Morphing Detection Based on Regional Analysis of Local Frequency Content

Jelle Meijer
University of Twente
P.O. Box 217, 7500AE Enschede
The Netherlands
j.j.w.meijer@student.utwente.nl

ABSTRACT

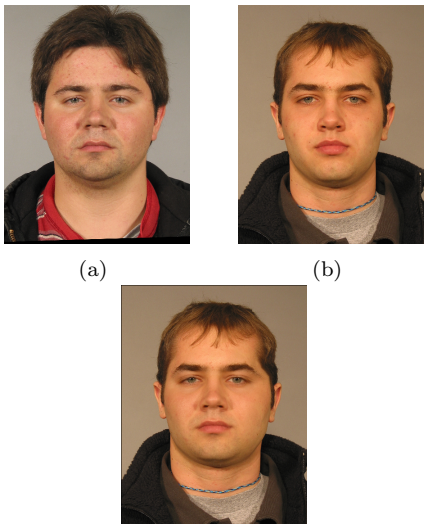
Face recognition software is known to be vulnerable to a presentation attack in the form of face morphing. Face morphing detection is an active field of research. Creating strong face morphing detection algorithms will make face recognition software more robust. This paper investigates how regional analysis of the frequency spectrum of face images can be used to detect morphs in both a differential and non-differential setting. Three methods are explained and assessed for their performance. The first method utilizes the Kullback Leibler Divergence. The second is a Support Vector Machine (SVM). The third a Deep Feed Forward Neural Network (DFF). The latter two are trained on the frequency spectrum. The Kullback Leibler Divergence proved to be not discriminate enough to classify morphs. Both the SVM and DFF were able to detect morphs with an accuracy of around 80%.

Keywords

Face Morphing, Facial Recognition, Frequency Domain, Machine-Learning, Presentation Attack.

1. INTRODUCTION

Face Recognition Technology (FRT) has seen world-wide adoption in various industries, such as surveillance and access control. Face recognition is being utilized in automated border controls to increase efficiency through automation.[3] In automated border controls eGates are responsible for veryifying the identity of the traveler. FRT is used in eGates to verify that an individual matches the image in his/her passport.
In 2014 *Ferara et al.* showed in his paper titled 'the magic passport' a vulnerability of face recognition software in the form of a presentation attack.[5] These attacks were later given the name: face morphing attacks. Face morphing attacks allow an attacker to create an image, which will fool face recognition systems to accept two different individuals. This would allow a criminal to cross the border under the name of an accomplice. Furthermore, it was shown by *Robertson et al.* that human observers also have difficulty differentiating between morphed and non-morphed images.[19] Figure 1 shows an example of a morphed face image. Note how the morphed face in Figure 1c resembles both the face of 1a and 1b.



(a) (b)

(c) Morph

Figure 1: Example of a morphed face image c created from a and b

Morphing detection tries to deal with the threat through detection of morphed images. The task of a morph detection algorithm is to classify an image either as morphed or bona-fide. Several different techniques have been proposed to detect morphs. But as of yet, there is no robust morphing detection method solution available. In general two classes of morph detection exist.[21] No-reference morph detection and differential morph detection. No-reference morph detection takes a single image as its input. It has a single point of reference to classify an image as either morphed or non-morphed. Contrary to no-reference morphing detection, a differential morphing detection approach is supplied with two face images. A trusted reference image and a questioned image.

This paper investigates how regional analysis of the grayscale frequency content of facial images can be used to detect morphed images. Furthermore, it is investigated which classifications methods are suitable in both a differential and non-reference setting. Some facial regions may be more affected by morphing than others. Therefore, different regions are compared in their effectiveness at distinguishing morphed images.

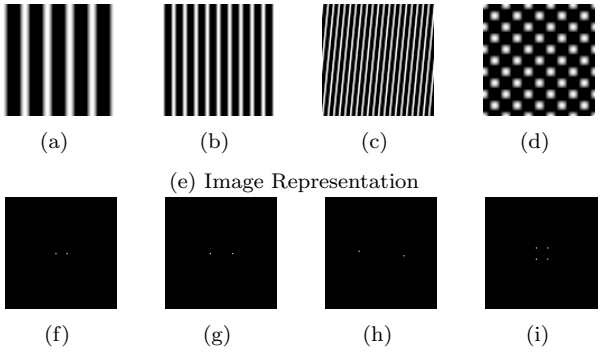
2. BACKGROUND

28^{th} Twente Student Conference on IT Febr. 2^{nd}, 2018, Enschede, The Netherlands.

A large range of different variations of morphing methods exist. The realism of the created morphed images is heavily dependent on the morphing pipeline used. The morphing process can create disocclusions and artifacts.[22][9] An attacker can apply multiple techniques to increase the quality of a morph. This can be done by altering the morphing process itself or by editing the created morphed images in post. Artifacts tend to appear when the alignment of landmarks is incomplete or imperfect. Areas around the face, such as hair and the background are especially vulnerable to this.A common way of preventing this is through splicing. Splicing only morphs the inner part of the face and leaves the outer part unaffected. This however, does not prevent artifacts from appearing within the face. This however does not prevent artifacts from appearing within the face.

