h_da
HOCHSCHULE DARMSTADT
UNIVERSITY OF APPLIED SCIENCES

Hochschule Darmstadt

– Fachbereich Informatik –

Face Morphing Attack Detection (MAD)

Wissenschaftliche Arbeit zum Modul “Biometric Systems”

vorgelegt von

Patrick Eidemüller

Matrikelnummer: 768465

Referent	:	Prof. Dr. Christoph Busch
Korreferent	:	Dr. Lazaro Janier Gonzalez-Soler

ERKLÄRUNG

Ich versichere hiermit, dass ich die vorliegende Arbeit selbstständig verfasst und keine anderen als die im Literaturverzeichnis angegebenen Quellen benutzt haben.

Alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten oder noch nicht veröffentlichten Quellen entnommen sind, sind als solche kenntlich gemacht.

Die Zeichnungen oder Abbildungen in dieser Arbeit sind von mir selbst erstellt worden oder mit einem entsprechenden Quellennachweis versehen.

Darmstadt, 25.04.2023

Patrick Eidemüller

ABSTRACT

Biometrics verification Systems are gaining increasing importance in modern society due to the improved performance of recognition systems and the convenience they offer in everyday life. Facial recognition-based unlocking of smartphones and the application of biometric systems in areas like automated border controls have become indispensable. However, despite their advantages, biometric systems remain vulnerable, especially in facial recognition, where various attack vectors such as printed faces or silicone masks can be exploited. Alongside well-known presentation attacks, the emerging concern of morphing attacks poses a significant threat in today's context [1]. The field of morphing attacks is relatively new and has received limited research attention thus far. The objective of this paper is to detect single morphing attacks by implementing a Convolutional Neural Network in Python. The proposed method will be bench marked against ISO standards using the HDA Morphing Database, which contains both morphed and bonafide images. Additionally, a comparison will be made with existing literature results. The paper aims to demonstrate the complexity of distinguishing between bonafide and morphed faces and will provide insights on improving model accuracy.

INHALTSVERZEICHNIS

1 INTRODUCTION

In the Field of biometric security, the automated recognition of individuals based on their biological characteristics, face morphing attacks have emerged as a significant concern. Due to the replacement of paper documents through electronic Machine Readable Travel Documents (eMRTD) storing biometric features for machine-assisted identify verification, such as Automatic Border Controls (ABC). In an ABC system, the verification of the association between the electronic Machine Readable Travel Document (eMRTD) and the passport holder, who presents the eMRTD to the border guard, is automatically performed by comparing the live captured face image with the facial reference image stored in the eMRTD passport. This integration has greatly amplified the advantages offered by ABC systems, resulting in highly dependable and accurate border control procedures. [4] Face morphing attacks involve manipulating facial images to create a hybrid image that can delude facial recognition systems. The consequences of such attacks are extensive, as they can compromise the integrity and reliability of biometric security measures. Detecting and disclose face morphing attacks is crucial to maintain the security and reliability of these systems. In the following subsection, we will introduce face morphing and explore the methods and techniques for detecting and identifying face morphing attacks, aiming to develop robust solutions that can effectively counter this emerging threat.

1.1 FACE MORPHING

In recent years, face morphing has emerged as a well known technique in the field of image manipulation. It involves merging two or more facial images to create a seamless and visually confusingly similar transition between them. The resulting image demonstrates characteristics of both original faces, making it difficult to detect the manipulation with the human eye.

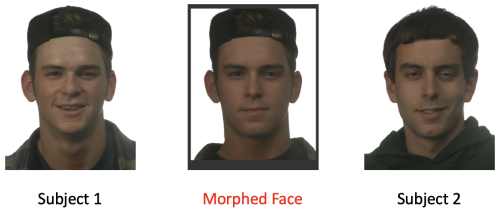

Abbildung 1.1: Illustration of face morphing

Figure 1 provides a visual representation of face morphing. Presenting two distinct faces and the resulting morphed image. The morphed image

smoothly combines the facial features of the two individuals, resulting in a face which is indistinguishable to the human eye with the appearance and similarity of both input faces. Face morphing techniques rely on advanced image processing algorithms, including geometric and texture-based methods, to smoothly merge the facial attributes of different individuals. Geometric methods align the facial landmarks, such as eyes, nose, and mouth, between the source images to ensure accurate blending. Texture-based methods focus on smoothly blending the skin textures and color tones to create a realistic transition.

