Multi-task learning for joint face recognition and presentation attack detection

Tim Kambič
Faculty of Electrical Engineering, University of Ljubljana
Tržaška cesta 25, SI-1000 Ljubljana
tim.kambic12@gmail.com

Abstract

Presentation attack detection is an important part of face recognition systems and it is a hard problem to solve in general. A Convolutional deep learning model for joint face recognition and presentation attack detection is presented in this paper. The model consists of convolutional "feature extraction" layers (backbone) that are common for both tasks (Face Recognition and Presentation Attack Detection). Two heads follow the feature extraction layers that are used for classification of presentation attack detection and face recognition. Head that is responsible for PAD outputs single hypothesis of presentation attack. Head responsible for FR outputs feature vector that is later used for comparing two or more faces for the task of facial recognition.

The model is evaluated on two datasets: Replay-Attack and Oulu-NPU to test the generalization of presentation attack method. The model is able to detect presentation attack successfully but its ability to generalize is not that good.

1. Introduction

Face recognition (FR) is a task of identifying or verifying a person from a digital image or video based on facial features. Face recognition is mostly used for security purposes, though there is increasing interest in other areas of use. In fact, face recognition technology has received significant attention as it has a potential for a wide range of application related to law enforcement as well as other enterprises. Face biometrics has a prominent role due to its widespread use in international border control and its non-intrusive capture of biometrics data and low-cost sensors.

Face recognition can be done using from simple smartphone cameras to more complex cameras operating in near infrared spectrum or even 3D cameras. Face recognition is usually done in two steps. The first one involves feature extraction and selection while the second step is classification. Traditional methods involve simple linear sub-space modeling (methods like Eigenfaces [25]) or algorithms such as hidden Markov model [16]. In recent years machine learning with deep neural networks has been widely used as a feature extraction tool for facial recognition. Near perfect recognition accuracy can be obtained even on unconstrained datasets such as LFW [10] (Labeled Faces in the Wild) by convolutional neural networks (CNN).

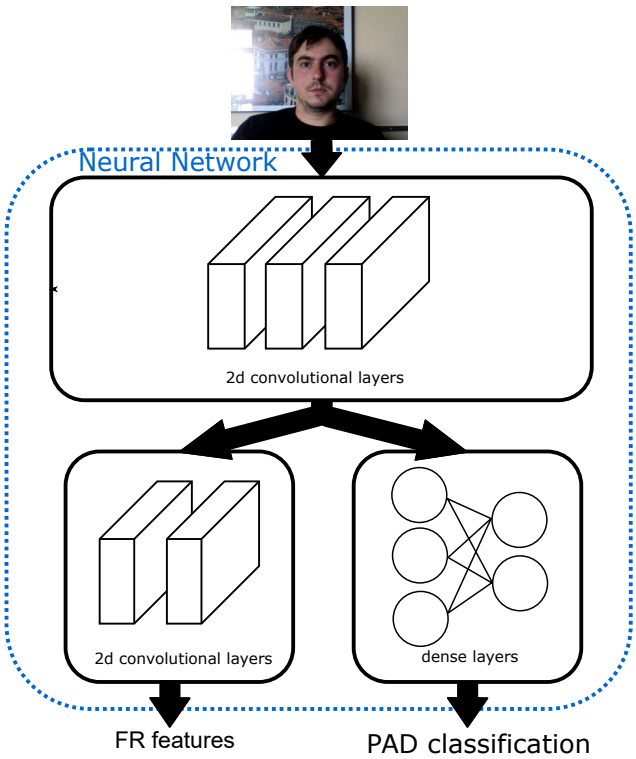

Figure 1. Neural network based model for simultaneous Face Recognition and Presentation Attack detection. Input image is feed through initial "feature extraction" layers that are common for both tasks. After that separate classification layers are used to detect presentation attacks and to calculate feature vector that are used for facial recognition.

Due to non-intrusive biometric data capture, it is also fairly easy to obtain an image of the face of the target individual that can be used as an artifact to fool face recognition

system. Automatic biometric systems that operate without human supervision are especially susceptible to presentation attacks as it is rather effortless to present an image or video to the sensor.