Multiple morphing detection approaches have been proposed. Multiple studies have looked at image descriptors to determine whether an image is a morph. *Raghavendra et al.* trained a Support Vector Machine (SVM) on the Binarized Statistical Image Features (BSIF) of images.[16] Other studies have also used Local Binary Patterns (LBP) and Scale-Invariant Feature Transform (SIFT) to detect morphs.[18][20]
Photo Response Non-Uniformity (PRNU) is a noisy pattern in images as a result of imperfections in the camera's sensor. This feature of images has also been used successfully to detect morphs as shown by the paper of *Debiasi et al.*[4] In their study it was noticed that higher frequencies would get lost as a result of the morphing pipeline.

As mentioned, morphing also affects the frequency content of an image. Any analog electric signal can be seen as a composition of frequencies. The same applies to images. Images can also be considered to be compositions of various frequencies. To inspect the frequencies of an image one can apply the 2-Dimensional Fourier Transform. This converts the image into it's frequency domain representation. In the frequency domain each frequency is represented by a complex number. Where the absolute value is the frequency's magnitude and the angle the frequency's phase. In the case of images, frequencies take the form 2-D Sinusoidals. Figure 2 shows different images consisting of one or two frequencies and their respective frequency domain representation (magnitude only).



(a) (b) (c) (d)

(e) Image Representation

(f) (g) (h) (i)

(j) Frequency Domain Representation

Figure 2: Images and their respective frequency domain representation

Neubert et al. used the magnitude of the frequency domain to detect morphs. They separated the frequency domain into 25 blocks and took the mean magnitude as input for their classifier. Their experiments showed that analysis of the frequency domain could be used as a morphing detection approach. They were able to detect morphs with an accuracy of 75.2% on eMRTD images.[12]
This paper extends upon the previously done research by investigating how regional analysis of the frequency domain can detect morphs.

3. FREQUENCY DOMAIN CLASSIFICATION

3.1 Regions

Different regions of the face have different characteristics. The chin for example can contain facial hair and the bridge of the nose tends to have two vertical edges. In this paper a method is proposed that extracts different regions. A classifier can be trained on a single region. The performance of different classifiers can say something about a region's ability to recognize morphs.
To reliably extract different regions one could use landmarks. Landmarks can be detected using a landmark detection algorithm. The different landmarks can then define different regions of the face. One possible method of creating square regions is by having different landmarks define a central point from which a square is created. To ensure that the regions are consistent it is advised to ensure that the faces are aligned and scaled the same.

3.2 Frequency Extraction

To extract the frequency domain of an image the 2-Dimensional Fourier Transform can be used. The standard Fourier Transformation however is defined as a continuous function. Solving continuous functions computationally can be difficult and resource intensive. Fortunately, there are Discrete Fourier Transformation algorithms available. A common used algorithm is the Fast Fourier Transformation (FFT). The FFT can also be used to compute the frequency domain of a 2-dimensional function. The 2-Dimensional Fast Fourier Transform (2D-FFT) can be used to transform images to their respective frequency domain representation. This results in a matrix where frequencies are represented by complex numbers. Taking the absolute values of all elements results in a matrix where the frequencies are represented by their magnitude. The frequency domain is a powerful tool for image-processing.
Taking the magnitude of the frequency domain representation of an image results in as many magnitudes as there are pixels. This can be a lot of input for a classifier to train on. To shrink the possible input size *Neuber et al.*[12] divided the frequency into 25 distinct blocks and took the mean magnitude of each block. By doing so they took the average magnitude of frequencies with approximately the same direction and wavelength.
This paper proposes an alternative method of shrinking the input size. A power spectrum can be computed from the frequency domain. By doing so the directional data of the frequency is discarded and only the wavelength is considered. This results in a graph containing the mean magnitude of frequencies with the same wavelength. An example is of such a graph is given in Figure 4. Extracting the power spectrum can be done by computing the frequency domain of an image and shifting the zero-component to the center. Finally, computing the averaging magnitudes along a circle of radius $r = w$ gives the mean magnitude of frequencies with wavelength w. Figure 3 illustrates this

process. As is common in image-processing the values at the center point can be ignored, because it contains only the average intensity. Therefore, values at $r = 0$ can be discarded. One disadvantage of this method is that it ignores values at the corners of the frequency domain, because circles can't reach these values. This method shrinks the input size significantly. For an image of size 512 x 512 the frequency domain would contain $512 \cdot 512 = 262144$ magnitudes. With this method the input size is decreased from 262144 values to only $512/2 = 256$ values.

3.3 Pre-processing

There are multiple different ways of pre-processing the obtained spectral data. Different pre-processing steps might improve the performance of a classifier. This section describes various possible pre-processing steps. The steps are listed in the same hierarchical order that they should be applied from top to bottom.

Radius Weighted Mean.

Circles consisting of a larger radius (in pixels) consist of more pixels in the frequency domain than circles of a smaller radius. It is clear from Figure 3 that larger circles will contain more magnitudes than those of smaller size. This means that the magnitudes of frequencies with smaller wavelengths are represented by a smaller sample size compared to those of larger wavelength. To include this information it is possible to weigh the mean by its. It is proposed to multiply the mean each magnitude by the radius of its circle.

Logarithm.

