Recent Advances in Biometric Security: A Case Study of Liveness Detection in Face Recognition

Koichi Ito*, Takehisa Okano* and Takafumi Aoki*
* Graduate School of Information Sciences, Tohoku University, Japan.
E-mail: ito@aoki.ecei.tohoku.ac.jp

***Abstract*—Biometrics using biological or behavioral features to authenticate a person has attracted much attention as a new authentication approach against traditional ones such as key, password, etc. Biometrics technologies provide us better security and greater convenience than traditional person authentication technologies such as key, password and card. Biometric systems with cameras involve the risk of spoofing. For example in face recognition systems, when a malicious person turns a printed face photo of an authenticated user to a camera, a face recognition system may accept the malicious person as the authenticated user. To address the above problem, liveness detection is important to develop secure biometric recognition systems. This paper presents a liveness detection method using deep learning for face recognition systems.**

I. Introduction

Biometrics uses biological or behavioral features to authenticate a person and has attracted much attention as a new authentication approach against traditional ones such as key, password, etc [4]. Biometric traits such as fingerprint, face, iris, voice, signature and gait are not stolen and forgotten compared with traditional tokens such as key and password. Therefore, biometrics technologies provide us better security and greater convenience than traditional authentication technologies. Person authentication systems using fingerprint, face, iris, etc. have been commercially available and used in access control, ATM, etc.

There are some attack scenarios for biometric systems as shown in Fig. 1, although the convenient person authentication systems can be realized by biometric technologies. The inside of systems may be protected against such attacks by cryptographic technologies, while the attack scenario of spoofing, which is the outside of systems, cannot be protected by any cryptographic technologies. Therefore, we have to address the spoofing attack in order to keep biometric systems secure. This paper focuses on a face recognition system as a major example of this problem. A face image captured by a camera is used to recognize a person in the typical face recognition system, where facial features are extracted from the input face image, the similarity is measured by comparing input and registered features, and a decision is made according to the similarity [8]. When a malicious person turns a printed face photo of an authenticated user to a camera, the face recognition system may accept the malicious person as the authenticated user. This result is not necessarily wrong for the face recognition system, since face matching algorithms are designed so as to be robust against various factors such as noise, pose changes, facial expression changes, etc. Liveness detection of input images has to be introduced as a countermeasure against spoofing attacks [9].

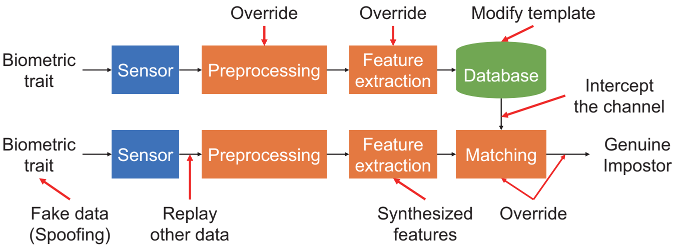


Fig. 1. Attacks on a biometric recognition system.

Spoofing attack can be modeled that the input image is degraded by factors between the real face of the authenticated user and the camera on the face recognition system, assuming that the printed photo is taken by another camera and is printed by a printer. In this case, such degradation factors are a camera, a printer, a paper, etc. Another case is to use a tablet and a smartphone, where a malicious person turns a displayed photo to a camera. In this case, the degradation factor is a display device. Therefore, liveness detection in face recognition is equivalent to detect degradation factors included in the input image.

There are some works on liveness detection in face recognition [2], [11], [5]. Most methods employ Local Binary Patterns (LBP) [12] to extract local features for liveness detection. LBP is obtained by thresholding neighborhoods of each pixel with the center pixel value, and then the histogram of LBPs is used as a texture descriptor. Chigovska et al. created the REPLAY-ATTACK database[1] for evaluating the accuracy of liveness detection methods and demonstrated the effectiveness of LBP-based methods [2]. Pereira et al. proposed a liveness detection method using a Local Binary Pattern from Three Orthogonal Planes (LBP-TOP) extracted from a video sequence represented in the spatio-temporal domain [11]. Kim et al. proposed a Local Speed Pattern (LSP), which is based on illumination changes [5]. The error rate of these LBP-

[1] https://www.idiap.ch/dataset/replayattack

based methods were 10%∼20% through experiments using the REPLAY-ATTACK database. The accuracy of LBP-based methods is not sufficient for liveness detection, since it is hard to empirically design local features under such complex conditions in general. Therefore, deep learning techniques [3] can be used to address the above problem.

