorithms was because of the following characteristics of our situation. The amount of data is never smaller than the amount of feature vectors. Also, the feature vectors are not independent of one another. Furthermore we are aiming for a high performance and the interpretability of the results is not a priority.

After testing multiple feature vector sets for multiple algorithms we can conclude how much we can increase the performance by differentiating landmark shifts due to pose variation and due to morphing using a specific feature set and algorithm.

3.2 Pose variation

Increasing the performance of the algorithm by including more pose variation in the data is done by using the PUT database [8]. More information about the database and how the morphs were created is mentioned in the section §4.

To determine how much training the algorithm on a different database with more pose variation increases the performance of differential morphing attack detection algorithms based on facial landmarks, we need to evaluate the performance. As the performance of machine learning algorithms heavily relies on the data used to train and test the algorithm, the methodology needs to be carefully considered.

Databases of portraits are considered personal data and so data privacy rights apply to these databases. Also not many portrait databases exist and not all researchers within this topic have access to them. All methods of morphing attack detection currently published use different databases for evaluating the performance of the algorithms. Depending on the database and the quality of the morphs, the performance of the algorithm can differ considerably. This is a known issue when trying to solve the problem of morphing attack detection and is described in Raja et al. [16]. This paper made an effort to solve this problem by creating a large database of realistic bona fide and morphed images and evaluating the performance of existing implementations of different kinds of morphing attack detection algorithms on this database. This resulted in some algorithms having a considerably lower performance than previously measured. It would be the most accurate performance comparison if our proposed algorithm was also tested on this database. However, within the time and access constraints of this research it was not feasible.

To evaluate the performance of our proposed algorithm and to be able to possibly conclude that our proposed method increases the performance of MAD based on landmark shifts, a controlled setting is needed. In order to properly compare the performance of two algorithms, they need to be trained on the same database and also tested on the same database. We will train and test using the PUT database [8] to be able to train our proposed algorithm with any kind of pose variation and train an alternate algorithm with a limited amount of pose variation. By then testing on a separate set of images with any kind of pose variation we can deduce if training on a database with more pose variation affects the performance.

The alternate algorithm will be exactly the same as our algorithm, only differentiating in the training set that is used. By having two identical classifiers, we can properly measure how much the performance can be improved by using a different training set. A consequence of this setting is that we cannot properly compare the performance of our proposed algorithm to other existing MAD techniques. Still if an increase can be seen in the performance of our proposed algorithm, our goal of increasing the performance by better differentiating landmark shifts due to morphing and due to pose variation has been reached.

4. DATA

In this section we will discuss which images were used, how the morphs were created and which methods were used to normalise the images.

4.1 Database

For this research portrait pictures were needed that have minimal pose variation to the maximum amount of pose variation possible while still considered to be a frontal images. The PUT database [8] is ideal for this as it contained portrait images of 100 data subjects in different poses. The images that were considered frontal images were used for this research. The frontal images were hand selected using the criteria that all facial features have to be visible in the image.

4.2 Morphed images

Before the images can be morphed, they have to be selected first. In order to create high quality morphs, the two subjects are chosen based on a high similarity score. From these subjects two images with similar pose are selected to be morphed. Repeating this process results in multiple morphs with different poses for each subject pair.

The technique used to create the morphed images is based on the most common method used for research. First the landmarks are detected and then a Delaunay triangulation is applied for both images. These non overlapping triangles are used to warp the images by averaging until the triangles match. Then the images are combined by blending. To prevent possible ghosting outside the con-

tours of the face, the face is spliced into the background of the original picture of one of the subjects.

The landmark shifts of morphed images were calculated using a random bona fide image of the subject that the morph was sliced into that was not used for creating the morph. This way the morph landmark shifts are also affected by pose variation. We kept a 1:1 ratio of morph landmark shift data and bona fide landmark shift data when training and testing the algorithms.

4.3 Normalisation

For the images to be used, image processing has to be done first. It is important that the landmark shifts are as accurate as possible. Since the images contain pose variation we opted to use the centers of the eyes calculated using the eye corner landmarks for the image normalisation. The landmarks of the corners of the eyes are always present in the images as we will be only using frontal images. Another method of using the landmarks of the contour of the face was considered, however for the images with quite a bit of pose variation the contour of the face is not symmetric. This could possibly have a negative effect when normalizing the images.