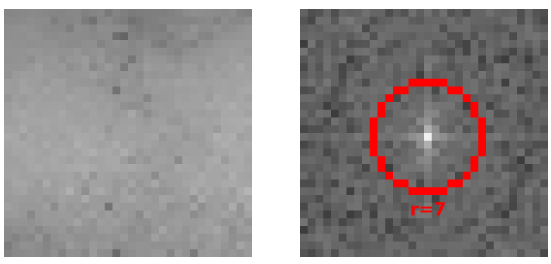
The magnitudes of lower frequencies are often several orders larger than the magnitudes of higher frequencies. For example, a random sample's mean magnitude at r=1 is equal to 2830 and 87 at r=16. It is possible to apply the logarithm to the mean magnitude in order to decrease the large differences, while still maintaining the relative order.

Differential vs. Non-reference.

Differential morphing detection can be useful for passport image verification during the passport application process. The trusted image might be a point of reference for determining whether the questioned image has been morphed. A non-reference approach does not have the advantage of having a reference point, but non-differential morphing detection is more flexible as no trusted image is required.

Normalization.

Normalization of input data is often performed in order to transform the input to a common scale. There are two



(a) (b)

Figure 3: 2D-FFT Conversion of the chin region.

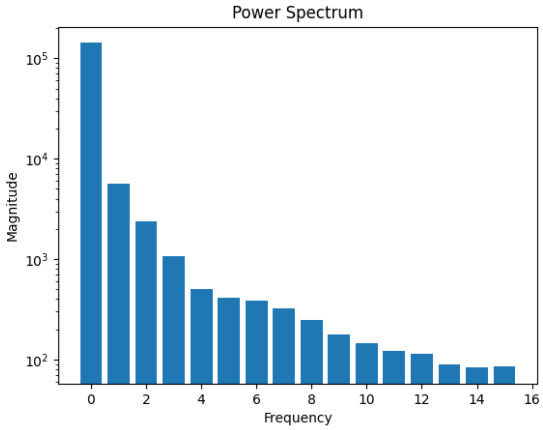


Figure 4: Power Spectrum of an image

common ways of normalizing input vectors. The input vectors can be either L1-normalized or L2-normalized. The equations for the norms are:

$$L1(w) = \sum_i |w_i|$$

$$L2(w) = \sqrt{\sum_i w_i^2}$$

Where w is a vector containing for example the intensity of different frequencies.
After either the L1 or L2 norm is calculated the vector us divided by the norm, hence the elements of the input vector are either:

$$v_i = \frac{w_i}{L1(w)}$$

or

$$v_i = \frac{w_i}{L2(w)}$$

Depending on the chosen normalization method.

3.4 Kullback Leibler Divergence

One special proposed method is the classification of morphs through the use of the Kullback Leibler Divergence. The power spectrum can be interpreted as a probability distribution. The Kullback Leibler Divergence (KBD) is a measure of how two probability distributions are different from each other. The underlying assumption is that the KBD will be higher when the questioned image is a morph. The questioned and trusted image need to be each L1-Normalized. After normalization the KBD is computed. This results in a single number. It is possible to set a threshold for which all entries with a KBD higher than the thresholds will be classified as morphs. This makes it possible to create a Detection Error Trade-off Curve The KBD approach is only possible in a differential setting as both the trusted and questioned image are required.

4. EXPERIMENTS

In total three different experiments were conducted. Two experiments tested two different classifiers on data with various combinations of pre-processing methods. The third experiment investigated the viability of using the Kullback Leibler Divergence to detect morphs.

4.1 Data

The classifiers were trained on a selection of morphed faces and bona-fide faces from the *FRGC_V2.0* and *IST-EURECOM* dataset. It is of importance that the pictures resemble legitimate passport photos. Therefore, photos were manually selected, such that the image quality is close to the quality requirements of passport photos. Furthermore, the images have been aligned and resized in order to be more representative of passport photos. Both datasets contain multiple images of each individual. This makes it possible to create pairs, which is necessary for a differential morphing detection approach.

4.1.1 Morphing Pipeline

Both the *FRGC* and *IST-EURECOM* dataset do not contain morphed images. Therefore, it was necessary to construct a morphing pipeline in order to create morphed images.
Generally, morphing is a three-step process. First, a correspondence for two faces is established. The most common technique is to detect sets of corresponding landmarks on the the face. Second, an average correspondence is created. This step is called warping. Third, the image colors are merged by blending the two images.[21] The realism of the created morphs is heavily dependent on the quality of the morphing pipeline. The morphing process also affects the frequency content of an image.
This study uses a morphing pipeline based on the morphing attack type II as described by *Kraetzer et al.*[8] with some minor modifications. One important attribute of this morphing method is that it uses splicing. Only the area inside of the face is morphed and the outside stays unaffected. When two individual's faces are not properly aligned the blending step can create ghost artifacts. Areas around the face like hair and clothes are very prone to having ghost artifacts. Splicing is used to prevent this from happening. The primary difference compared to *Kraetzer et al.*'s method is that STASM[11] is used for landmark detection. Morphing faces with large differences results in less realistic morphs. To prevent this morphed pairs were selected according to their similarity scores. The similarity scores were computed using *dlib*'s face recognition module.[6] To further increase the realism of the morphs Poisson image editing was applied in post.[15] Poisson image editing is useful when dealing with situations were images are blended together.