Deep learning is one of machine learning techniques and is a multi-layer neural network with a lot of parameters. In the field of image processing, Convolutional Neural Network (CNN) proposed by LeCun et al. [7] exhibits efficient performance on image recognition problems [6]. Yang et al. [14] proposed a liveness detection method using CNN. They employed AlexNet [6] to extract features from images and classified them into real and fake by Support Vector Machine (SVM). Menotti et al. [10] proposed a liveness detection method using CNN and filter optimization. This method extracts features from images by CNN and classifies them into real and fake by SVM as well as Yang et al [14]. Alotabib et al. [1] proposed a simple CNN-based method for liveness detection. Unlike the above methods, images are classified by CNN. Through experiments using the REPLAY-ATTACK database[1], the error rate of methods proposed by Yang et al. [14] and Menotti et al. [10] is 2.30% and 0.75%, respectively, although that of Alotabib et al. [1] is 10%. Therefore, CNN-based methods exhibit significantly higher accuracy for liveness detection than LBP-based methods.

This paper proposes a CNN-based liveness detection method for face recognition systems. Our method is based on the CNN architectures from CIFAR-10[2] and AlexNet [6]. Conventional methods extract a face region from the input image at first. The use of the whole image makes it possible to improve the accuracy of liveness detection, since the purpose is to detect unexpected degradation factors included in the input image. The accuracy of the proposed method is evaluated using the REPLAY-ATTACK database as well as conventional methods. We demonstrate that the accuracy of the proposed method is the highest in liveness detection methods through experiments using the REPLAY-ATTACK database.

II. Liveness Detection Method Using CNN

This section describes the CNN-based liveness detection method proposed in this paper. The proposed method employs CNN architectures inspired by CIFAR-10 and AlexNet. We implement the proposed method using the open-source machine learning library TensorFlow[3] provided by Google.

A. Fundamentals of CNN

CNN consists of the combination of convolution, pooling, normalization and fully-connected layers, which varies depending on problems. The brief description of each layer is given below.

- `conv` indicates a convolution layer, which applies filters to the image. Then, an activation function, i.e., rectified linear function, is applied to outputs, where this paper uses Rectified Linear Unit (ReLU) as a rectified linear function.

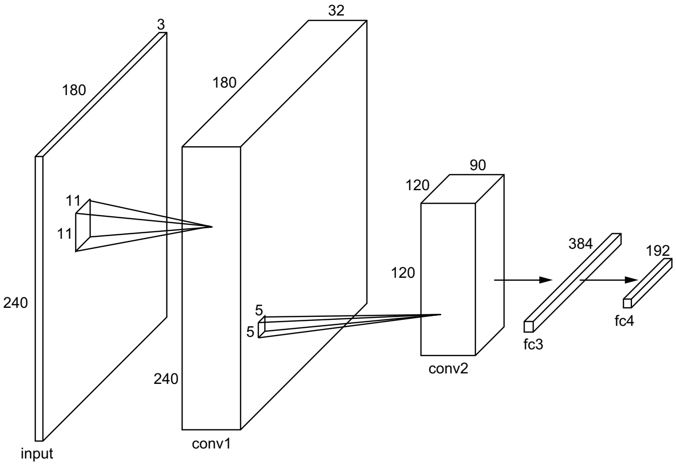


Fig. 2. Architecture of CIFAR-10 used in the proposed method.

- `pool` indicates a pooling layer, which enhances the robustness against translational displacement by replacing the rectified region by its maximum or average pixel value. This paper employs the max pooling, which replaces the region by its maximum pixel value. The pooling layer can set the stride as well as the convolution layer.
- `norm` indicates a normalization layer, which normalizes the pixel values as well as general image processing. For example, the normalization is performed by subtracting the average pixel value from each pixel. This paper employs Local Response Normalization (LRN) [6], which normalizes pixel values in every local region.
- `fc` indicates a fully-connected layer. The softmax function is applied to the output of the fully-connected layer so as to obtain the final output as the probability of classification.

B. CIFAR-10

The CIFAR-10[2] dataset consists of about 60-thousand images selected from about 80 million images in 80 Million Tiny Images[4]. Images are 32×32 pixels and are classified into 10 classes, i.e., airplane, automobile, bird, cat, deer, dog, frog, horse, ship and truck. The result of the baseline algorithm using cuda-convnet[5] has been reported. The CNN architecture used in the baseline algorithm is called CIFAR-10. The architecture used in the proposed method inspired by CIFAR-10 is illustrated in Fig. 2 and Table I.