Figure 3. Normalisation example on two stock images.

If an image is more closeup than the other, the landmark shifts will be bigger than they are supposed to be as the shifts are calculated in pixels. If one face contains more pixels it will distort the landmark shifts. To counter this the centers of the eyes are calculated using the landmarks of the corners of the eyes. Then the distance between the centers of the eyes in pixels is made equal for both images.

A head tilt in the picture or a picture taken at an angle can cause unnecessary shifts in landmarks which could make it more difficult for the machine learning algorithm to decide if a picture is morphed. This can easily be corrected by rotating the image slightly until the centers of the eyes are aligned horizontally. The centers of the eyes are calculated using the landmarks of the eye corners as mentioned before.

Furthermore the position of the face in the image is not always centered. This could also cause unnecessary landmark shifts. In order to avoid this black pixels can be added to the picture to ensure the centers of the eyes have the same pixel location in the images.

An example of this process can be seen in figure 3. After processing the images the landmark shifts calculated from the images are not cancelled out by unnecessary shifts due to different resolutions, the picture being taken at an angle or the face not being centered in the picture.

Normalising the images could possibly have an effect on the landmark shifts. However, if these methods of normalising the image are omitted, the shifts due to morphing will be much harder to detect. Especially since the shifts due to morphing are relatively small and can be easily cancelled out by shifts due to one of the pictures not being taken straight for example.

5. EXPERIMENT

In this section we will discuss in more detail certain aspects for the experiments and evaluations previously mentioned in the methodology § 3.

5.1 Metrics

To evaluate the performance of selecting specific input feature vectors and algorithms we will use the metrics seen below.

- The Bona fide Presentation Classification Error Rate (BPCER): the rate of bona fide images that are incorrectly classified as morphed images.
- The Attack Presentation Classification Error Rate (APCER): the rate of morphed images that are incorrectly classified as bona fide images.
- Accuracy: the rate of images classified correctly.

To evaluate the performance of including pose variation in the training set we will use the following metrics.

- The Bona fide Presentation Classification Error Rate (BPCER): the rate of bona fide images that are incorrectly classified as morphed images.
- The Attack Presentation Classification Error Rate (APCER(t)): the rate of morphed images that are incorrectly classified as bona fide images.
- The EER of the differential morph attack detection (D-EER): the error rate at the threshold t for which APCER(t) = BPCER(t).
- BPCER10: the lowest BPCER(t) under the condition that APCER(t) $\leq$ 10%.
- BPCER20: the lowest BPCER(t) under the condition that APCER(t) $\leq$ 5%.
- BPCER100: the lowest BPCER(t) under the condition that APCER(t) $\leq$ 1%.

5.2 Images

The following table shows how many images were used for testing multiple feature vector combinations and multiple algorithms.

	Morphs	Bona fides
Training	2165	2165
Testing	1443	1443

The table below shows how many images were used to train and evaluate the classifiers to test if pose variation in the training set can increase performance.

	Morphs	Bona fides
Training	498	498
Testing	332	332

5.3 Tools

Certain python packages were used to complete this research. We will only mention the packages used to calculate the results.

Dlib was used to extract the landmarks from the images. We used scikit learn for the classifiers. Furthermore the module metrics from scikit learn and matplotlib were used for plotting the results.

For the scikit learn classifiers only the default parameters were used, except for the Random Forest classifier for which we specified 500 trees.

6. RESULTS AND DISCUSSION

In this section the results that were found are shown and discussed for each method.

6.1 Feature vectors and algorithm selection

We tested multiple sets of feature vectors in order to find out if performance could be increased. 5-fold cross-validation was used to prevent overfitting the feature vectors to the data. We also tested multiple machine learning algorithms, specifically Support vector machine (SVM) with linear, polynomial and Radial basis function (RBF) kernels and Random Forest (RF) with 500 trees. As cross-validation was used, all measurements are average percentages with their average standard deviations.

The base performance was measured using all 68 landmark shifts by using the x and y coordinates of the shift vector as input feature vectors for the algorithms.