Although convolutional neural networks are very good at the task of face recognition they are still very susceptible to presentation attacks [15].Popular convolutional neural networks have not been trained explicitly for presentation attack detection and are as such unable to deal with them. Additional standalone systems such as [1] can be used to detect presentation attack in the context of FR and another system to do the actual facial recognition.

In this paper, the issue of presentation attack detection for CNN face recognition models is addressed in the way that the same model is employed for face recognition task while at the same time it can detect any presentation attack with just a few additional layers in the neural network. The benefit of this is that only single model must be trained for both the FR and PAD tasks instead of two separate systems.

The rest of the paper is structured as follows: in Section 2 is an overview of existing face recognition and presentation attack detection methods. Proposed method and models are described in Section 3. Datasets, protocols for training and testing and results are covered in Section 4. In Section 5 the summary of results and conclusion are presented.

2. Related Work

Face recognition has been studied quite well in the recent years by many researchers. Presentation attack detection has also been studied but not in such depth as face recognition and usually as a separate topic from face recognition.

2.1. Face recognition

Deep Convolutional Neural Networks (CNNs) have been shown to be very successful at face recognition (FR) tasks [18] [28] [23]. They are able to achieve accuracy above 95% and in some cases higher than 98% [18] [23] on LFW [10] and YTF [27] datasets with regard to FR. Some work has also been done on multi-task learning with regard to faces by [22] where they combined face detection, landmark localization, pose estimation and gender recognition in one deep CNN model. In [29] multi-task learning was used for facial landmark extraction with the help of head pose estimation and facial attribute inference.

2.2. Presentation attack detection

Presentation attack (PA) is the presentation to the biometric data capture subsystem with the goal of interfering with the operation of the biometric system [11]. Presentation attack detection (PAD) is automated determination of a presentation attack [11].

Quite some studies have been presented regarding presentation attacks and their detection in the field of face recognition. But most studies [13] [9] have investigated the vulnerability of systems that rely on handcrafted features and are not based on deep learning methods. PAD systems based on handcrafted features (non deep learning) usually use color texture technique [2], local binary patterns [17] or local phase quantization [20] features and usually operate in HSV or YCbCr color space. Liveness detection can also be used to detect presentation attacks. [24], using information dynamics of the video such as lips movement, eye blinking and facial dynamics.

Some work [19] has also been done on face PAD based on specialized hardware - the Light Field Camera that can record the direction of each incoming ray in addition to the intensity.

Models based on deep neural networks have been shown by Mohammadi A *et al*. [15] to be highly susceptible to presentation attacks. They showed, that CNN-based methods specifically VGG-Face [18] and LightCNN [28] are very poor at detecting presentation attacks. As described in [15] tests were performed on REPLAY-ATTACK [5], REPLAY-MOBILE [7] and MSU-MFSD [26] datasets. On this datasets VGG-Face and LightCNN showed IAPMR (impostor attack presentation match rate - a proportion of PAs that are accepted by FR system as genuine presentations) [21] score above 90% and in some cases even higher than 99%. Recent competitions [1] performed on OULU-NPU [3] dataset have shown the lack of generalization of PADs when operating in real-world conditions. Presentation attack detection methods also do not generalize well to new unseen PAs. In [1] they focused only on building models for PA detection without face recognition but it might be useful to combine the FR and PAD tasks together. In [1] it was also mentioned that current public datasets may not include enough data to train deep CNNs for PAD from scratch.

3. Face Recognition and Presentation attack detection

In multi-task learning, we aim to maximize the performance of multiple related tasks by learning them jointly.

In this paper we use the same convolutional filters (also called backbone) in the neural network for presentation attack detection and facial recognition. Upon that feature extraction layers, two heads of classification layers are added. The first head is used for facial recognition and the second one for presentation attack detection. Three models (labeled A, B and C) are designed that differ in architecture and training procedures.

Most of the architecture of the proposed model is taken from Vgg-Face [18] a deep CNN used for the task of facial recognition. Vgg-Face uses end-to-end learning with a high number of training samples (2.6 million).