The total number of pairs consist of 203 bona-fide *FRGC* pairs, 100 bona-fide *IST-EURECOM* pairs, 628 morphed *FRGC* pairs and 268 morphed *IST-EURECOM* pairs. Note that the resulting dataset is imbalanced and that there are more morphed pairs than bona-fide pairs. There are more morphs, because some individuals appear multiple times in different pairs. For the experiments using machine-learning 20% was used for testing and 80% for training.

4.1.2 Region Selection

In various locations of the face different regions were selected. The selected regions were the chin, eyes, cheeks, cheekbones, nose and forehead. For every location a 32 x 32 region was extracted. Landmarks needed to be detected to be able to locate the different regions. This was done using *dlib*'s 68 shape landmark predictor. The center of every region consists of a weighted average of three different landmarks. From this center a 32 x 32 rectangle is constructed and the grayscale content extracted. Figure 5 shows the different regions highlighted. Each region has been labeled with a prefix. Table 1 displays the mapping of regions to prefixes. The morphing pipeline makes use of splicing. This means that practically the forehead should stay untouched and should not be indicative of the region being part of a morphed face. This region serves as a zero-measurement for the experiments.

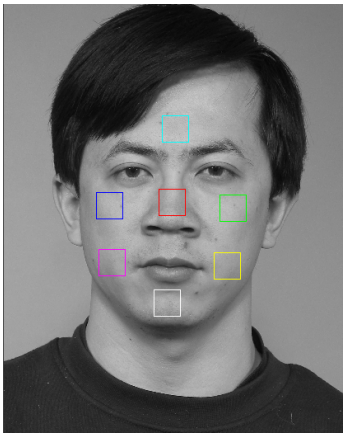


Figure 5: Region Detection

Colour	Prefix	Name
	NO	Nose
	LB	Left Cheekbone
	RB	Right Cheekbone
	LC	Left Cheek
	RC	Right Cheek
	FH	Forehead
	CH	Chin

Table 1: Region Mapping

4.1.3 Fourier Transformation

After extraction, the regions are converted to their frequency domain representation using *Numpy*'s Discrete FFT module[13]. After conversion the zero-frequency point is shifted to the center. After the shift the absolute values of the complex numbers are computed. To obtain the power spectrum from the frequency domain the images are polar warped. This is done using *OpenCV2*[2]. As a last step the mean magnitude is computed of frequencies with the same wavelength.

Two different machine-learning algorithms are used throughout the experiments. A Support Vector Machine (SVM) and a Deep Feed Forward Neural Network (DFF). Both algorithms have been used successfully for the detection of morphs in the past.[16][17]. For each region a separate classifier is trained.

4.2 SVM

A SVM was tested for different combination of the preprocessing steps as described in Section 3.3. An SVM in its pure form is only capable of learning linearly separated classes. Since, the classes of morphs are not linearly separated this would make a SVM useless for detecting morphs. Fortunately, it is possible to classify non-linearly separated classes using kernels. Throughout the experiment a polynomial kernel is used. *Sklearn*'s SVM implementation is used.[14] To prevent overfitting the SVM uses L2 Regularization with the parameter set to 1.0 *Sklearn*'s implementation also provides functionality for the prediction of probabilities using Platt scaling. Platt scaling uses

5-fold cross-validation internally in order to predict probabilities. If a SVM is trained on an imbalanced dataset it tends to become biased to the majority class. To counteract this the SVM is trained with class weights. This should prevent the SVM from being overly biased towards the morphs class.
Different parameters of the SVM were fine-tuned for the best performance after which the same paramteres were used for all experiments. For all combinations of the different pre-processing mentioned in Section 3.3 the performance of every region was measured and evaluated. It was also tested whether the performance of the differential approach would be different from the non-reference approach. In the case of the differential approach the input vectors are concatenated prior to normalization.

4.3 DFF

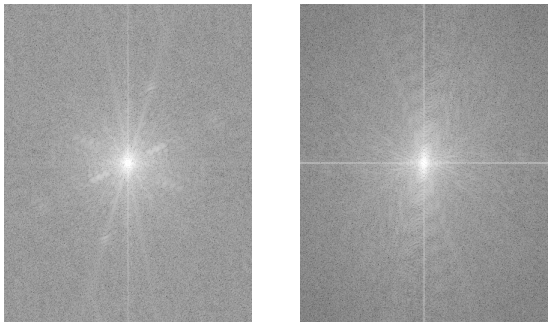
The second experiment uses a neural network in order to make prediction. A Deep Feed Forward (DFF) neural network is used. The implementation was taken from *Tensorflow's Keras*[1] package. Often Deep Convolutional Neural Networks (CNN) are used for classification tasks when images are involved. CNN's are very good at interpreting image patterns. In this study however, a standard DFF is used. The reasoning being that the frequency domain does not contain very clear patterns to interpret. A comparison of the frequency domain of a trusted and morphed image is shown in Figure 6. It is clear from the image that there are no obvious patterns to pick up on. Furthermore, the input size of a region's frequency domain would be $2 \cdot 32 \cdot 32 = 1024$ neurons. This means that there are a lot of learnable parameters. Given the relatively small dataset this would not be viable and most likely lead to overfitting. On the contrary, the spectral data only requires a maximum input size of $2 \cdot 16 = 32$ neurons. Because of this reason the frequency spectrum is used as input instead of the extracted frequency domain's image.
The DFF has either 16 or 32 input neurons depending on whether the approach is differential or non-reference. 2 hidden layers with a *sigmoid* activation functions are used with a varying amount of neurons. The output consists of two neurons with a *softmax* activation function. One neuron for signifying a bona-fide case and one neuron signifying a morphed case. The learning process is optimized with the *Adam*[7] optimizer with an initial learning rate of 0.01. 20% of the training set is used for validation. Since DFF's are also sensitive to imbalanced datasets, class weights are used. The different neural networks are trained for 100 epochs. All neural networks would stabilize after around 100 epochs of training. More training would result in overfitting and no additional gain. Three different combinations of pre-processing steps have been tested with hidden layers of varying size. The results are listed in 5.3.