C. AlexNet

AlexNet has exhibited the best result in the classification task of 1,000 object categories in ImageNet Large Scale Visual

[2] https://www.cs.toronto.edu/∼kriz/cifar.html

[3] https://www.tensorflow.org/

[4] http://groups.csail.mit.edu/vision/TinyImages/

[5] https://code.google.com/p/cuda-convnet/
https://github.com/akrizhevsky/cuda-convnet2

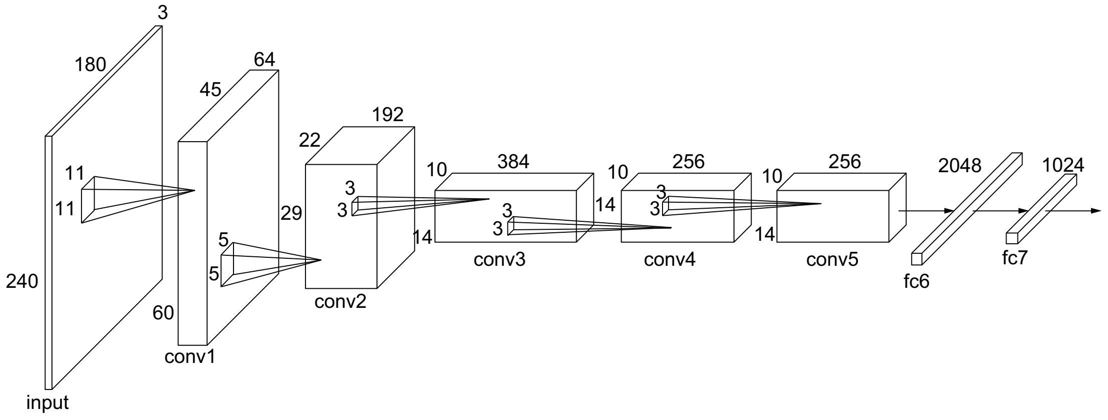


Fig. 3. Architecture of AlexNet.

TABLE I
DETAIL OF LAYERS IN CIFAR-10 USED IN THE PROPOSED METHOD.

Layer	Kernel	Stride	Outputs	Function
input	—	—	180×240×3	—
conv1	11×11	4	180×240×32	ReLU
pool1	3×3	2	90×120×32	—
norm1	—	—	90×120×32	LRN
conv2	5×5	1	90×120×20	ReLU
norm2	—	—	90×120×20	LRN
pool2	3×3	2	45×60×20	—
fc3	384	—	384	ReLU
fc4	192	—	192	ReLU
softmax	2	—	2	softmax

TABLE II
DETAIL OF LAYERS IN ALEXNET USED IN THE PROPOSED METHOD.

Layer	Kernel	Stride	Outputs	Function
input	—	—	180×240×3	—
conv1	11×11	4	45×60×64	ReLU
pool1	3×3	2	22×29×64	—
norm1	—	—	22×29×64	LRN
conv2	5×5	1	22×29×192	ReLU
pool2	3×3	2	10×14×192	—
norm2	—	—	10×14×192	LRN
conv3	3×3	1	10×14×384	ReLU
conv4	3×3	1	10×14×256	ReLU
conv5	3×3	1	10×14×256	ReLU
pool5	3×3	2	4×6×256	—
norm5	—	—	4×6×256	LRN
fc6	2048	—	2048	ReLU
fc7	1024	—	1024	ReLU
softmax	2	—	2	softmax

Recognition Challenge 2012 (ILSVRC2012)[3]. AlexNet has reduced classification errors more than 10% and has had a significant impact on the field of image recognition at that time. The architecture used in the proposed method inspired by AlexNet is illustrated in Fig. 3 and Table II. This architecture has 3 more convolution layers and one more fully-connected layer compared with CIFAR-10.

D. Training

In general, network parameters of CNN are trained using hundreds of thousands of images. Hence, a large-scale image dataset must be prepared in advance. If parameters are trained using a small dataset, overfitting may occur. On the other hand, a large-scale dataset cannot be prepared depending on the problem. In this case, data augmentation has to be introduced to prevent overfitting. For example, more images are generated by translation, rotation, scaling, flip, etc. This paper introduces the following data augmentation in training. A sub-image with 240×180 pixels is extracted from the random location of RGB-colored images with 320×240 pixels. Then, we add some variations to the sub-images such that the sub-image is flipped, the brightness is changed and the contrast is changed. Note that a sub-image with 240×180 pixels is extracted from the center of an image with 320×240 pixels in testing, which is used as an input.