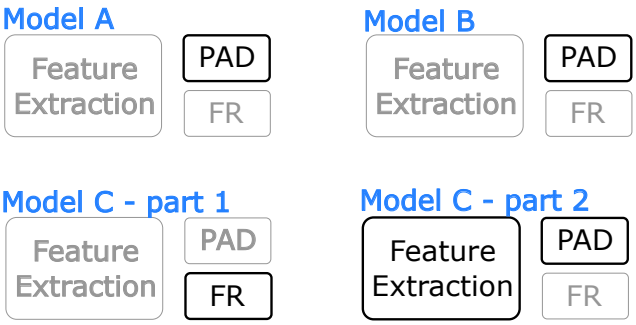

Figure 2. Training of different models. Parts of the network that are trained are displayed in bold black color.

3.1. Feature extraction

The first part of the network is used as a feature extraction tool whose features are further used for face recognition and presentation attack detection. All three models A, B and C use the same convolutional layers for feature extraction. This part of the network consists of 13 convolutional layers with architecture that can be seen in Table 1. This architecture is the same as the first part of the well-known face recognition neural network VGG-Face [18]. The input to this part of the network is the input tensor of RGB image with the size of 224x224x3. The output of this part of the network is tensor with the size of 14x14x512. A rectified linear unit (ReLu) is used as an activation function for all convolutional layers.

In Model A and Model B variants, the weights for this layers are taken from pretrained VGG-Face network and are not changed during training. In Model C variant the convolutional layers are further fine tunned starting from pretrained VGG-Face weights. The training procedure is described in Section 4.2. Basic overview of trained parts are shown in Figure 2.

3.2. Face Recognition

The second part of the network is the part responsible for face recognition task. For the facial recognition part of the network the same architecture as the second part of the VGG-Face [18] network is used. Architecture of layers can be seen in Table 2. The activation function for all of the layers is Rectified Linear Unit (ReLu). The input to this part of the network is the output tensor of the *Feature extraction* with the size of 14x14x512 that is feed through a *Max Pooling* operation (pool size of 2x2, a stride of 2) so that its dimensions changes to 7x7x512. The output of this part of the network is a feature vector with the size of 2622. When this feature vector is computed for two images the actual face recognition task can be done by comparing the two feature vectors. Comparison can be done by simple Euclidean distance (Equation 1) or Cosine distance (Equation 2) calculation and thresholding. Where a_i and b_i are i-th elements of the output vector and D is the resulting distance.

	number of filters	kernel size	additional
1	64	3x3	
2	64	3x3	*MaxPooling*
3	128	3x3	
4	128	3x3	*MaxPooling*
5	256	3x3	
6	256	3x3	
7	256	3x3	*MaxPooling*
8	512	3x3	
9	512	3x3	
10	512	3x3	*MaxPooling*
11	512	3x3	
12	512	3x3	
13	512	3x3	

Table 1. Convolutional feature extraction layers. *MaxPooling* indicates 2D max pooling operation with pool size of 2x2 and stride of 2

$$D = \sqrt{\sum_{i=1}^{n}(a_i - b_i)} \tag{1}$$

$$D = \frac{\sum_{i=1}^{n} a_i b_i}{\sqrt{\sum_{i=1}^{n} a_i^2}\sqrt{\sum_{i=1}^{n} b_i^2}} \tag{2}$$

When training the model for facial recognition additional densely connected layer with softmax activation function has to be added to the last layer of this part of the network. The number of outputs in that additional layer has to be the same as the number of different identities/persons in used training set. This ensures that model can be trained with one-hot encoded labels (one bit that corresponds to the identity of the person is set to one and all other bits are set to zero) for the task of face recognition.

All three models (Model A, B and C) have the same face recognition part of the network. In Model A and Model B the weights for the layers are taken from pretrained VGG-Face neural network and are not changed during training. That decision was made due to VGG-Face already having an excellent score at face recognition task [18]. For Model C the face recognition part of the network is fine tunned starting from pretrained VGG-Face weights.