4.4 Kullback Leibler Divergence

As a final experiment the Kullback Leibler Divergence (KBD) has been investigated. For different pairs the KBD was computed. This results in a single number per pair. If the KBD of a pair was above the threshold it would be considered to be a morph. This threshold allows the computation of a DET-Curve for the KBD approach.

4.5 Performance Metrics

For all experiments Detection Error Trade-off Curves (DET-Curve) are computed. These curves display the trade-off between the APCER and BPCER. APCER stands for Attack Presentation Classification Error Rate (proportion of attack presentation classified as bona-fide) and BPCER stands for Bona-Fide Classification Error Rate (proportion of bona-fide presentations classified as attack presentations).[10] DET-Curves are a useful tool for comparing the performance of different classifiers. The Area Under Cuve (AUC) also sometimes is used as a metric of performance.



(a) Trusted (b) Morphed

Figure 6: Frequency Domain image of a pair.

5. RESULTS

This section lists and explains the results of the various performed experiments. Interpretation of the results will be covered in section 6.

5.1 Kullback Leibler Divergence

Table 2 lists the sample mean and sample variance of the KBD for the bona-fide pairs and presentation attack pairs. BF and M stand for Bona-Fide and Morphed respectively. Furthermore, Figure 7 lists the DET-curves of the different regions. The number after the region prefix is its respective AUC. The dotted line represents the values were $APCER = BPCER$. A classifier which classifies each sample randomly with a 50/50 chance would lie on approximately on this line.

Prefix	BF Mean	M Mean	BF Var	M Var
CH	0.002390	0.004865	0.000018	0.000061
FH	0.004119	0.004349	0.000197	0.000231
LB	0.001654	0.002469	0.000007	0.000010
LC	0.003767	0.004820	0.000057	0.000064
NO	0.001778	0.002943	0.000010	0.000012
RB	0.001620	0.002507	0.000005	0.000008
RC	0.003468	0.004439	0.000057	0.000054

Table 2: KBD Sample Mean and Sample Variance

Table 2 shows the sample mean and sample variances for the bona-fide pairs and presentation attack pairs.

5.2 SVM

Table 3 shows the SVM's performance for various pre-processing methods. The columns indicate whether a method is applied. **WM** indicates whether a weighted mean was used, **N** indicates what normalization method was used. and **Diff** indicates whether the approach was differential. The Accuracy (ACC), ACPER and BPCER are averaged over all regions.
The ability of predicting probabilities allows for the creation of a threshold. Any samples with a probability of being a morph above the threshold will be classified as morph. This makes it possible to construct a DET-Curve. Figures 8 and 9 show the DET-Curves of the regions for experiment 7 and 12 respectively.

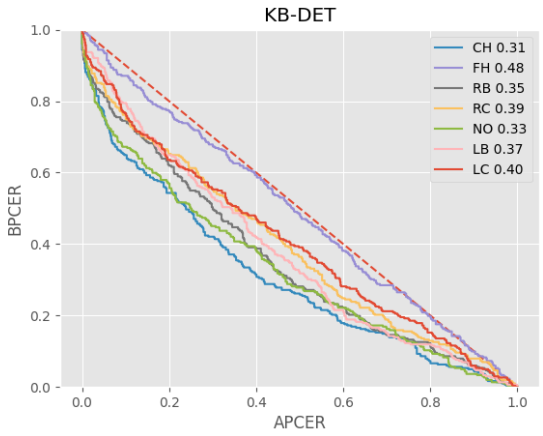


Figure 7: DET-Curves for Kullback Leibler Divergence

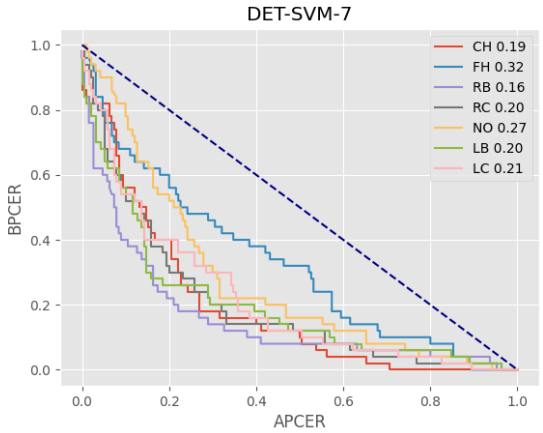


Figure 8: DET-Curves of different regions for differential, L2 normalized approach (SVM)