Trained parameters are publicly available for the famous network architecture such as AlexNet. For example, the parameters of AlexNet trained using ImageNet is available, which is optimized to classify images into 1,000 object categories. Some recent studies demonstrated that such pre-trained parameters are also effective in solving other problems [13]. Therefore, this paper utilizes the parameters of AlexNet trained using ImageNet as initial parameters of AlexNet so as to construct the refined network. Table III shows the architecture of the pre-trained AlexNet used in the proposed method.

E. Liveness Detection

The proposed method extracts features from an input image using CNN and classifies the image into real and fake according to features. The proposed method uses the whole image, while conventional methods extract a face region from an input image. Spoofing can be considered as a model of adding degradation factors from a camera, a printer and a display device to an input image. If only a face region is focused, features may be extracted based only on facial features. Such features are not always effective to detect degradation factors

TABLE III
DETAIL OF LAYERS IN PRE-TRAINED ALEXNET USED IN THE PROPOSED METHOD.

Layer	Kernel	Stride	Outputs	Function
input	—	—	227×227×3	—
conv1	11×11	4	57×57×96	ReLU
norm1	—	—	57×57×96	LRN
pool1	3×3	2	28×28×96	—
conv2	5×5	1	28×28×256	ReLU
norm2	—	—	28×28×256	LRN
pool2	3×3	2	13×13×256	—
conv3	3×3	1	13×13×384	ReLU
conv4	3×3	1	13×13×384	ReLU
conv5	3×3	1	13×13×256	ReLU
pool5	3×3	2	6×6×256	—
fc6	2048	—	4096	ReLU
fc7	1024	—	4096	ReLU
softmax	2	—	2	softmax

included in images. Therefore, the proposed method uses the whole image in liveness detection.

The proposed method uses the output of the last convolution layer or the output of the last fully-connected layer as features for liveness detection. The output of pool2 and fc4 is used in CIFAR-10, while the output of norm5 and fc7 is used in AlexNet. This paper employs linear SVM as a classifier for liveness detection. In addition, the final output of CNN, i.e., the output of softmax, is used for the purpose of comparison in this paper, which is usually used in CNN-based image classification.

III. EXPERIMENTS AND DISCUSSION

This section describes performance evaluation of the proposed method. This paper uses the REPLAY-ATTACK database[1] to evaluate the effectiveness of the proposed method in liveness detection for face recognition systems.

The REPLAY-ATTACK database consists of a set of video sequences taken from 50 subjects of both real and fake scenarios. Video sequences are taken under two different conditions: (i) “controlled” indicates that the background is uniform and the illumination is a fluorescent lamp and (ii) “adverse” indicates that the background is non-uniform and the illumination is daylight. Each video sequence is with 320×240 pixels at 25 fps and of 15 sec. (375 frames). Note that the image frames, i.e., still images, extracted from each video sequence are used in the experiments. In the case of using still images as inputs, motion cannot be used in liveness detection. Therefore, using still images is a more difficult condition in liveness detection than using video sequence. The REPLAY-ATTACK database has three attack scenarios: (i) “print-attack” is that a malicious person turns a printed photo of an authenticated user to a camera, (ii) “phone-attack” is that a malicious person turns a smartphone screen with displaying a photo or a video of an authenticated user to a camera and (iii) “tablet-attack” is that a malicious person turns a tablet screen with displaying a photo or a video of an authenticated user to a camera. Each attack video sequence is captured for about 10 sec. under two different conditions: (i) “hand” is that a malicious person holds the attack media or device using own hands and (ii) “fixed” is that the attack media or device is fixed so as not to move during the spoofing attack. The database is classified into three subsets: (i) train, (ii) dev and (iii) test so as to evaluate the accuracy of liveness detection methods under the same condition. “train” is used to train classifiers, “dev” is used to estimate when to stop training and to determine the threshold for evaluating the accuracy and “test” is used to evaluate the accuracy. Table IV shows the configuration of the REPLAY-ATTACK database and Fig. 4 shows an example of images in the database.

TABLE IV
CONFIGURATION OF THE REPLAY-ATTACK DATABASE: THE LEFT SIDE INDICATES THE NUMBER OF “HAND” AND THE RIGHT SIDE INDICATES THE NUMBER OF “FIXED”.