3.3. Presentation attack detection

The final part of the network is responsible for presentation attack detection task. This part consists of densely connected layers of neurons (2 layers for Model A and 3 layers for Model B and C) as can be seen in Table 3. The input to this part of the network is output tensor from *Feature extraction* part that is feed through a *Global Average Pooling*

number of filters	kernel size	additional
4096	7x7	*Dropout*
4096	1x1	*Dropout*
2622	1x1	

Table 2. Face recognition layers of the models. *Dropout* indicates dropout operation (randomly setting a fraction of input units to 0 at each update during training time) with the rate of 0.5

layer	number of neurons		activation
	Model A	*Model B & C*	
1	/	*100*	*ReLu*
2	100	*80*	*ReLu*
3	1	1	*TanH*

Table 3. Presentation attack detection layers of the model. *ReLu* indicates rectified linear unit activation function and *TanH* indicates hyperbolic tangent activation function

operation [14] that reduces spatial three-dimensional tensor and outputs vector with the size of 512. The final output is the classification whether or not the input image is a presentation attack or a genuine sample. Presentation attack detection can thus be performed by simple thresholding of output as it will range from -1 (for a presentation attack) to 1 (for a genuine person).

Weights for this part of the network in Model A and Model B are trained from scratch. For Model C the weights are initialized to the values of trained Model B and are further fine tuned.

4. Experimental Results

4.1. Datasets

REPLAY-ATTACK [5] For training and evaluating presentation attack detections and face recognition the Replay-Attack database was used. It consists of 1300 video clips of video and photo attack attempts under different lighting conditions. The database is already split into four standard groups: training data (for training anti-spoof classifier), development data (for threshold estimation), test data (for reporting error figures) and enrollment data (used to verify spoofing sensitivity). All videos of the attacks are 9 seconds long with a resolution of 320x240px. Examples of video frames in database can be seen in Figure 3 (top row). Videos were taken under two illumination conditions: *controlled*, i.e. uniform background and a fluorescent lamp was used to illuminate the scene, and *adverse*, i.e. non-uniform background and the day-light was the only source of illumination. The training set consists of 60 real-access and 300 attack videos while Testing set consists of 80 real-access and 400 attack videos. The database also includes annotations of identities of persons in the videos so it was also used for fine tuning the face recognition part of the network.

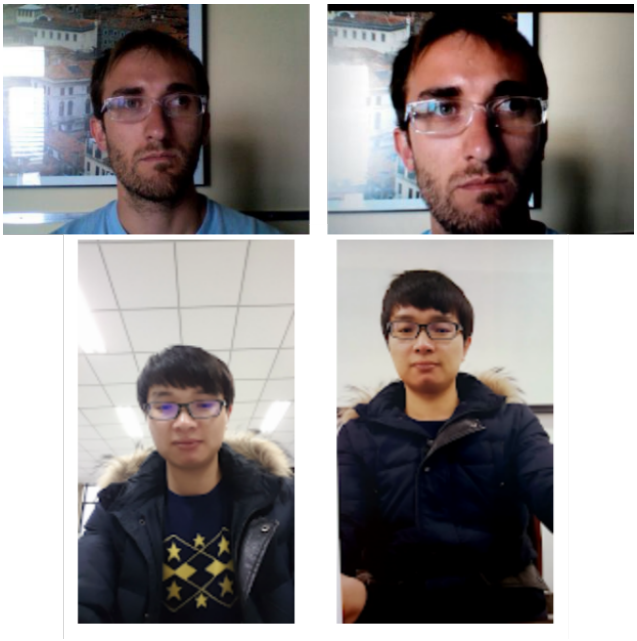

Figure 3. Replay-Attack example frames: real-access (top left) and attack (top right), Oulu-NPU example frames: real-access (bottom left) and attack (bottom right)

Oulu-NPU [4] The second dataset for evaluating PAD was Oulu-NPU database. The dataset consists of 4950, 5 seconds long real-access and attack videos recorded by a front-facing camera on six different smartphones. Resolution of captured video is 1920x1080px. The videos were collected in three sessions with different illumination conditions. The videos of the real-accesses and attacks, corresponding to the 55 subjects, are divided into three subject-disjoint subsets for training, development and testing with 20, 15 and 20 users, respectively. Each of the training and testing set has 360 real-access and 1440 attack videos. Example of video frames can also be seen in Figure 3 (bottom row). For easier evaluation, only the videos for the first evaluation protocol described in the database were used. First protocol in Oulu-NPU database is used to evaluate generalization of PAD methods (different illumination conditions and location) but it does not matter for our experiments as the models are never trained on Oulu-NPU dataset.