1.2 FACE MORPHING ATTACK DETECTION MAD

Face morphing attack detection (MAD) is a challenging task. MAD procedures can be separated in two cases as illustrated in the following figures.

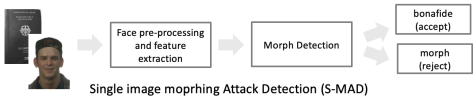

Abbildung 1.2: Illustration Single Image Morph Attack Detection (S-MAD)

Figure 1.2 illustrates the concept of single morph attack detection (S-MAD). This approach examines a single face image without any reference pictures and classifies whether the face is morphed or bona fide. Detecting such attacks requires advanced algorithms and techniques that can effectively differentiate between bona fide and morphed faces.

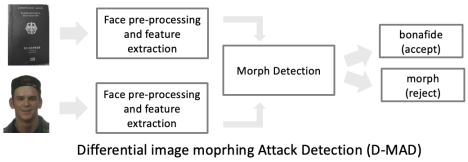

Abbildung 1.3: Illustration of Differential morphing attack detection D-MAD

Figure 1.3 describes the concept of differential moprh attack detection (D-MAD) where the potentially morphed image is compared with a trusted probe image.

In this work we will focus on Single Morph Attack Detection.

2

CNN

A Convolutional Neural Network (CNN) is a specialized type of artificial neural network. Traditional non-neural networks face limitations in image recognition tasks due to the requirement of having the number of input layer nodes equal to the pixel count of the image. This becomes computationally intensive as images have a high number of pixels. The main distinction of a CNN lies in its composition of one or more convolutional layers, which extract local features from input data using convolutional kernels. These convolutional layers are complemented by activation functions to model non-linear relationships. Consequently, the network can automatically learn relevant features and abstract them hierarchically. To reduce dimensionality and make the learned features invariant to small translations, the output of the convolutional layers is further reduced using pooling layers. During the training process, the CNN is adjusted by iteratively optimizing its weights. CNNs are commonly employed in image classification, object detection, and speech processing tasks. In this paper, we aim to investigate the suitability of a CNN for detecting face morphing.

2.1 THE ARCHITECTURE OF VGG16

VGG-16 is a convolutional architecture developed by Karen Simonyan and Andrew Zisserman of the Visual Geometry Group Lab at the University of Oxford in 2014. The architecture gained recognition for its exceptional performance in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC). The VGG16 model requires RGB images as input, with a size of 224x224 pixels. It consists of a total of 13 convolutional layers and three fully connected layers. Compared to other architectures, VGG16 uses smaller filter kernels with a size of 3x3 pixels.

The first two layers are convolutional layers, each with 64 3x3 filter kernels and a stride of one. This results in 64 feature maps with a size of 224x224 pixels. The filter kernels extract features from the input images, detecting characteristics such as edges, texture details, or color information.

After each convolutional layer, a ReLU activation is applied, removing non-negative values and enhancing the network's representational capacity. This helps capture non-linear features in the data.

The first pooling layer uses the max-pooling technique with a 2x2 kernel and a stride of two, reducing the image size to 112x112 pixels. Pooling layers, in general, aim to reduce the dimensionality of the feature maps, leading to computational efficiency. In max-pooling, the maximum activation value within a certain range is selected, introducing spatial invariance to small shifts and reducing the size of the activation maps.

Following the pooling layer, two more convolutional layers with 128 kernels are applied, increasing the number of feature maps to 128. The subsequent max-pooling layer further reduces the size to 56x56 pixels.
Two additional convolutional layers with 256 kernels are then applied, followed by a pooling layer that reduces the size to 28x28 pixels.
Next, we have three sets of two consecutive convolutional layers, each separated by a max-pooling layer. After the last pooling layer, we reach a size of 7x7 pixels with 512 feature maps, resulting in 4096 input pixels for the network.
Finally, the abstracted features of the images in the VGG16 model are processed in the fully connected layers to perform classification. These layers have a high number of neurons connected to the features. In the first two fully connected layers of the VGG16 model, ReLU activations and dropout regularization are applied to prevent overfitting. The last layer uses the sigmoid activation function to generate a probability for binary classification.