Type	train	dev	test	Total
Real-access	60	60	80	200
Print-attack	30+30	30+30	40+40	100+100
Phone-attack	60+60	60+60	80+80	200+200
Tablet-attack	60+60	60+60	80+80	200+200
Total	360	360	480	1,200

TABLE V
COMBINATION OF THE PROPOSED METHODS USED IN THE EXPERIMENTS.

Network	Face detection	Classification
CIFAR-10	No	softmax
		SVM
	Yes	softmax
		SVM
AlexNet	No	softmax
		SVM
	Yes	softmax
		SVM
AlexNet Pre-trained	No	softmax
		SVM
	Yes	softmax
		SVM

The proposed method with and without face detection is evaluated to demonstrate the effectiveness of face detection in liveness detection. We employ the cascade classifier using Haar-like features implemented in OpenCV[6] to detect a face region in an input image. The accuracy of liveness detection is evaluated in two cases: (i) the final output of CNN and (ii) the output of SVM with CNN-based features as mentioned in Sec. II-E. We employ linear SVM implemented in scikit-learn[7], which is a machine learning library for Python. Table V shows all the combination of methods evaluating the accuracy in the experiments. In the training of CNN, we set parameters that the number of iteration is 100,000, the learning rate is 0.1, the batch size is 100 and the weight decay λ is 0.004.

The accuracy of liveness detection is evaluated by Half Total Error Rate (HTER) [9], [2] at the subset “test.” HTER is calculated as an average of False Rejection Rate (FRR) and

[6] http://opencv.org/
[7] http://scikit-learn.org/

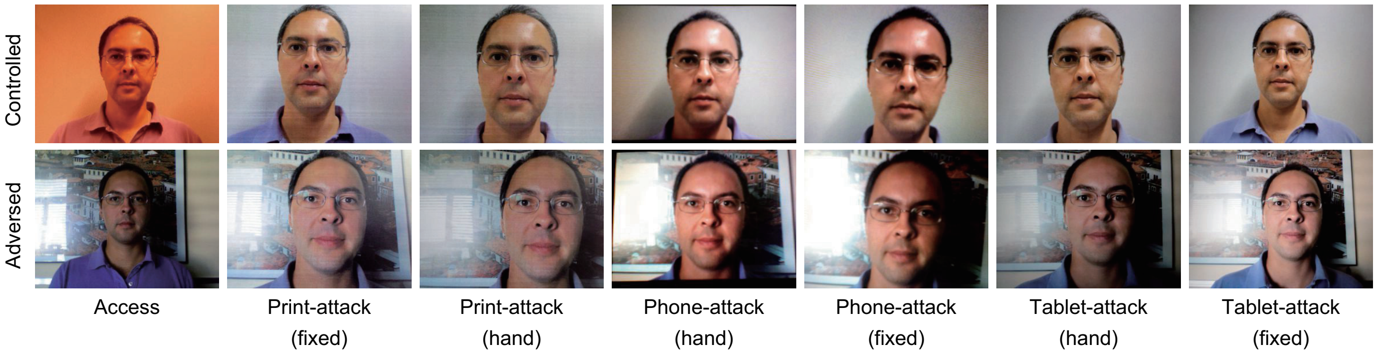


Fig. 4. Example of images in the REPLAY-ATTACK database.

False Acceptance Rate (FAR) as follows:

$$\mathrm{HTER} = \frac{\mathrm{FRR}(\tau) + \mathrm{FAR}(\tau)}{2}, \tag{1}$$

where τ is the threshold determined from the threshold on the Equal Error Rate (EER) at the subset "dev."