4.2. Experimental setup

The official evaluation protocol of Replay-Attack database was used to allow comparison with other methods proposed in the literature [2],[5] with regard to presentation attack detection. Receiver operating characteristic (ROC) curve is displayed for both PAD and facial recognition tasks. The results of PAD are given on the development set of the database in terms of EER (equal error rate) and HTER (half total error rate) on the test set. EER is the point

on ROC where false acceptance rate and false rejection rate are the same. HTER is the mean value of FAR(false acceptance rate) and FRR(false rejection rate). For comparison of the results two other models were selected: model proposed by the authors of Replay Attack database [5] and CoALBP. CoALBP is the best performing model from [2] operating on images in YCbCr colorspace and is based on color texture analysis.

For training the model 60 real-access and 150 attack videos in the Replay-Attack's training group were used. From videos single frames were extracted at fixed periods and saved for training. Images were resized to 224x224px RGB as that is the input dimension for the proposed CNN. Pixel values are scaled to the range -1 to 1. The same procedure was also applied to videos in the test group of Replay-Attack and also when processing videos from Oulu-NPU dataset.

Videos were sampled at 1 second interval which produced 2400 images for training and 4947 test images (3200 from Replay-Attack and 1747 from Oulu-NPU datasets). Training data, therefore, consists of 1500 attack and 900 real access images and test data consists of 2000 attack and 1200 real access images from Replay-Attack database. Distribution of data in Oulu-NPU's test set is 1398 attack and 349 real access images. Due to manageable size of images all of the training and test images were loaded to RAM at the start of the training which increased training and inception speed. When training the facial recognition part of the network the 15 different identities in the training set of Replay-Attack dataset were used. One epoch of training took from 18 seconds (for Model A) to 56 seconds (for Model C) with GPU (Nvidia GTX 1070 Ti, 8GB of RAM) and DDR3 RAM. Inception on the model takes about 9 milliseconds per image on the same GPU hardware (when processing in batches of 16 samples) and 270 milliseconds on i7-6700HQ CPU and DDR4 RAM (in batches on one sample). For training the batch size was set to 16 due to GPU's RAM constraints. The model was constructed and trained in Keras [6] using the Tensorflow backend. Adam optimizer [12] was used for training with the following default parameters: learning rate=0.001, beta1=0.9, beta2=0.999, decay=0. Weights in densely connected layers that were trained from scratch were initialized using Glorot [8] uniform initializer. For presentation attack detection head of the network the Mean Square Error loss was chosen and for the face recognition head of the network the categorical cross-entropy loss was selected.

Overview of models training procedure, that can also be seen in Figure 2:

- *Model A:* Two densely connected layers for PAD are trained, all other parts of the network are frozen for training.
- *Model B:* Three densely connected layers for PAD are trained, all other parts of the network are frozen for training.
- *Model C:* Densely connected layer is added to the output of the FR part with softmax activation function and trained to the desired accuracy with all other layers frozen (part 1). PAD layers are initialized from Model B. After that the convolutional layers of Feature extraction part and PAD head are fine tuned for both PAD and FR (part 2).

Training was stopped when the model started to overfit and the accuracy on test set started to drop. Model A was trained for three epochs, Model B also for threes epochs and Model C for five epochs for the first part and two epochs for the second part.

4.3. Results

The first part of the experiments was used to evaluate the model's ability to detect presentation attacks when being tested with the same dataset as the model was trained with. Therefore Replay-Attack [5] was first used to train the head of the network responsible for presentation attack detection for all three models (as it was described in Section 4.2). The resulting Half Total Error rate and Equal Error Rate can be seen in Table 4 and ROC curves in Figure 4 (first three curves - orange, blue, red). Performance of all three models is comparable. Model A has better HTER than Model B but worse EER. Model C probably started to overfit so its performance was degraded and not improved. All three developed models have better performance on Replay-Attack database than the initial method proposed by the authors of the database. But the proposed models are still not better than the CoALBP model [2]. Additional layer in Model B over Model A did not significantly improve or degrade the performance.