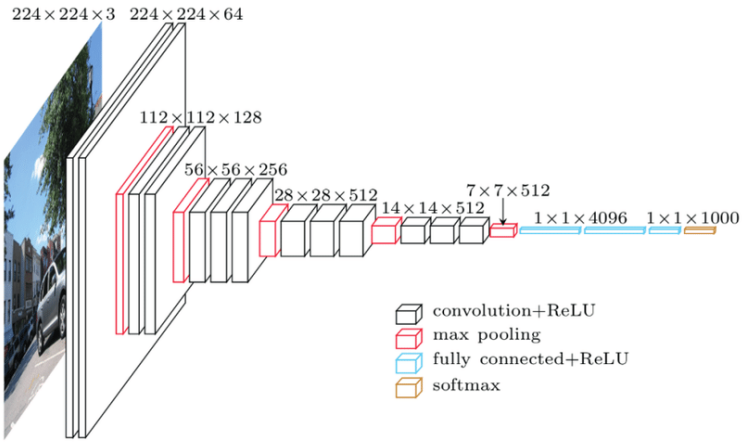

Abbildung 2.1: Architecture of the VGG CNN [5]

2.2 USING TRANSFER LEARNING AND VGG16 FOR FACE MORPHING DETECTION

In the context of transfer learning using the VGG16 model, the pre-trained model trained on a large dataset (e.g., ImageNet) is used as a starting point to solve a new task, such as the Face Morphing problem. Instead of training the model from scratch, the weights and features of the pre-trained model are retained, and only the last layers of the network are adjusted or replaced.
In the case of the Face Morphing problem, the last fully connected layer of the VGG16 model is removed and replaced with new layers specifically trained for the classification of genuine and morphed faces. The pre-trained weights of the earlier layers of the VGG16 model are preserved.
Transfer learning allows the model to benefit from the learned features and patterns contained in the pre-trained model. The earlier layers of the VGG16

model have learned to recognize general visual features that are also relevant for face classification, such as edges, textures, or basic facial shapes. By fine-tuning the last layers on the specific dataset of the Face Morphing problem, the model can learn to recognize and classify the specific differences between genuine and morphed faces.
Transfer learning enables achieving good performance even with limited data, as the model already has a good understanding of the relevant features. It significantly reduces training time and resource requirements since it is not necessary to train the entire model from scratch but only adjust the last layers.
The VGG16 model is well-suited for extracting features in faces and solving complex classification tasks due to its deep architecture and ability to capture complex features. This includes distinguishing between genuine and morphed faces in the case of the Face Morphing problem.

3

IMPLEMENT A CNN FOR SINGLE MORPH DETECTION

3.1 DATABASE

The provided Datasets namely the Facial Recognition Technology (FERET) and Facial Recognition Grand Challenge (FRGC) are used for facial recognition system evaluation. The FERET Database consist of 1199 individuals and 365 duplicate sets of images. [2] The FRGC Database contains more images with more complexity [3]. The Images where manipulated with 4 kinds of morphing techniques: facefusion, facemoprher, opencv and ubo.

3.2 DATA PRE-PROCESSING

Pre-processing face images is an essential step in the process of morphing detection with CNNs. These steps making it easier for the network to learn meaningful features, reduce noise, standardize the data, and enhance generalization. These pre-processing steps facilitate the training process, improve the network's performance, and enable the CNN to achieve better accuracy and robustness in its predictions.

- Face Detection: First, face detection algorithms are employed to locate and extract the facial region from the input image. This step helps to isolate the face and remove irrelevant background information. In this research we used 3 different approaches to crop the faces, first the Haar Cascade Classifier which is very accurate but slowly and second the standard pytorch method center crop, which is significantly faster but not that accurate. Third we used the MTCNN to extract the faces from the images which characteristics are very similar to the Haar Cascade Classifier.

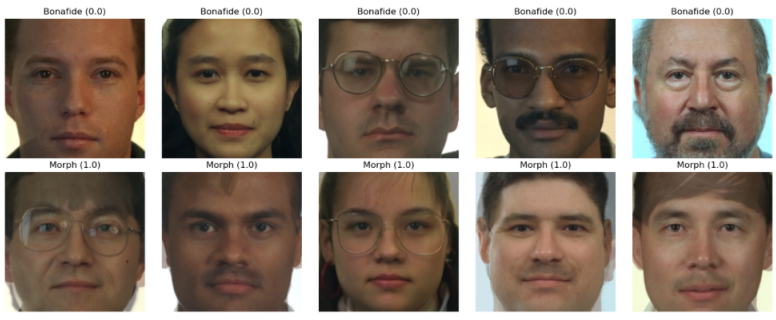

Abbildung 3.1: Example of Face Detection with Cascade

- Resize: It is important that all images have the equivalent size therefore it is important to resize them. Further more the VGG16 Network was

trained on images with the size of 224x224 Pixels which we adopted for training the network.