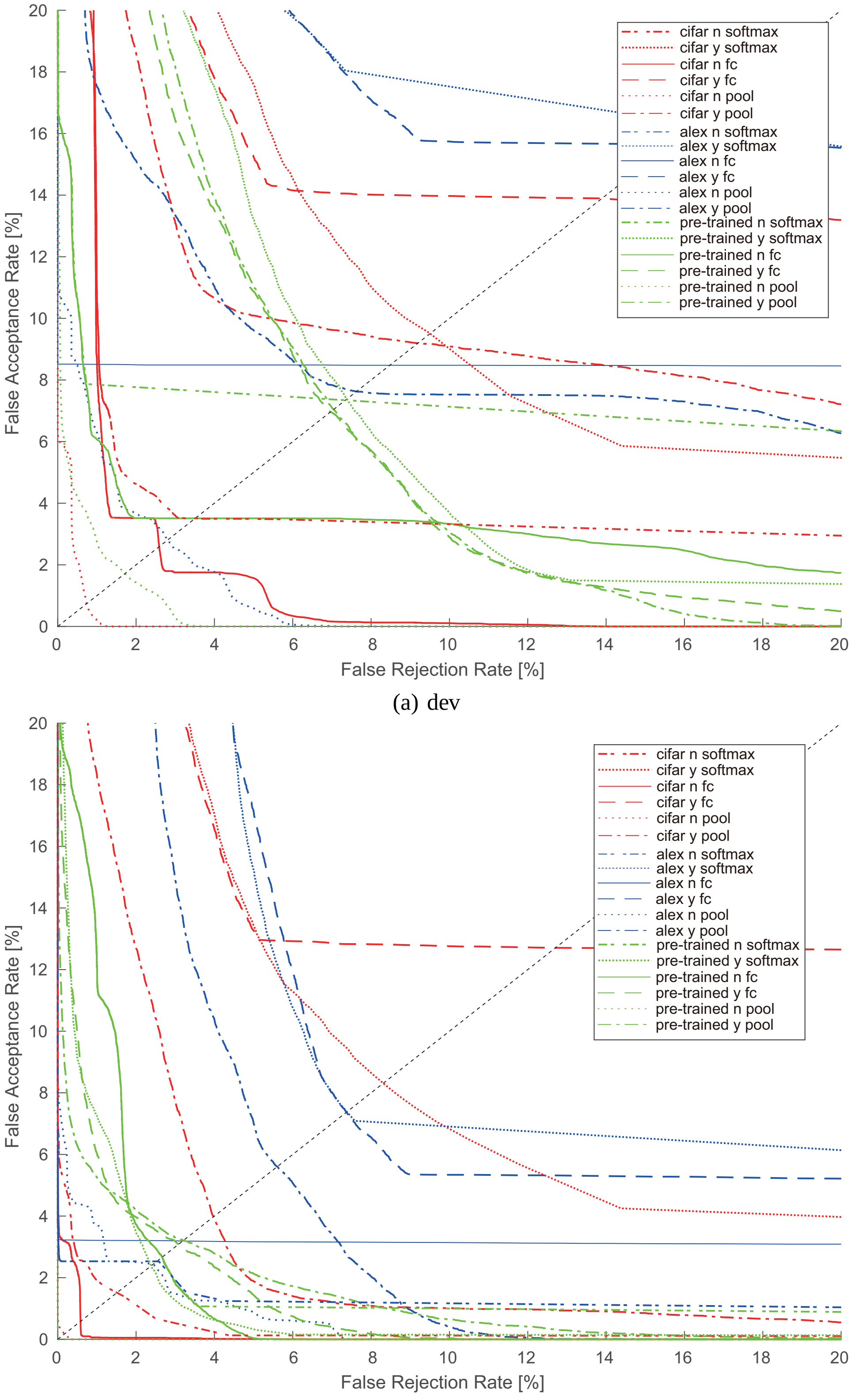
Fig. 5 shows the Receiver Operating Characteristic (ROC) curves for "dev" and "test" and Table VI shows a summary of HTER of the proposed methods. The evaluation protocol of the REPLAY-ATTACK database is that the threshold on the EER at "dev" is used to calculate HTER of methods at "test." In addition, the EER at "test" is also evaluated in the experiments to explore the best performance on "test" for reference. First, the effectiveness of face detection is discussed. EER and HTER when using face detection are higher than others. The results suggest that good features cannot be extracted from a face region to evaluate degradation factors between a real face and a camera. Hence, degradation factors are accurately evaluated by extracting features from the whole image. Next, the effectiveness of CNN-based classification and SVM-based classification with CNN-features is discussed. SVM-based classification exhibits better performance on liveness detection than CNN-based classification. Among SVM-based classification, features from the pooling layer are better than those from the fully-connected layer. The CNN architecture of CIFAR-10 is better than that of AlexNet. Liveness detection does not always require deep CNN architecture, since liveness detection is a 2-class classification problem. AlexNet with pre-trained parameters improves the accuracy of AlexNet. In the case of deep architecture, the best performance is obtained when a large-scale data is used in training. The reason is that the use of pre-trained parameters makes it possible to cover the number of training data.