Method	**HTER**	**EER**
Replay Attack baseline	0.138	/
CoALBP	0.047	0.014
Model A	0.082	0.055
Model B	0.065	0.084
Model C	0.085	0.094

Table 4. HTER and EER scores for different models.

For the second part of the experiments, the model's PAD ability was tested on different dataset than the model was trained on. In this instance, the model was again trained on Replay-Attack database but the test set came from Oulu-NPU [4] dataset. The performance of presentation attack detection can be seen in Figure 4 (last two curves - dark/light green). The performance is strongly degraded but it still has some meaning at HTER of 0.453 for Model A

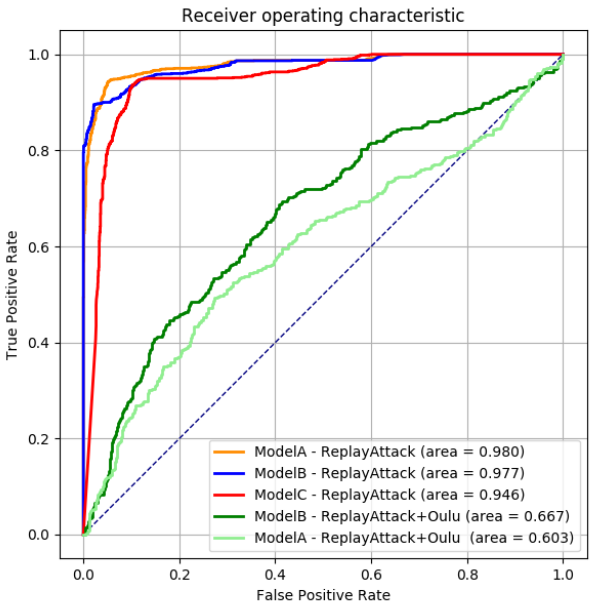

Figure 4. ROC curves for presentation attack detection task of different models described (*area* is used to describe area under the curve)

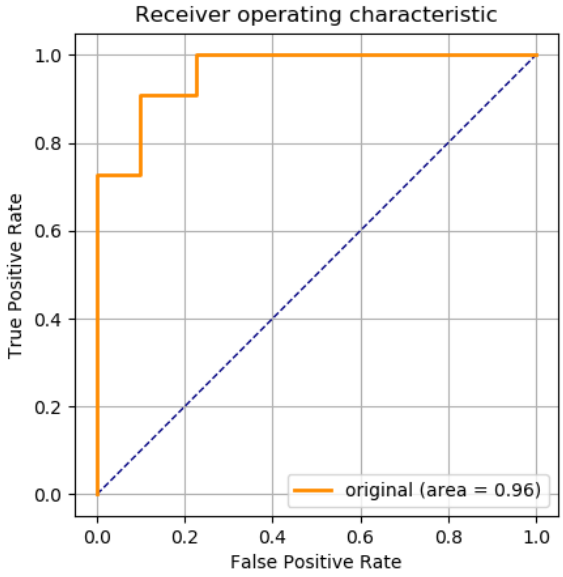

Figure 5. ROC curve for face recognition task of the model (*area* is used to describe area under the curve)

and HTER of 0.400 for Model B. Here the higher complexity (one additional hidden layer) of Model B comes to effect and better performance and generalization can clearly be viewed. In contrast to the first part of the experiment where additional layer did not show any improvement. Due to worse performance in the first part Model C was not evaluated with Oulu-NPU dataset.

Facial recognition part of the network was not changed in Model A and Model B and its performance is that of the original VGG-Face. It is not near perfect because the images were not cropped to only include the facial region. That decision was made so that more features can be used for presentation attack detection. ROC curve for the task of facial recognition evaluated on 40 pairs of images from the Replay-Attack dataset (20 of matching persons and 20 of different persons) can be seen in Figure 5. The similarity of faces is computed using Euclidean distance (Equation 1) on the two output vectors of size 2622. Performance of Model C in the task of face recognition was not evaluated as the training did not yield any useful results.