- Data Normalization: pre-processing often involves normalizing the input data to a standardized range. By scaling the pixel values of images to a common scale, such as [0, 1] or [-1, 1], the computations within the CNN become faster and more efficient. The normalization step eliminates the need for the network to adapt to varying input data scales, thereby accelerating the training process. In our case, Data Normalization did not lead to an improvement of the network.

- Data Augmentation: pre-processing techniques like data augmentation generate additional training samples by applying random transformations to the original data. These transformations, such as rotations, translations, or flips, create variations of the input images without the need for additional data collection. Data augmentation effectively increases the training data set size, allowing the CNN to learn from a more diverse set of examples. This augmentation aids in faster convergence and improved generalization of the network. In our case, data augmentation did not lead to an improvement of the network.

- Batch Processing: pre-processing enables the CNN to efficiently process data in batches rather than individually. Batching involves grouping multiple input samples together and processing them simultaneously. This technique takes advantage of parallel computing architectures, such as GPUs, to perform computations on multiple samples simultaneously. By pre-processing the data into batches, the CNN can take advantage of parallelism and expedite the training process.

In the following example the pictures where pre processed with different kind of techniques such as random rotations, flips, random color variations, changing lighting conditions, normalization and center crop. There was no significant effect on the accuracy of the network by using these pre processed pictures.

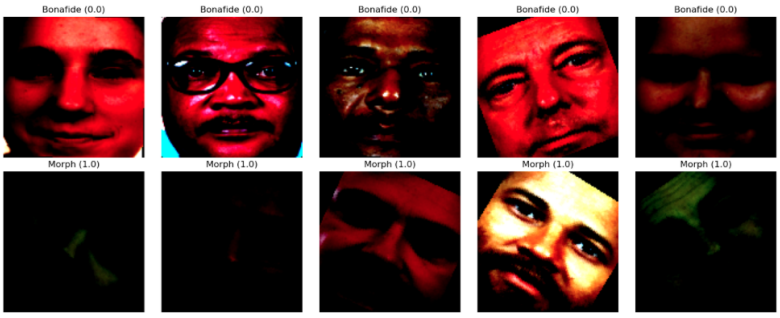

Abbildung 3.2: Example of Pre-processed Images with Center Cropcentercrop

3.3 HYPERPARAMETER TUNING FOR CNN

Hyperparameter tuning is a crucial step in optimizing the performance of a Convolutional Neural Network (CNN). Hyper parameters cannot directly be learned from the training data and have to be set before the training starts, sometimes it is difficult to find the best settings. They determine the architecture, learning rate, regularization techniques, and other settings of the CNN. The importance of hyper parameter tuning lies in the fact that choosing the appropriate values for these hyper parameters can significantly impact the network's performance. It can make the difference between a CNN that struggles to converge or overfits the data and one that achieves high accuracy and generalizes well. Finding the right hyper parameters for a CNN involves an iterative process of experimentation and evaluation. Here are some approaches which were used to train the CNN, which is introduced in the next section.

- Manual Tuning: Manually selecting hyper parameters based on prior knowledge and experience. This approach is often time-consuming and requires expertise, but it can be effective for simple models or small data sets.
- Grid Search: Specifying a grid of hyper parameter values to be explored exhaustively. It involves training and evaluating the CNN for all possible combinations of hyper parameters within the defined grid. While this approach guarantees finding the best hyper parameters within the search space, it can be computationally expensive for large parameter grids.
- Random Search: Randomly sampling hyper parameters from a defined search space. It offers a more efficient alternative to grid search by exploring different combinations of hyper parameters without exhaustively searching the entire space. Random search allows for a better exploration of the hyper parameter space and can often find good solutions faster than grid search. During the experiments, it turned out that it is almost equally time-consuming as grid search.

3.4 THE ARCHITECTURE OF THE PROPOSED CNN MODEL

In section 2.1, the architecture of the VGG16 CNN has already been explained in detail. Now, we will focus on the fine-tuning of the last layers of the network.

The choice of fine-tuning the last layers of the VGG16 network architecture for face morphing detection is based on several considerations.