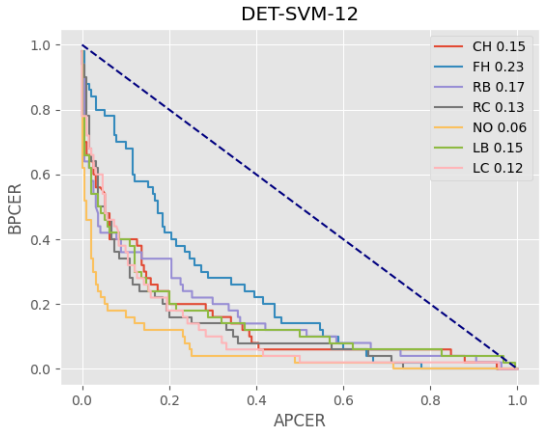


Figure 9: DET-Curves of different regions for non-reference, logarithmic, L2-normalized approach (SVM)

ID	Diff	Log	N	WM	ACC	APCER	BPCER
1	yes	no	L1	yes	0.629	0.332	0.517
2	yes	yes	L1	yes	0.612	0.405	0.323
3	yes	no	L2	yes	0.791	0.305	0.517
4	yes	yes	L2	yes	0.651	0.420	0.306
5	yes	no	L1	no	0.726	0.278	0.260
6	yes	yes	L1	no	0.7	0.283	0.363
7	yes	no	L2	no	**0.736**	**0.274**	**0.180**
8	yes	yes	L2	no	0.705	0.277	0.366
9	no	no	L1	no	0.71	0.280	0.329
10	no	no	L2	no	0.737	0.169	0.397
11	no	yes	L1	no	0.795	0.160	0.183
12	no	yes	L2	no	**0.799**	**0.153**	**0.186**

Table 3: Accuracy, APCER, BCPER for different pre-processing methods (SVM)

5.3 DFF

ID	PP	HLN	ACC	APCER	BPCER
1	7	8	0.752	0.259	0.206
2	7	16	**0.785**	**0.217**	**0.206**
3	7	32	0.758	0.259	0.177
4	7	64	0.758	0.256	0.191
5	9	4	0.681	0.329	0.283
6	9	8	0.662	0.357	0.266
7	9	16	0.733	0.239	0.371
8	9	32	0.751	0.217	0.371
9	12	4	0.679	0.347	0.226
10	12	8	**0.767**	**0.216**	**0.297**
11	12	16	0.653	0.38	0.22
12	12	32	0.642	0.392	0.229
13	10	4	0.744	0.229	0.357
14	10	8	0.71	0.298	0.26
15	10	16	0.725	0.268	0.303
16	10	32	0.657	0.368	0.249

Table 4: Accuracy, APCER and BPCER for different Neural Network configurations and pre-processing functions.

Table 4 shows different architectural configurations and their respective performance. HLN denotes the number of neurons in the 2 hidden dense layers. PP denotes the ID of the pre-processing method used. These correspond to the IDs in Table 3. The best differential and non-reference performance are outlined in bold.

Figure 10 and 11 show the DET-Curves for the different regions for DFF experiment 2 and 10 respectively. The numbers after the region prefix denote the AUC. Furthermore, Figure 12 and 13 show the training curves for these experiments for the training and validation set.

6. DISCUSSION

6.1 Kullback Leibler Divergence

Table 2 clearly shows that the initial assumption of morphed pairs having a higher KBD value is correct. The mean of the Bona-Fide pairs are consistently lower than those of the Morphed pairs. Furthermore, the sample variance is quite low which suggests that the values are not very far spread apart.
This initial table suggests that the Kullback Leibler Divergence might be a good indication of morphs. But the plot of the DET-Curve shows that this is not true. The accuracy is far too bad to be used for biometric tasks. Also note that the KBD is incapable of differentiating between bona-fide and morphed pairs in the forehead region. This is in line with what is expected, because the morphing pipeline

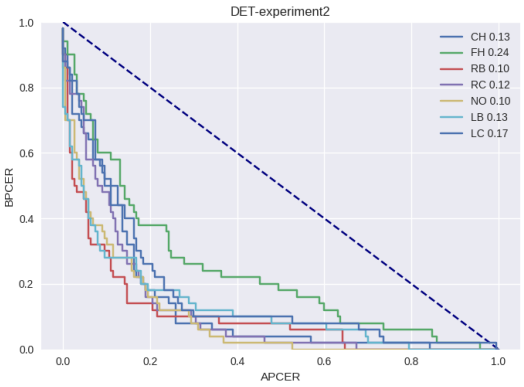


Figure 10: DET-Curves of different regions for differential L2-Normalized approach (DFF)

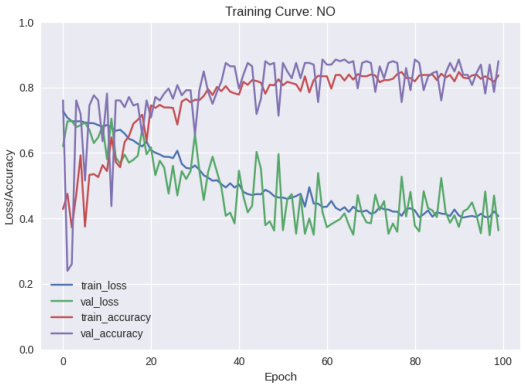


Figure 12: Training Curve of Nose region for differential, L2-Normalized approach (DFF)