68 shifts - x and y			
Algorithm	APCER	BPCER	Accuracy
SVM linear	19±2	15±1	66±2
SVM poly	32±1	8±0	59±1
SVM rbf	18±2	10±2	72±1
RF 500	16±2	12±3	72±2

From these measurements it can be seen that the Support vector machine algorithm with a Radial basis function kernel and the Random Forest algorithm with 500 trees performed best.

Then we considered how bona fide shifts can be differentiated when pose variation is included. We expect that bona fide images should have some shifts that are symmetrical. Including the length of all the shifts as feature vectors could increase the performance as the length of some shifts should be very similar if not identical because of symmetry. It could also result in better performance as more feature vectors can improve performance for SVM with linear kernels.

68 shifts - x, y and length			
Algorithm	APCER	BPCER	Accuracy
SVM linear	17±1	13±2	70±1
SVM poly	16±2	13±2	70±1
SVM rbf	16±2	11±2	74±1
RF 500	16±2	13±2	71±3

An increase in performance can be seen across the SVM machine learning algorithms but especially the SVM algorithms with the linear and polynomial kernel as those kernels perform better with more features. A slight decrease can be seen for the Random Forest algorithm. This indicates that adding the length of the shifts likely does not increase the performance and that the performance was increased by adding more feature vectors in general.

We also tested the performance by adding the direction of the shift in degrees. In a bona fide image certain shifts should have a very similar if not the same direction, again because of the expected symmetry.

68 shifts - x, y and angle			
Algorithm	APCER	BPCER	Accuracy
SVM linear	19±2	15±1	65±2
SVM poly	26±1	12±1	63±1
SVM rbf	18±2	11±2	71±1
RF 500	16±2	12±2	72±2

For most algorithms adding the angle of the shift in degrees slightly decreased the performance.

When analyzing which landmarks could give an indication if an image is morphed, we hypothesized that it would be the x shift of the eye corner landmarks. An eye should only increase or decrease in width due to morphing. The height of the eye is heavily dependent on expressions so it would specifically be the shift in the horizontal direction.

Due to pose variation, shifting in the same direction is expected. To distinguish shifting in the same direction versus shifting in opposite direction, the x coordinates of the vector shift are multiplied. A shift in opposite direction results in a negative value and a shift in the same direction results in a positive value. Only negative values will be useful as those can indicate that the width of the eye possibly changed, suggesting that the image is morphed.

We plotted 200 bona fide images and 200 morphs resulting in the matrix scatter plot seen in figure 4. LM36_x*LM39_x is the multiplication of the eye corner shifts along the x direction for the left eye. Similarly, LM42_x*LM45_x for the right eye. In the histograms the values per eye are plotted. It can be concluded that shifts in opposite direction for only one eye does not mean anything as the histograms do not indicate any separation. When the second eye is also taken into account, the values can be seen in the scatter plots. In these plots we can see that if both eyes have eye corner shifts in opposite direction and so both have a negative value, that 6 out of 8 images are morphed. This indicates that an image is likely morphed when the eye corner shifts are in opposite direction for both eyes. As there are only 8 cases in total out of 400 this conclusion should be taken lightly.

68 shifts - x, y and x*x of eye corner shifts			
Algorithm	APCER	BPCER	Accuracy
SVM linear	20±2	15±1	65±2
SVM poly	32±1	9±0	59±1
SVM rbf	18±2	10±2	72±1
RF 500	16±2	12±2	72±2

We did not measure an increase in performance when including x*x of the eye corner shifts. It could be because only adding two extra feature vectors would not have much of an impact or because the positive values complicated the classification.

For decision tree algorithms feature vector importance can be calculated. The features that are more important are more useful to classify the image. Using the same data to train a decision tree algorithm and then computing the feature vector importance resulted in the top 10 features seen below. The dlib [9] landmarks are counted from 0 to 67. The numbered landmarks can be seen in figure 6 in the appendix.