- Adaptive Average Pooling: The AdaptiveAvgPool2d layer is used to convert the output feature maps from the previous layers into a fixed-size representation. In this case, it converts the output shape of [-1, 512, 7, 7] to [-1, 512]. By applying adaptive pooling, we ensure that the

```
----------------------------------------------------------------
        Layer (type)               Output Shape         Param #
================================================================
            Conv2d-1         [-1, 64, 224, 224]           1,792
              ReLU-2         [-1, 64, 224, 224]               0
            Conv2d-3         [-1, 64, 224, 224]          36,928
              ReLU-4         [-1, 64, 224, 224]               0
         MaxPool2d-5         [-1, 64, 112, 112]               0
            Conv2d-6        [-1, 128, 112, 112]          73,856
              ReLU-7        [-1, 128, 112, 112]               0
            Conv2d-8        [-1, 128, 112, 112]         147,584
              ReLU-9        [-1, 128, 112, 112]               0
        MaxPool2d-10          [-1, 128, 56, 56]               0
           Conv2d-11          [-1, 256, 56, 56]         295,168
             ReLU-12          [-1, 256, 56, 56]               0
           Conv2d-13          [-1, 256, 56, 56]         590,080
             ReLU-14          [-1, 256, 56, 56]               0
           Conv2d-15          [-1, 256, 56, 56]         590,080
             ReLU-16          [-1, 256, 56, 56]               0
        MaxPool2d-17          [-1, 256, 28, 28]               0
           Conv2d-18          [-1, 512, 28, 28]       1,180,160
             ReLU-19          [-1, 512, 28, 28]               0
           Conv2d-20          [-1, 512, 28, 28]       2,359,808
             ReLU-21          [-1, 512, 28, 28]               0
           Conv2d-22          [-1, 512, 28, 28]       2,359,808
             ReLU-23          [-1, 512, 28, 28]               0
        MaxPool2d-24          [-1, 512, 14, 14]               0
           Conv2d-25          [-1, 512, 14, 14]       2,359,808
             ReLU-26          [-1, 512, 14, 14]               0
           Conv2d-27          [-1, 512, 14, 14]       2,359,808
             ReLU-28          [-1, 512, 14, 14]               0
           Conv2d-29          [-1, 512, 14, 14]       2,359,808
             ReLU-30          [-1, 512, 14, 14]               0
        MaxPool2d-31            [-1, 512, 7, 7]               0
AdaptiveAvgPool2d-32            [-1, 512, 7, 7]               0
           Linear-33                 [-1, 4096]     102,764,544
             ReLU-34                 [-1, 4096]               0
          Dropout-35                 [-1, 4096]               0
           Linear-36                 [-1, 4096]      16,781,312
             ReLU-37                 [-1, 4096]               0
          Dropout-38                 [-1, 4096]               0
           Linear-39                    [-1, 1]           4,097
================================================================
Total params: 134,264,641
Trainable params: 134,264,641
Non-trainable params: 0
----------------------------------------------------------------
Input size (MB): 0.57
Forward/backward pass size (MB): 218.77
Params size (MB): 512.18
Estimated Total Size (MB): 731.53
----------------------------------------------------------------
```

Abbildung 3.3: Transfer Learning with VGG16 Modell Architecture

network can handle input images of different sizes while preserving the spatial information. This is important for face morphing detection, as the network needs to capture relevant features irrespective of the size or aspect ratio of the input images.

- Fully Connected Layers: The Linear layers are responsible for learning the complex mapping between the extracted features and the final output. Two linear layers are added: one with 4096 units and ReLU activation, and another with a single unit and sigmoid activation.
  - The first linear layer serves as a bottleneck layer that reduces the dimensionality of the feature representation. This helps in lear-

ning more compact and meaningful representations that are relevant for face morphing detection.

  - The second linear layer (single unit with sigmoid activation) is responsible for binary classification, indicating whether an image is a genuine face or a morphed face. The sigmoid activation function maps the output to a range between 0 and 1, where values closer to 1 indicate a higher probability of being a morphed face.

- Dropout: Dropout is applied after each linear layer to mitigate overfitting. It randomly sets a fraction of the input elements to zero during training, which helps in preventing the network from relying too heavily on specific features and promotes better generalization
  By combining adaptive pooling, fully connected layers with ReLU activation, dropout regularization, and sigmoid activation, the fine-tuned VGG16 network is able to capture and learn discriminative features that are effective for distinguishing between genuine and morphed faces.

