IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received July 27, 2019, accepted August 8, 2019, date of publication August 22, 2019, date of current version September 6, 2019.

*Digital Object Identifier 10.1109/ACCESS.2019.2936890*

# Fingerprint Liveness Detection by a Template-Probe Convolutional Neural Network

**HO YUB JUNG[1], YONG SEOK HEO[2], AND SOOCHAHN LEE[3]**
[1]Department of Computer Engineering, Chosun University, Gwangju 61452, South Korea
[2]Department of Electrical and Computer Engineering, Ajou University, Suwon 16499, South Korea
[3]School of Electrical Engineering, Kookmin University, Seoul 02707, South Korea

Corresponding author: Soochahn Lee (sclee@kookmin.ac.kr)

This work was supported in part by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science and ICT under Grant 2017R1C1B5074302, and in part by the Research fund from Chosun University, 2017.

**ABSTRACT** Fingerprints are known to be easily synthesized to trick identification systems. In this paper, we propose a new method that incorporates template fingerprints stored for identification in the liveness detection system. The fingerprint identification platform must have a list of template fingerprints stored for matching with new probe fingerprints trying to access the system. Thus, instead of simply detecting the liveness of the probe fingerprints, the proposed approach uses the matching template fingerprints along with probe fingerprints through convolutional neural networks to make the liveness decision, which comprises two sequential convolutional neural networks for classification. The proposed method can be built on the top of existing liveness detection methods to increase accuracy without a significant increase in computation time. The evaluation over the LivDet dataset shows that the proposed fingerprint liveness detection method is able to obtain state-of-the-art accuracy.

**INDEX TERMS** Convolutional neural network, fingerprints, LivDet, liveness detection, pretraining, transfer learning.

## I. INTRODUCTION

Fingerprint identification has been a reliable and convenient security measure for various systems, including smartphones and tablets. However, fingerprints are also known to be synthetically reproducible. Fake fingerprints can be easily made from common materials such as wood glue and Play-Doh-like materials [1], [2]. Even with crude synthetic fingerprints, the identification system can be compromised. Fig. 1 gives examples of a live finger press and a spoof fingerprint made out of wood glue.

**FIGURE 1.** **The left fingerprint is a live press fingerprint, whereas the right fingerprint is obtained by pressing a synthetic finger made out of wood-glue. The two fingerprints came from the same finger and so have matching identification [6] [2]. We aim to detect the liveness of the fingerprint press using a convolutional neural network.**

A fingerprint identification system works by first registering template fingerprints, often storing numerous different presses from the same finger. The access to the system is granted if the probe fingerprints match the template fingerprints. Such systems can be bypassed if a synthetic finger with a matching fingerprint is used to provide the probe fingerprint for the identification system. The matching fingerprints can be theoretically obtained from many different personal objects such as doorknobs or glasses with marked fingerprints. Sometimes employees make synthetic fingers of their own fingers and share them among coworkers to trick the

The associate editor coordinating the review of this article and approving it for publication was Kashif Munir.

system into counting false attendance. Thus, a more secure fingerprint identification system is required both to correctly match fingerprints and to detect spoof fingerprints.

In most previous works, the spoof fingerprint detection problem was separated from the fingerprint matching system. The accuracy of the spoof detection is evaluated by recognizing a probe fingerprint as live or spoof [1], [2]. However, as noted in [3], the intended purpose of spoof detection is to help in determining the identity of the probe fingerprints. The probe fingerprint must be matched to the template fingerprint and also be recognized as a live fingerprint for the system to be accessed. It is thus not only practical but also logical to use the template fingerprints in spoof detection.

In this paper, we present a new method that incorporates fingerprint matching into the problem of liveness detection of probe fingerprints. Our method is based on the assumption that the fingerprints provided when registering a new ID to the system are live. In the proposed system, fingerprint matching is first performed to retrieve the corresponding registered fingerprints. We believe that utilizing the corresponding registered fingerprints as live templates of the probe fingerprint can provide useful information when attempting to detect its liveness.

The proposed approach does not view the liveness detection problem as a straightforward classification problem. Rather, we believe that there are certain textural characteristics in the fingerprint that are found more in the spoof fingerprints and that there are certain characteristics that are found more in the live fingerprints. But such a viewpoint conversely suggests that it is also possible to have live fingerprints with more spoof characteristics and that some spoof fingerprints can have more live characteristics. These spoof-looking live fingerprints or live-looking spoof fingerprints are difficult to correctly recognize by themselves. However, if we know that a particular finger has more spoof or live characteristics based on the registered template, we can make an adjusted decision.

Convolutional neural networks (CNN) have shown to produce accurate liveness classification models without any prior knowledge or study of spoof and liveness features [4] [5]. CNN and deep learning methods are convenient to use because it is not necessary to extract definitive features for classification problems. Through an optimization process, CNN has the ability to adopt to the problem and produce a meaningful solution.