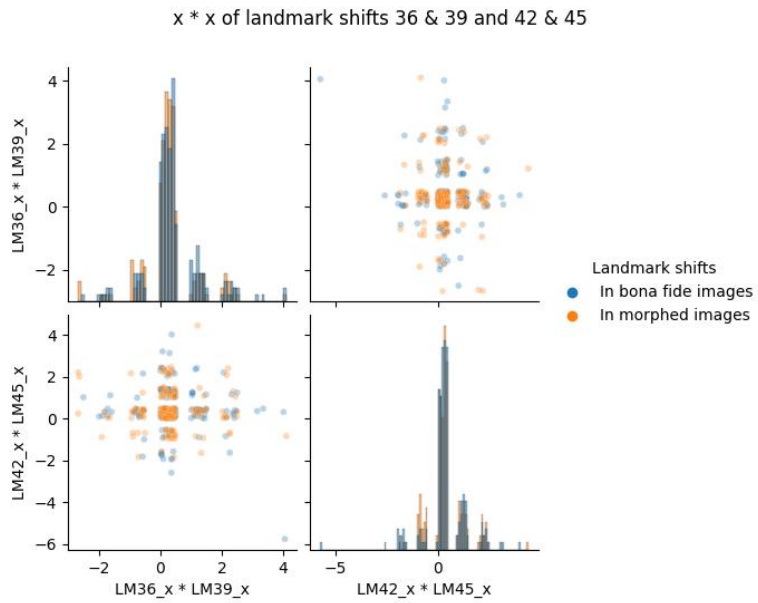


Figure 4. Matrix scatter plot of x*x

Feature vector importance top 10		
Feature vector	Score	Facial feature
17 x	0.039	eyebrow
19 x	0.036	eyebrow
39 x	0.021	eye corner
36 x	0.020	eye corner
35 x	0.019	nose nostril
53 x	0.018	mouth
45 y	0.015	eye corner
1 x	0.014	face contour
40 y	0.014	eye
10 x	0.014	face contour

As seen above we can conclude the eye corner shifts in the horizontal direction could indeed be quite important feature vectors (Nr.3 and 4 in the list).

6.2 Pose variation

To test how more pose variation in the training set influences the performance, we will use two exact same classifiers. One will be trained on a data set with minimal pose variation and the other will be trained on a data set with varied pose variation. We will then test the algorithms on a separate data set with varied pose variation.

As previously measured, the Random forest algorithm with 500 trees performed best in our case. Therefore, this algorithm was chosen to measure the performance impact of including pose variation in the training set.

In figure 5 an APCER vs BPCER plot can be seen. Here the APCER (false positive rate) is plotted against the BPCER (false negative rate). Ideally, these rates should be very low. From this graph can already be deduced that the algorithm trained with varied pose variation outperforms the algorithm trained with minimal pose variation. The measurements are again in percentages.

More precise measurements of the performance can be seen in the table below.

Pose	D-EER	BPCER10	BPCER20	BPCER100
Minimal	71	100	100	100
Varied	29	62	79	87

The results conclude that including more pose variation in the training set indeed increases the performance significantly.

As the algorithm trained on minimal pose variation has not seen images with more pose variation, the performance is very poor. In this case the D-EER is 71 percent, which is worse than chance which would result in a D-EER of 50 percent. When a support vector machine with a rbf kernel was used as the classifier instead, it resulted in very similar results.

7. CONCLUSION

For this research we explored multiple possibilities to increase the performance of landmark based differential morphing attack detection (D-MAD). The methods are based on differentiating landmark shifts due to morphing and due to pose variation, as pose variation can reduce the performance of MAD based on landmark shifts. One method to improve performance was to select certain feature vectors and algorithms. The other method was to include more pose variation in the training set.

The feature vector sets that were tested did not increase the performance. We hypothesized that bona fide images should have certain shifts that are symmetrical to each other. Adding feature vectors to better indicate the symmetry of landmark shifts did not necessarily result in an increase in performance. Any increase in performance was likely due to adding extra features in general.

We also hypothesized that morphs could be classified using the eye corner shifts, as the eyes should not increase or decrease in width. The feature importance indicated that horizontal eye corner shifts could indeed be important feature vectors to classify morphed images, although the feature vector selection did not result in a better performance.