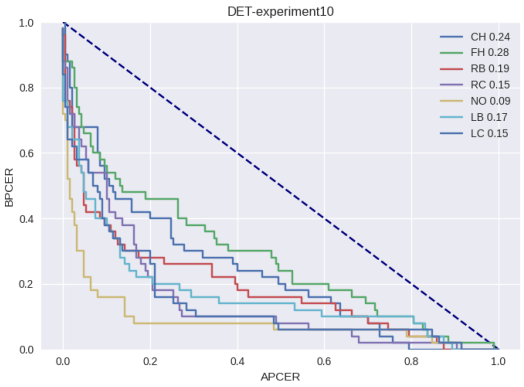


Figure 11: DET-Curves of different regions for non-reference, logarithmic, L2-Normalized approach (DFF)

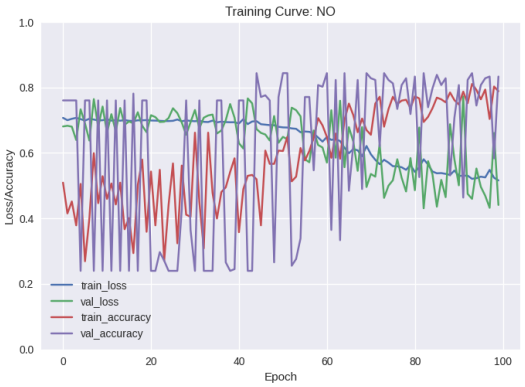


Figure 13: Training Curve of Nose region for non-reference, logarithmic approach (DFF)

splices the morphs. Despite the initial assumption being correct, the KBD is not a good metric for determining morphed face images.

6.2 SVM

The performance of the SVM is quite dependent on the pre-processing steps used. Table 3 shows a bad APCER and BPCER in the first four rows, which all utilize the radius weighted mean. The outlined results indicate the best performance for the differential and non-reference approaches. Generally, L2 Normalizing the input seemed to result in better performance. Surprisingly, the non-reference experiments perform better than the differential ones. In contrast to the differential method, the non-reference method performs better on logarithmic data. In the differential approach the SVM has not been able to improve its performance by comparing the reference image with the questioned image. Instead it seems that the additional data confuses the SVM more than it assists in the classification of morphs.
In Figure 8 can be see that unlike the KBD approach the SVM has been successful to some extend at the classification of morphs for the forehead region. The region does perform significantly worse than the other regions. It's surprising that the SVM has been able use the forehead region to differentiate between bona-fide and morphed pairs in both the differential and non-reference approach. This could be because the forehead region is partly inside spliced face.

6.3 DFF

Table 4 shows that the number of neurons does not have a significant effect on the performance of the DFF. The variations are most likely a result of random chance than the variations of the number of neurons. The table also shows that the differential method outperforms the non-reference methods by a very small margin as the APCER and BPCER is slightly lower. The DFF has been more effective at utilizing the reference image. Figures 10, 11 show the DET-Curve of the various regions. Across the experiments the DET-Curves of the non-reference method are more spread out than the curves of the differential method. Suggesting that the differential-method is more stable. The training curves of the experiments shown in Figure 12 and 13 support this claim. The exact cause of this is unclear. It could be that the larger input size of the differential approach stabilizes the results. There is no strong evidence that the DFF has been able to compare the reference image with the questioned image. The difference in performance is almost negligible.
Also, in similar fashion to the SVM the DFF is able to differentiate between bona-fide and morphed pairs in the forehead region. With limited effectiveness however. In all experiments the nose regions is one of the most effective regions, but only by a very small margin. This has not to do with the fact that the nose region is in the center of the image. The chin consistently is one of the less effective regions. This probably is due to the larger chance of facial hair being present on the chin.

6.4 SVM vs. DFF

The SVM and DFF approach both have about equal best performance rates. A surprising observation however has been that the DFF performs better when supplied with the reference image. On the contrary, the SVM performs better with a non-reference approach.

6.5 Regional analysis

There were no large differences in performance between the regions. This suggests that the classifiers were unable to pick up on unique local regional details, which would be affected by morphing. More research is needed to investigate whether regional analysis can contribute to morphing detection.

6.6 Comparison State of the Art

It is hard to directly compare this study to other studies. *Neubert et al.*[12] were able to detect morphs with 75.2% using a frequency approach. But the data they used was significantly different from the data used in this study. They had about ten times the amount of images, which would improve classification performance. On the contrary, their images were compressed to resemble the passport standard more closely. This in turn would worsen the performance. Furthermore, they did not include training datasets into the test datasets, further reducing performance. It is hard to compare effectiveness of different morphing detection algorithms without using identical data and evaluation methods. Future work should also aim at creating a large diverse database of morphed and bona-fide faces.

7. CONCLUSION

In this research was investigated how the grayscale frequency content of images can be used to detect morphs using specific regions of the face. The Kullback Leibler Divergence was used in a differential only morphing detection approach. Also, both a Support Vector Machine and Neural Network were trained on the frequency spectrum of 32x32 regions in a differential and non-reference setting.

The mean Kullback Leibler Divergence was higher for morphed pairs than bona-fide pairs. However, the variations were too high to use this observation in order to detect morphs.

The SVM and DFF both were able to detect morphs with a best performance of 79.9% and 78.5% accuracy respectively. The SVM performed better in a non-reference setting, while the DFF performed better in a differential setting.