4

RESULTS AND EVALUATION

4.1 OVERVIEW OVER EXPERIMENTS

Multiple experiments were conducted for the project, involving the implementation of the project using both TensorFlow and Keras initially, and subsequently incorporating PyTorch due to challenges encountered in prediction. Two variants of the CNN, namely a self-trained model and a pre-trained VGG16 model with fine-tuning the last layers, were evaluated in each experiment. Notably, the pre-trained model consistently outperformed the self-trained model across all experiments. Despite exhaustive grid searches to determine optimal hyper parameters, a carefully balanced data set, and extensive pre-processing techniques, the final model exhibited favorable training and validation accuracy but indicated deficient performance on the test data set. Specifically, the model demonstrated a tendency to classify new images predominantly as bona fide samples, as remonstrated in Figure 4.1 and 4.2

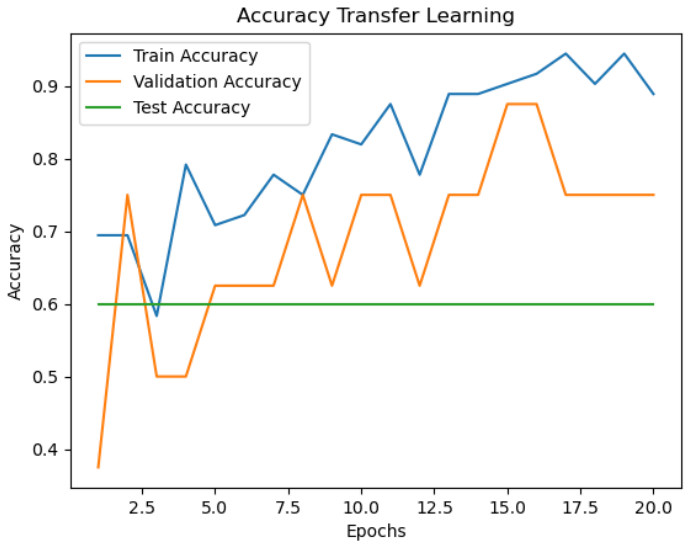

Abbildung 4.1: Training, Validation and Testing Accuracy

The CNN model exhibits a significant limitation as it predominantly classifies all input images as bona fide, failing to effectively discriminate morphed samples. Consequently, this leads to an unbalanced distribution of outcomes, with an over representation of false negatives.

In a scientific discourse, this phenomenon highlights a fundamental deficiency in the model's ability to accurately identify and differentiate between genuine and morphed images. The CNN's classification performance de-

monstrates a clear bias towards the bona fide class, rendering it inadequate for robust morph detection. The prevalence of false negatives indicates the model's failure to capture and discern the intricate visual cues and patterns associated with morphed facial features.

This inherent limitation demands further investigation and refinement of the CNN architecture, feature extraction techniques, and training methodologies to enhance its discriminatory capabilities and enable effective distinction between bona fide and morphed facial images.

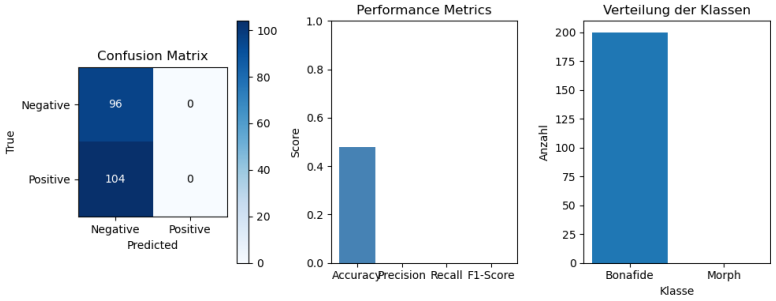

Abbildung 4.2: Confusion Matrix, Performance Metrics and Class Distribution

4.2 EVALUATION IN ACCORDING TO ISO/IEC IS 30107-1

ISO/IEC IS 30107-1 is a standard that provides guidelines for the evaluation of biometric presentation attack detection (PAD) methods. It defines the evaluation methodology and metrics to assess the performance of PAD techniques in the context of biometric systems.

- False Acceptance Rate (FAR): The rate at which the system incorrectly accepts an morph (sample as a bona fide sample.
- False Rejection Rate (FRR): The rate at which the system incorrectly rejects a bona fide sample as an morph sample.
- Receiver Operating Characteristic (ROC) Curve: A graphical representation of the trade-off between the False Positive Rate (FPR) and the True Positive Rate (TPR) at different classification thresholds or operating points of a binary classifier.
- Equal Error Rate (EER): The point on the ROC curve where FAR is equal to FRR. It represents the equal trade-off between the two error rates
- APCER (Attack Presentation Classification Error Rate): is a biometric performance metric that quantifies the proportion of attack presentations that are incorrectly classified as normal presentations at the component level in a specific scenario. It measures the error rate in distinguishing between genuine biometric presentations and spoof or fraudulent presentations.