Further work could explore different feature selections as

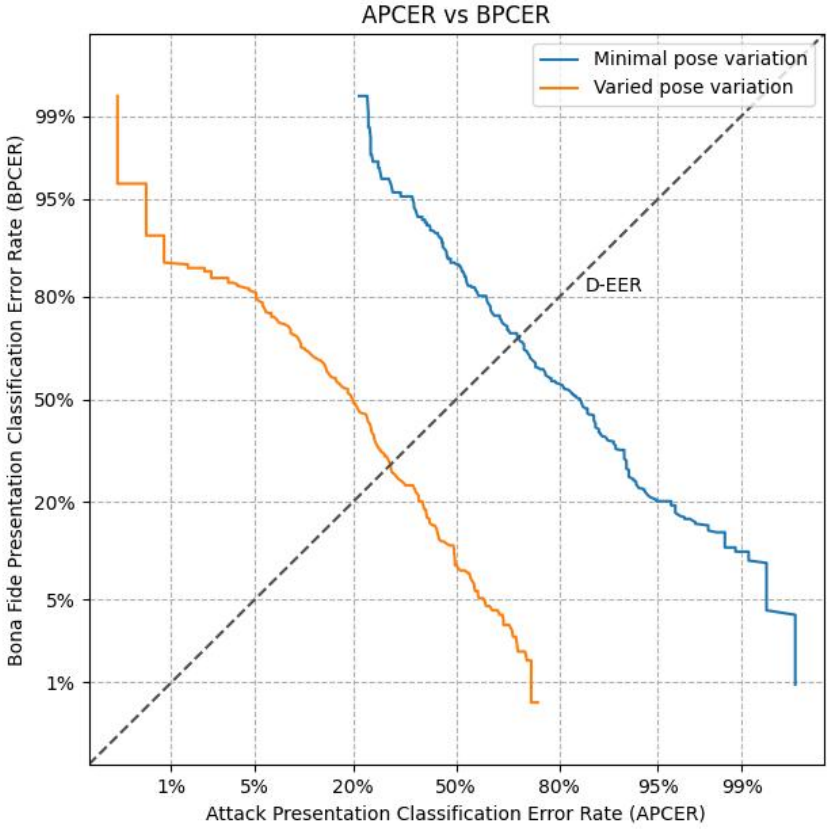


Figure 5. APCER vs BPCER plot

some feature vectors could impact the performance. For example also including the coordinates of the starting point of the shift. As it was hard to find a feature vector selection based on the 68 landmarks that can increase the performance, it might be better to put effort into more promising possibilities instead.

When comparing the performance of different classifiers, it was clear that the support vector machine with a rbf kernel and the random forest with 500 trees performed best. As these outperformed the support vector machine with the linear kernels it is clear that non-linear classifiers are preferred for this problem. The difference in performance between the support vector machine with the rbf kernel and the random forest with 500 trees was minimal.

Including more pose variation in the data set did result in a significant performance increase. Therefore adding more pose variation in the data set should definitely be considered for future work, as realistic scenarios will most likely also include more pose variation. Only including a minimal amount of pose variation in the data set results in a very poor performance when tested on a data set with a more realistic amount of pose variation. This is likely because images with more pose variation can be considered unseen by the algorithm and algorithms perform worse on unseen data.

For further research it would be interesting to use neural networks for MAD based on landmark shifts. This requires a large dataset but generally makes more accurate predictions compared to the supervised algorithms that were tested.

As the PUT database was used which did not include much natural variation like aging, no conclusion can be made on exactly how much the performance can be increased. Not having a realistic amount of natural variation in the data set could potentially have affected the differences in performance that were measured.

8. REFERENCES

[1] N. Damer, V. Boller, Y. Wainakh, F. Boutros, P. Terhörst, A. Braun, and A. Kuijper. Detecting face morphing attacks by analyzing the directed distances of facial landmarks shifts. In *German Conference on Pattern Recognition*, pages 518–534. Springer, 2018.

[2] J. Deng, J. Guo, N. Xue, and S. Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4690–4699, 2019.

[3] M. Ferrara, A. Franco, and D. Maltoni. The magic passport. In *IEEE International Joint Conference on Biometrics*, pages 1–7. IEEE, 2014.

[4] M. Ferrara, A. Franco, and D. Maltoni. On the effects of image alterations on face recognition accuracy. In *Face recognition across the imaging spectrum*, pages 195–222. Springer, 2016.

[5] M. Ferrara, A. Franco, and D. Maltoni. Face demorphing. *IEEE Transactions on Information Forensics and Security*, 13(4):1008–1017, 2017.