Fig. 6 shows examples of the failure case in the proposed method. In the case of false acceptance, the quality of printed and displayed fake images is good and such images are also difficult for a human to distinguish real from fake. In the case of false rejection, the quality of real-access images is low due to camera motion and head pose changes. Therefore, the input image with low quality is classified into fake, even if the real-access face is captured by a camera. Such errors described above are caused, since the proposed method evaluates degradation factors included in input images. Addressing this problem, new features for liveness detection, e.g., motion, have to be discussed with further investigation.

Table VII shows a summary of experimental results for conventional and proposed methods using the REPLAY-ATTACK database. Experimental results of the proposed method can be compared with those of conventional methods reported on papers, since the experiments in this paper follow the experiment protocol designated by the REPLAY-ATTACK database. The accuracy of the proposed method is compared with LBP-based methods [2], [11], [5] and CNN-based methods [14], [10], [1] to demonstrate the effectiveness of the proposed method in liveness detection. The error rate of the proposed method is lower than that of LBP-based methods as well as other CNN-based methods. Yang et al. [14] employ the same network architecture from AlexNet, while the differences from the proposed method are that a face region is extracted from an input image and features from the fully-connected layer and nonlinear SVM are used. Menotti et al. [10] employ the simple network architecture of CIFAR-10. The error rate of this method is lower than Yang et al. [14], since this method introduces optimization of architectures and filters. Only a face region is used in liveness detection as well as Yang et al [14]. On the other hand, the proposed method performs training without face detection. Focusing on spoofing in face recognition systems, it may be important to detect face regions in order to distinguish real from fake. In the case of spoofing using printed or displayed photos, it is necessary to evaluate degradation factors included both in a face and background. Therefore, it is important to use the whole image to evaluate liveness of input images.

IV. Conclusion

This paper considered to enhance the security of biometric recognition systems, especially in face recognition. Focusing on spoofing attacks on face recognition systems, we proposed a liveness detection method using Convolutional Neural Network (CNN) inspired from the network architecture of CIFAR-10 and AlexNet. The proposed approach assumed that spoofing is modeled that the input image is degraded by



(a) dev

(b) test

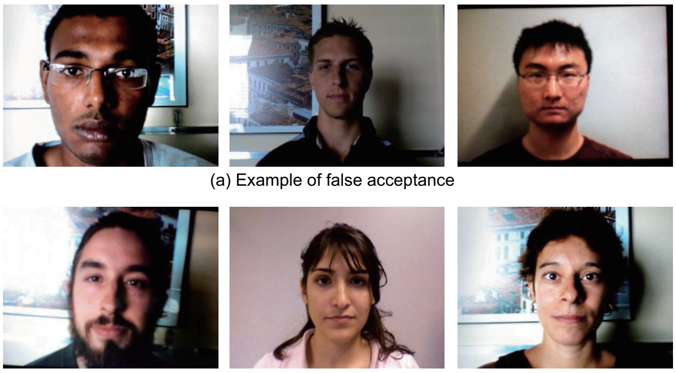
Fig. 5. ROC curves: “cifar” indicates CIFAR-10, “alex” indicates AlexNet and “pre-trained” indicates AlexNet with pre-trained parameters. “n” and “y” indicate the method with and without face detection, respectively. “softmax” indicates CNN-based classification, “fc” indicates SVM-based classification with features from the fully-connected layer and “pool” indicates SVM-based classification with features from the pooling layer.

TABLE VI
SUMMARY OF EXPERIMENTAL RESULTS.

Model	Face detection	Classification	dev (EER [%])	test (HTER [%])	test (EER [%])
CIFAR-10	No	softmax	3.5176	1.5583	1.4551
	Yes	softmax	9.5087	8.4912	8.3104
	No	SVM (pool2)	0.7219	1.0368	0.1762
	Yes	SVM (pool2)	9.1862	4.1385	3.9655
	No	SVM (fc4)	2.5827	1.3470	0.5667
	Yes	SVM (fc4)	13.892	13.796	12.5008
AlexNet	No	softmax	2.4532	2.5549	2.4442
	Yes	softmax	12.6945	7.3343	7.3383
	No	SVM (norm5)	2.8066	2.3828	2.4279
	Yes	SVM (norm5)	7.6259	4.9896	5.6060
	No	SVM (fc7)	7.9901	7.3122	2.9289
	Yes	SVM (fc7)	15.3695	10.9553	7.3709
AlexNet Pre-trained	No	softmax	4.3016	2.3342	2.6474
	Yes	softmax	7.3768	3.3398	2.3784
	No	SVM (pool5)	1.7127	0.4346	0.0006
	Yes	SVM (pool5)	7.0425	3.1891	3.2086
	No	SVM (fc7)	3.5023	4.3552	2.6633
	Yes	SVM (fc7)	6.9603	4.3636	3.0953

TABLE VII
SUMMARY OF EXPERIMENTAL RESULTS FOR CONVENTIONAL AND PROPOSED METHODS: THE VALUE IN THE BRACKET INDICATES THE EER [%] AT "TEST."