- NPCER (Normal Presentation Classification Error Rate): is a biometric performance metric that quantifies the proportion of normal presentations that are incorrectly classified as attack presentations at the component level in a specific scenario. It measures the error rate in distinguishing between genuine biometric presentations and misclassifying them as spoof or fraudulent presentations.
- DET (Detection Error Tradeoff): The DET curve plots the False Acceptance Rate (FAR) against the False Rejection Rate (FRR) for different operating points or threshold settings of a biometric system.

4.3 RESULTS AND EVALUATION

An AUC of 0.32 indicates poor performance, as it suggests that the system's ability to differentiate between bona fide and morph samples is no better than random chance. In this case, the CNN's classification results are not reliable, as it consistently assigns all samples as bona fide, regardless of their true nature.

Due to the poor classification performance of the biometric system, the evaluation plots lack meaningful interpretation and scientific validity.

The evaluation plots, such as the Receiver Operating Characteristic (ROC) curve or the Detection Error Tradeoff (DET) curve, are commonly used to assess the performance of a biometric system. These plots provide insights into the system's ability to discriminate between bona fide and morph samples.

However, when the classification performance of the system is inadequate, the evaluation plots become unreliable and their interpretation becomes challenging. In such cases, the plots may exhibit misleading patterns, making it difficult to draw meaningful conclusions from them.

It is crucial to address the underlying issues causing the poor classification performance before relying on the evaluation plots. These issues could be caused from various factors, such as insufficient training data, inappropriate feature extraction methods, or suboptimal choice of model architecture.

In scientific research, it is essential to critically assess the performance of a biometric system and ensure the reliability of the evaluation results. Therefore, when the classification performance is unsatisfactory, it is important to investigate and address the root causes, improve the system's design and implementation, and re-evaluate its performance using appropriate metrics and evaluation techniques.

Only when the classification performance reaches a satisfactory level can the evaluation plots regain their significance and provide meaningful insights into the system's ability to differentiate between genuine and impostor samples.

This situation may occur due to various reasons, such as inadequate training data, insufficient model complexity, or inappropriate choice of hyper parameters. It indicates that the CNN has not learned the necessary discriminatory features to distinguish morphed samples from genuine ones.

To improve the performance of the CNN, it would be necessary to reassess and modify different aspects of the training process. This could involve collecting more diverse and representative training data, adjusting the network architecture or hyper parameters, implementing appropriate data augmentation techniques, or applying more advanced training algorithms. By iteratively refining these aspects and evaluating the model's performance, it may be possible to improve its ability to correctly classify both bona fide and morphed samples.

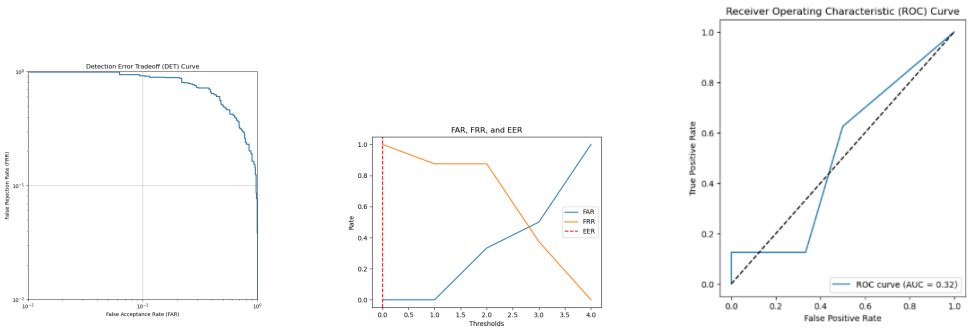

Abbildung 4.3: DET, FAR, FRR and EER and ROC Plot

APCER (Attack Presentation Classification Error Rate) and NPCER (Normal Presentation Classification Error Rate) are metrics used in biometric systems to evaluate the performance of presentation attack detection algorithms.

APCER represents the proportion of attack presentations (fake or morph attempts) that are incorrectly classified as normal presentations (bona fide attempts) at the component level in a specific scenario. It indicates the system's vulnerability to false acceptance of presentation attacks.

NPCER, on the other hand, represents the proportion of normal presentations (bonafide attempts) that are incorrectly classified as attack presentations (false rejection) at the component level in a specific scenario. It reflects the system's vulnerability to false rejection of bona fide attempts.