[6] M. Ferrara, A. Franco, and D. Maltoni. Face morphing detection in the presence of printing/scanning and heterogeneous image sources. *arXiv preprint arXiv:1901.08811*, 2019.

[7] Frontex. Best practice technical guidelines for automated border control (ABC) systems, 2015.

[8] A. Kasinski, A. Florek, and A. Schmidt. The put face database. *Image Processing and Communications*, 13(3-4):59–64, 2008.

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

[10] R. S. Kramer, M. O. Mireku, T. R. Flack, and K. L. Ritchie. Face morphing attacks: Investigating detection with humans and computers. *Cognitive research: principles and implications*, 4(1):1–15, 2019.

[11] M. Ngan, M. Ngan, P. Grother, K. Hanaoka, and J. Kuo. *Face recognition vendor test (frvt) part 4: Morph-performance of automated face morph detection*. US Department of Commerce, National Institute of Standards and Technology, 2020.

[12] D. Ortega-Delcampo, C. Conde, D. Palacios-Alonso, and E. Cabello. Border control morphing attack detection with a convolutional neural network de-morphing approach. *IEEE Access*, 8:92301–92313, 2020.

[13] F. Peng, L.-B. Zhang, and M. Long. FD-GAN: Face de-morphing generative adversarial network for restoring accomplice's facial image. *IEEE Access*, 7:75122–75131, 2019.

[14] P. J. Phillips, P. J. Flynn, T. Scruggs, K. W. Bowyer, J. Chang, K. Hoffman, J. Marques, J. Min, and W. Worek. Overview of the face recognition grand challenge. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, volume 1, pages 947–954. IEEE, 2005.

[15] R. Raghavendra, K. Raja, S. Venkatesh, and C. Busch. Face morphing versus face averaging: Vulnerability and detection. In *2017 IEEE International Joint Conference on Biometrics (IJCB)*, pages 555–563. IEEE, 2017.

[16] K. Raja, M. Ferrara, A. Franco, L. Spreeuwers, I. Batskos, F. d. W. M. Gomez-Barrero, U. Scherhag, D. Fischer, S. Venkatesh, J. M. Singh, et al. Morphing attack detection–database, evaluation platform and benchmarking. *arXiv preprint arXiv:2006.06458*, 2020.

[17] R. Ramachandra, S. Venkatesh, K. Raja, and C. Busch. Towards making morphing attack detection robust using hybrid scale-space colour texture features. In *2019 IEEE 5th International Conference on Identity, Security, and Behavior Analysis (ISBA)*, pages 1–8. IEEE, 2019.

[18] D. J. Robertson, R. S. Kramer, and A. M. Burton. Fraudulent id using face morphs: Experiments on human and automatic recognition. *PLoS One*, 12(3):e0173319, 2017.

[19] U. Scherhag, D. Budhrani, M. Gomez-Barrero, and C. Busch. Detecting morphed face images using facial landmarks. In *International Conference on Image and Signal Processing*, pages 444–452. Springer, 2018.

[20] U. Scherhag, L. Debiasi, C. Rathgeb, C. Busch, and A. Uhl. Detection of face morphing attacks based on prnu analysis. *IEEE Transactions on Biometrics, Behavior, and Identity Science*, 1(4):302–317, 2019.

[21] U. Scherhag, C. Rathgeb, J. Merkle, and C. Busch. Deep face representations for differential morphing attack detection. *IEEE Transactions on Information Forensics and Security*, 15:3625–3639, 2020.

[22] S. Venkatesh, R. Ramachandra, K. Raja, and C. Busch. Single image face morphing attack detection using ensemble of features. In *2020 IEEE 23rd International Conference on Information Fusion (FUSION)*, pages 1–6. IEEE, 2020.

[23] S. Venkatesh, R. Ramachandra, K. Raja, L. Spreeuwers, R. Veldhuis, and C. Busch. Detecting morphed face attacks using residual noise from deep multi-scale context aggregation network. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 280–289, 2020.

APPENDIX

A. IMAGES

Figure 6. Facial landmarks of library dlib.

This image shows the numbered facial landmarks extracted by dlib [9].

B. RESULTS

Some analyses did not give any meaningful results and so were not referenced in the paper. Therefore these results are shown here.