All of the implementation code can be found on *https://github.com/timkambic/ FaceRecognitionAndPresentationAttackDetection*.

5. Conclusion

Convolutional deep learning model for joint face recognition and presentation attack detection was presented in this paper. The model consists of convolutional "feature extraction" layers that were common for both tasks (FR and PAD) - the backbone. Following that are two heads used for classification of PAD and FR. PAD head outputs single hypothesis of presentation attack. FR head outputs feature vector that is later used for comparing two or more faces for the task of facial recognition.

The results showed that the same convolutional filters can be used for both PAD and FR tasks effectively. This means that models can be smaller and computationally more efficient while achieving two tasks.

The generalization of the models showed to be quite a difficult problem. When testing the model with another dataset that was not used for training the performance of the model dropped significantly. Transferability of the PAD models to other PAD datasets and especially unseen attacks is still an unsolved issue.

Face recognition of the proposed models is high due to the models only being an upgrade to the VGG-Face [18] model that was developed solely for the task of facial recognition. Face recognition performance could be even higher if images of faces were cropped to only include the face of the person without the background. But the decision was made to not crop the images to and to include the background to aid in the task of presentation attack detection.

Additional data and training could be used to further improve the model's performance. But as mentioned above data has to come from many different sources to aid in a generalization of PAD. In some applications where people are enrolled (e.g. where FR is used for access control) model could also be fine tuned to specific persons after en-

rollment.

When trying to train the model on Oulu-NPU dataset the problem of imbalanced data came into the effect, as there is significantly more attack than real access videos. Some data augmentation techniques could be used to increase the number of real access samples. Mixing the two datasets together could also yield interesting result as that would increase the number of samples and also the number of different attacks, illumination conditions and locations. But that would also increment the difficulty of presentation attack detection and would maybe need a more complex model.

It was showed that the same convolutional filters can be used for PAD and FR tasks quite effectively. And as models for great facial recognition already exist there is no need to train the models for presentation attack detection from scratch. If some presentation attack samples exist then the model can easily be fine tuned and that quickly yields better results.

References

[1] Z. Boulkenafet, J. Komulainen, Z. Akhtar, A. Benlamoudi, D. Samai, S. E. Bekhouche, A. Ouafi, F. Dornaika, A. Taleb-Ahmed, L. Qin, F. Peng, L. B. Zhang, M. Long, S. Bhilare, V. Kanhangad, A. Costa-Pazo, E. Vazquez-Fernandez, D. Perez-Cabo, J. J. Moreira-Perez, D. Gonzalez-Jimenez, A. Mohammadi, S. Bhattacharjee, S. Marcel, S. Volkova, Y. Tang, N. Abe, L. Li, X. Feng, Z. Xia, X. Jiang, S. Liu, R. Shao, P. C. Yuen, W. R. Almeida, F. Andalo, R. Padilha, G. Bertocco, W. Dias, J. Wainer, R. Torres, A. Rocha, M. A. Angeloni, G. Folego, A. Godoy, and A. Hadid. A competition on generalized software-based face presentation attack detection in mobile scenarios. In *2017 IEEE International Joint Conference on Biometrics (IJCB)*, pages 688–696, Oct 2017.

[2] Z. Boulkenafet, J. Komulainen, and A. Hadid. Face spoofing detection using colour texture analysis. *IEEE Transactions on Information Forensics and Security*, 11(8):1818–1830, Aug 2016.

[3] Z. Boulkenafet, J. Komulainen, L. Li, X. Feng, and A. Hadid. Oulu-npu: A mobile face presentation attack database with real-world variations. In *2017 12th IEEE International Conference on Automatic Face Gesture Recognition (FG 2017)*, pages 612–618, May 2017.

[4] Z. Boulkenafet, J. Komulainen, L. Li, X. Feng, and A. Hadid. OULU-NPU: A mobile face presentation attack database with real-world variations. May 2017.