	Feature	Face detection	dev (EER [%])	test (HTER [%])
Chigovska et al. [2]	LBP	Yes	19.60	17.17
Pereira et al. [11]	LBPTOP	Yes	8.17	8.51
Kim et al. [5]	LSP	Yes	13.72	12.50
Yang et al. [14]	CNN (AlexNet)	Yes	7.33	2.30
Menotti et al. [10]	CNN (CIFAR-10)	Yes	—	0.75
Alotabib et al. [1]	CNN (Own)	Yes	—	10
Proposed	CNN (CIFAR-10)	No	0.7219	1.0368 (0.1762)
	CNN (AlexNet)	No	2.8066	2.3828 (2.4279)
	CNN (Pre-trained AlexNet)	No	1.7127	0.4346 (0.0006)



(a) Example of false acceptance

(b) Example of false rejection

Fig. 6. Example of failure cases in the proposed method: (a) false acceptance and (b) false rejection.

factors between the real face of the authenticated user and the camera on the face recognition system. Though experiments using the REPLAY-ATTACK database, we demonstrated that the proposed method exhibits efficient performance on liveness detection compared with conventional methods. Accurate liveness detection was realized when using the whole image. SVM-based classification with CNN features exhibited better performance than CNN-based classification. In future, we will explore the effectiveness of the proposed method in other biometric traits such as fingerprint, iris, palm, etc.

REFERENCES

[1] A. Alotaibi and A. Mahmood. Deep face liveness detection based on nonlinear diffusion using convolution neural network. *Signal, Image and Video Processing*, pages 1–8, Nov. 2016.

[2] I. Chingovska, A. Anjos, and S. Marcel. On the effectiveness of local binary patterns in face anti-spoofing. *Proc. Int'l Conf. Biometrics Special Interest Group*, Sept. 2012.

[3] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. The MIT Press, 2016.

[4] A. Jain, P. Flynn, and A. Ross. *Handbook of Biometrics*. Springer, 2008.

[5] W. Kim, S. Suh, and J. Han. Face liveness detection from a single image via diffusion speed model. *IEEE Trans. Image Processing*, 24(8):2456–2465, Aug. 2015.

[6] A. Krizhevsky, I. Sutskever, and G. Hinton. Imagenet classification with deep convolutional neural networks. *Proc. Annual Conf. Neural Information Processing Systems*, pages 1–9, 2012.

[7] Y. LeCun, B. Boser, J. Denker, D. Henderson, R. Howard, W. Hubbard, and L. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1:541–551, 1989.

[8] S. Li and A. Jain. *Handbook of Face Recognition*. Springer, 2011.

[9] S. Marcel, M. Nixon, and S. Li. *Handbook of Biometric Anti-Spoofing*. Springer, 2014.

[10] D. Menotti, G. Chiachia, A. Pinto, W. Schwartz, H. Pedrini, A. Falcão, and A. Rocha. Deep representations for iris, face, and fingerprint spoofing detection. *IEEE Trans. Information Forensic and Security*, 10(4):864–879, Apr. 2015.

[11] T. Pereira, A. Anjos, J. De Martino, and S. Marcel. Can face anti-spoofing countermeasures work in a real world scenario? *Proc. Int'l Conf. Biometrics*, June 2013.

[12] M. Pietikäinen, A. Hadid, G. Zhao, and T. Ahonen. *Computer Vision Using Local Binary Patterns*. Springer, 2011.

[13] A. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson. CNN features off-the-shelf: An astounding baseline for recognition. *Proc. IEEE Conf. Computer Vision and Pattern Recognition Workshops*, pages 512–519, June 2014.

[14] J. Yang, Z. Lei, and S. Li. Learn convolutional neural network for face anti-spoofing. *CoRR*, abs/1408.5601:1–8, 2014.