In figure 7 a scatter plot can be seen of the landmark shifts for 200 bona fide images. For each landmark shift, x is plotted along the x-as and y is plotted along the y-ax. In figure 8 the same scatter plot can be seen for 200 morphed images. These graphs did not indicate a possible difference between bona fide shifts and morph shifts.

Analysing the shift in horizontal direction of the eye corner landmarks resulted in the matrix plot seen in figure 9. No clear separation could be seen between the bona fide shifts and the morph shifts. After this we focused on shifts specifically in opposite direction.

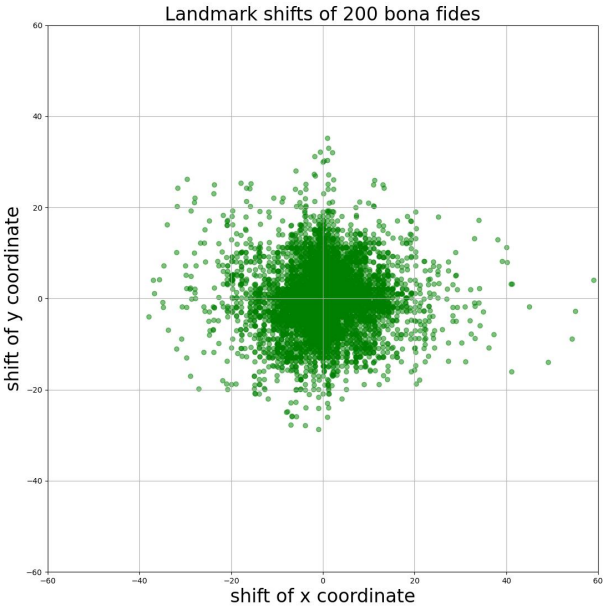


Figure 7. Scatter plot of 200 bona fides.

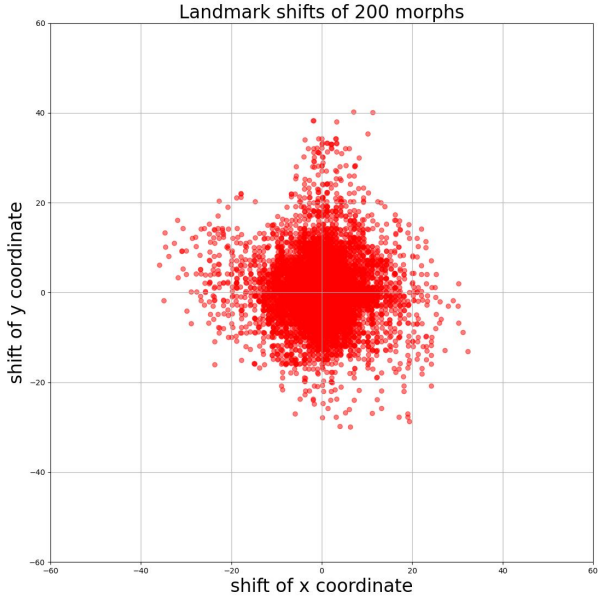


Figure 8. Scatter plot of 200 morphs.

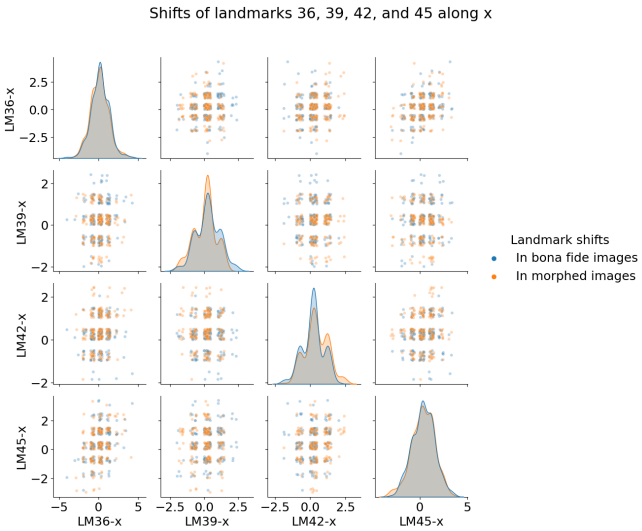


Figure 9. Matrix scatter plot of eye shifts x.