[5] I. Chingovska, A. Anjos, and S. Marcel. On the effectiveness of local binary patterns in face anti-spoofing. In *2012 BIOSIG - Proceedings of the International Conference of Biometrics Special Interest Group (BIOSIG)*, pages 1–7, Sept 2012.

[6] F. Chollet et al. Keras. `https://keras.io`, 2015.

[7] A. Costa-Pazo, S. Bhattacharjee, E. Vazquez-Fernandez, and S. Marcel. The replay-mobile face presentation-attack database. pages 1–7, Sept 2016.

[8] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In *JMLR W&CP: Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics (AISTATS 2010)*, volume 9, pages 249–256, May 2010.

[9] A. Hadid. Face biometrics under spoofing attacks: Vulnerabilities, countermeasures, open issues, and research directions. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2014.

[10] G. B. Huang, M. Mattar, T. Berg, and E. Learned-Miller. Labeled Faces in the Wild: A Database forStudying Face Recognition in Unconstrained Environments. In *Workshop on Faces in 'Real-Life' Images: Detection, Alignment, and Recognition*, Marseille, France, Oct. 2008. Erik Learned-Miller and Andras Ferencz and Frédéric Jurie.

[11] Information technology — Biometric presentation attack detection. Standard, International Organization for Standardization, Geneva, CH, Jan. 2016.

[12] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization, 2014.

[13] N. Kose and J. Dugelay. On the vulnerability of face recognition systems to spoofing mask attacks. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 2357–2361, May 2013.

[14] M. Lin, Q. Chen, and S. Yan. Network in network, 2013.

[15] A. Mohammadi, S. Bhattacharjee, and S. Marcel. Deeply vulnerable: a study of the robustness of face recognition to presentation attacks. *IET Biometrics*, 7(1):15–26, 2018.

[16] A. V. Nefian and M. H. Hayes. An embedded hmm-based approach for face detection and recognition. In *1999 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings. ICASSP99 (Cat. No.99CH36258)*, volume 6, pages 3553–3556 vol.6, March 1999.

[17] T. Ojala, M. Pietikainen, and T. Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):971–987, July 2002.

[18] P. O.M., V. A., and Z. A. Deep face recognition. *Machine Vision Conf.*, 1(1):41.1–41.12, 2015.

[19] R. Raghavendra, K. B. Raja, and C. Busch. Presentation attack detection for face recognition using light field camera. *IEEE Transactions on Image Processing*, 24(3):1060–1075, March 2015.

[20] E. Rahtu, J. Heikkilä, V. Ojansivu, and T. Ahonen. Local phase quantization for blur-insensitive image analysis. *Image and Vision Computing*, 30(8):501 – 512, 2012. Special Section: Opinion Papers.

[21] R. Ramachandra and C. Busch. Presentation attack detection methods for face recognition systems: A comprehensive survey. *ACM Comput. Surv.*, 50(1):8:1–8:37, Mar. 2017.

[22] R. Ranjan, V. M. Patel, and R. Chellappa. Hyperface: A deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2018.

[23] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. In *The*

IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2015.

[24] S. Tirunagari, N. Poh, D. Windridge, A. Iorliam, N. Suki, and A. T. S. Ho. Detection of face spoofing using visual dynamics. *IEEE Transactions on Information Forensics and Security*, 10(4):762–777, April 2015.

[25] M. A. Turk and A. P. Pentland. Face recognition using eigenfaces. In *Proceedings. 1991 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 586–591, June 1991.

[26] D. Wen, H. Han, and A. K. Jain. Face spoof detection with image distortion analysis. *IEEE Transactions on Information Forensics and Security*, 10(4):746–761, April 2015.

[27] L. Wolf, T. Hassner, and I. Maoz. Face recognition in unconstrained videos with matched background similarity. In *CVPR 2011*, pages 529–534, June 2011.

[28] X. Wu, R. He, Z. Sun, and T. Tan. A Light CNN for Deep Face Representation with Noisy Labels. *ArXiv e-prints*, Nov. 2015.

[29] Z. Zhang, P. Luo, C. C. Loy, and X. Tang. Facial landmark detection by deep multi-task learning. In D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 94–108, Cham, 2014. Springer International Publishing.