A histogram can visualize the APCER and NPCER by representing the distribution of error rates for different scenarios or classification models. The x-axis of the histogram represents the error rates, while the y-axis represents the frequency or count of scenarios or models falling within each error rate interval or category.

The histogram will have separate bars or bins for APCER and NPCER. The height of each bar represents the frequency or count of scenarios or models that exhibit the corresponding error rate.

In summary, the histogram provides a visual representation of the distribution of error rates for APCER and NPCER, allowing to assess the system's performance in terms of false acceptance and false rejection of presentation attacks and bona fide attempts, respectively. It helps in understanding the system's vulnerability to different types of errors and provides insights into the effectiveness of presentation attack detection algorithms.

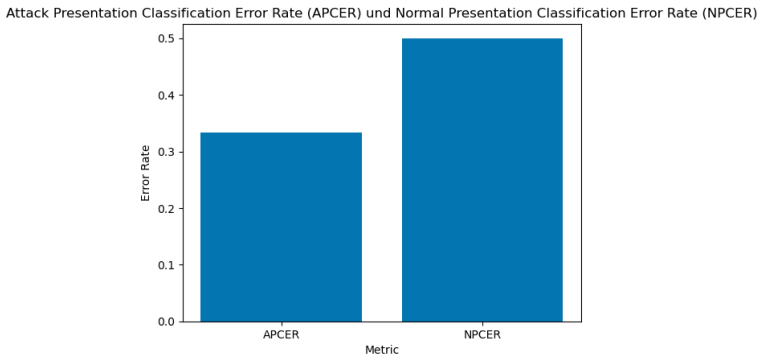

Abbildung 4.4: APCER vs. NPCER

4.4 FURTHER WORK

The persistent issue of misclassifying all samples as bona fide in the CNN's predictions is a significant challenge that needs to be addressed in order to improve the performance of the biometric system.

Misclassification occurs when the CNN fails to distinguish between bona fide samples and morphed samples, categorizing all inputs as bona fide. This behavior severely limits the system's ability to detect and differentiate between legitimate and fraudulent identities, undermining its overall effectiveness in biometric authentication tasks.

To overcome this problem, several strategies can be considered for future improvements. Firstly, it is crucial to revisit the training process and data set. Ensuring a more comprehensive and diverse data set that includes an adequate representation of morphed samples can help the CNN learn the discriminative features necessary to accurately classify them. Additionally, carefully curating the training data by incorporating a variety of morphing techniques and severity levels can further enhance the system's ability to recognize morphed identities.

Furthermore, exploring more advanced CNN architectures or alternative deep learning models specifically designed for face morphing detection may yield better results. These models could incorporate specialized layers or modules that are more sensitive to morphing artifacts, or leverage additional data augmentation techniques tailored for morphed samples.

Regular monitoring and evaluation of the system's performance using appropriate metrics and evaluation protocols are also essential.In addition, exploring ensemble methods or combining multiple classifiers can be beneficial. Ensemble approaches, such as combining the predictions of multiple CNN models or incorporating complementary machine learning algorithms, can enhance the system's robustness and accuracy by leveraging diverse classification strategies.

Overall, addressing the issue of misclassification in the context of face morphing detection requires a comprehensive approach that encompasses data set improvements, model selection, advanced architectures, ongoing evaluation, and the incorporation of feedback from real-world usage scenarios. By

continually refining and optimizing the system, it becomes possible to achieve more reliable and accurate classifications, thus enhancing the security and effectiveness of biometric authentication systems.

LITERATUR

[1] D.Maltoni M. Ferrara A.Franco. "The Magic Passport". In: *IEEE International Joint Conference on Biometrics* (2014). DOI: 10.1109/BTAS.2014.6996240. URL: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6996240.

[2] *NIST*. URL: https://www.nist.gov/programs-projects/face-recognition-technology-feret.

[3] *NIST*. URL: https://www.nist.gov/programs-projects/face-recognition-grand-challenge-frgc.

[4] C.Busch R.Ramachandra K.Raja. "Detecting Morphed Face Images". In: *BTAS 2016* (2016). DOI: 10.1109/BTAS.2016.7791169. URL: https://christoph-busch.de/files/Raghavendra-FaceMorphingDetection-BTAS-2016.pdf.

[5] Timea-Bezdan. "VGG16 Architektur". In: (). URL: https://www.researchgate.net/profile/Timea-Bezdan/publication/333242381/figure/fig2/AS:760979981860866@1558443174380/VGGNet-architecture-19.ppm.