There were no large differences in the performance of different regions. The nose performed only slightly better than the other regions with a very small margin.

This study is fairly limited in scope. The performance might be drastically different if the classifier is trained and tested on two separate datasets. More research is needed to investigate how the frequency domain can contribute to detecting morphs. Future work could look at the directional data of the frequency content, which was discarded in this study. Furthermore, different colour channels could be analyzed instead of just the grayscale values. Frequencies in the skin could for example, be represented more in specific colour channels. Also, different approaches of regional analysis might reveal a methods, which are capable of picking up on unique regional features. If any multiple unique features are found in facial regions these can possibly be combined to increase the performance of morphing detection.

8. REFERENCES

[1] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[2] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.

[3] L. R. Carlos-Roca, I. H. Torres, and C. F. Tena. Facial recognition application for border control. In *Proceedings of the International Joint Conference on Neural Networks*, volume 2018-July. Institute of Electrical and Electronics Engineers Inc., oct 2018.

[4] L. Debiasi, U. Scherhag, C. Rathgeb, A. Uhl, and C. Busch. PRNU-based detection of morphed face images. In *IWBF 2018 - Proceedings: 2018 6th International Workshop on Biometrics and Forensics*, pages 1–7. Institute of Electrical and Electronics Engineers Inc., jun 2018.

[5] M. Ferrara, A. Franco, and D. Maltoni. The magic passport. In *IJCB 2014 - 2014 IEEE/IAPR International Joint Conference on Biometrics*. Institute of Electrical and Electronics Engineers Inc., dec 2014.

[6] D. E. King. Dlib-ml: A Machine Learning Toolkit. Technical report, 2009.

[7] D. P. Kingma and J. L. Ba. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*. International Conference on Learning Representations, ICLR, dec 2015.

[8] C. Kraetzer, A. Makrushin, T. Neubert, M. Hildebrandt, and J. Dittmann. Modeling attacks on photo-ID documents and applying media forensics for the detection of facial morphing. In *IH and MMSec 2017 - Proceedings of the 2017 ACM Workshop on Information Hiding and Multimedia Security*, pages 21–32, New York, New York, USA, jun 2017. Association for Computing Machinery, Inc.

[9] J. Liao, R. S. Lima, D. Nehab, H. Hoppe, P. V. Sander, and J. Yu. Automating image morphing using structural similarity on a halfway domain. *ACM Transactions on Graphics*, 33(5):1–12, aug 2014.

[10] A. Martin, A. Martin, G. Doddington, T. Kamm, M. Ordowski, and M. Przybocki. The DET Curve in Assessment of Detection Task Performance. 4:1895 – 1898, 1997.

[11] S. Milborrow and F. Nicolls. Locating Facial Features with an Extended Active Shape Model. Technical report.

[12] T. Neubert, C. Kraetzer, and J. Dittmann. A face morphing detection concept with a frequency and a spatial domain feature space for images on eMRTD. In *f*, pages 95–100, New York, New York, USA, jul

2019. Association for Computing Machinery, Inc.

[13] T. E. Oliphant. *A guide to NumPy*, volume 1. Trelgol Publishing USA, 2006.

[14] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[15] P. Pérez, M. Gangnet, and A. Blake. Poisson image editing. In *ACM SIGGRAPH 2003 Papers, SIGGRAPH '03*, pages 313–318, New York, New York, USA, 2003. ACM Press.

[16] R. Raghavendra, K. B. Raja, and C. Busch. Detecting morphed face images. In *2016 IEEE 8th International Conference on Biometrics Theory, Applications and Systems, BTAS 2016*. Institute of Electrical and Electronics Engineers Inc., dec 2016.

[17] R. Raghavendra, K. B. Raja, S. Venkatesh, and C. Busch. Transferable Deep-CNN Features for Detecting Digital and Print-Scanned Morphed Face Images. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 2017-July:1822–1830, 2017.

[18] R. Ramachandra, S. Venkatesh, K. Raja, and C. Busch. Towards making Morphing Attack Detection robust using hybrid Scale-Space Colour Texture Features. In *ISBA 2019 - 5th IEEE International Conference on Identity, Security and Behavior Analysis*. Institute of Electrical and Electronics Engineers Inc., jan 2019.

[19] D. J. Robertson, A. Mungall, D. G. Watson, K. A. Wade, S. J. Nightingale, and S. Butler. Detecting morphed passport photos: a training and individual differences approach. *Cognitive Research: Principles and Implications*, 3(1):27, dec 2018.

[20] U. Scherhag, C. Rathgeb, and C. Busch. Towards detection of morphed face images in electronic travel documents. In *Proceedings - 13th IAPR International Workshop on Document Analysis Systems, DAS 2018*, pages 187–192. Institute of Electrical and Electronics Engineers Inc., jun 2018.

[21] U. Scherhag, C. Rathgeb, J. Merkle, R. Breithaupt, and C. Busch. Face Recognition Systems under Morphing Attacks: A Survey. *IEEE Access*, 7:23012–23026, 2019.

[22] L. Spreeuwers, M. Schils, and R. Veldhuis. Towards robust evaluation of face morphing detection. In *European Signal Processing Conference*, volume 2018-Septe, pages 1027–1031. European Signal Processing Conference, EUSIPCO, nov 2018.