In this paper, we first train a CNN to generate the liveness map of the fingerprint using a network architecture and loss function similar to our previous work [5]. Then, the liveness maps are obtained for template and probe fingerprints. The liveness map of the two fingerprints are stacked and used as the input to another CNN which is trained to predict the liveness of the probe fingerprint. Thus, the second CNN considers the liveness characteristics of the registered fingerprint for making the live or spoof decision on the probe fingerprints.

The additional computational time for the second CNN is minimal because a relatively small liveness map of size $32 \times 32$ is used compared to the $512 \times 512$ fingerprint size for the first CNN for liveness map generation. Furthermore, the liveness maps for the template fingerprints can be precomputed at the time of registration. The stored liveness maps can then be reused for new probe fingerprints with no additional computation time during the identification procedure.

For evaluation and training, we used the LivDet 2015 dataset, which contains test and training sets for four scanners [1]. All training sets were used for training the liveness map CNN. For training the second decision CNN, only the fingerprints that have matching live fingerprints are used to train the network. We note that both probe fingerprints and matching live fingerprints were available for most of the test set. Thus, we were able to evaluate the additional accuracy increase compared to the simple liveness CNN. The comparison with previous methods are evaluated using all the test sets; the liveness map was used to make the final decision if a test fingerprint did not have matching live fingerprints. The proposed method demonstrated the state-of-the-art accuracy.

In the next section, we will discuss some of the previous liveness detection networks in detail. The structure and training parameters for both CNNs will be presented in section III. The LivDet data set and the evaluation are then explained, followed by the conclusion.

## II. RELATED WORK

There has been continual research for discriminating spoof fingerprints from the live fingerprints. Texture analysis has been a common approach for spoof and live classification. Abhyakar and Schuckers used multiresolution texture analysis and local ridge frequencies as inputs to a fuzzy c-means classifier [7]. New descriptors, such as binarized statistical image features, were introduced in a similar fashion to local binary patterns and local phase quantization representations [8]. The ridge texture features were further examined using histograms of invariant features in a similar spirit to histograms of oriented gradients and the scale invariant feature transforms [9]. Instead of trying to find the statistical differences between spoof and live fingerprints, Sequeira and Cardoso aimed to model the live samples and distinguish the fake samples by detecting deviations from the model [10]. A local binary pattern over a Gaussian pyramid is also examined as a liveness feature [11]. Toosi et al. performed a comprehensive analysis on different liveness features and studied the effectiveness of different feature fusion methods [12].

Also important to fingerprint liveness detection research, LivDet.org has been holding international competitions since 2009 with various different datasets [1], [2], [6], [13]–[15]. Numerous approaches have been introduced through the competition. However, deep learning-based approaches have become dominant in recent years. The advantage of the machine learning methods comes from the ease of use, where the features and classifications are trained automatically if there are enough training samples. A CNN has

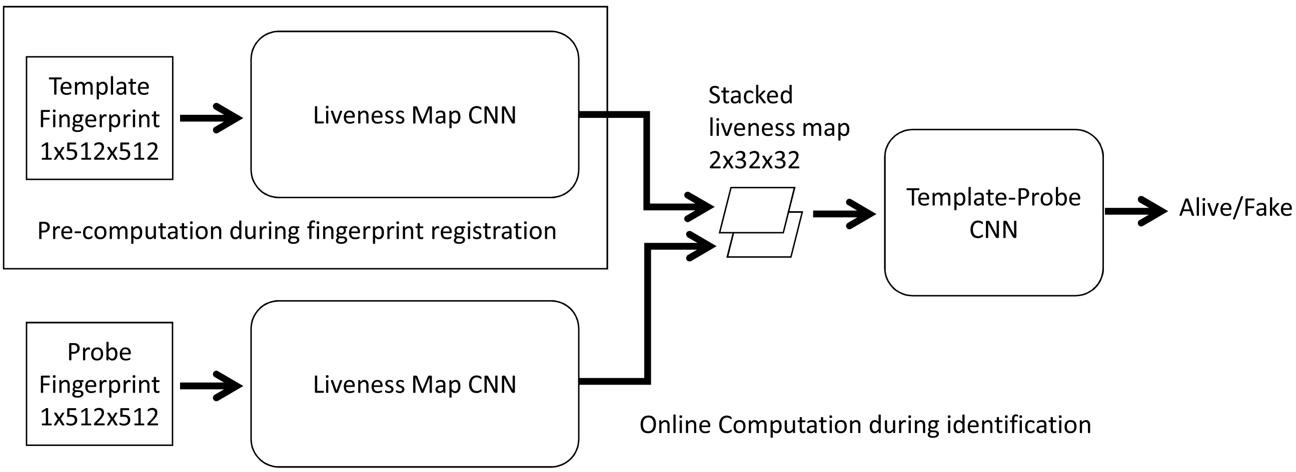



**FIGURE 2.** **Visual description of the overall process of the proposed approach.**

been frequently used as the model, particularly for evaluating images, such as fingerprints.

Nevertheless, deep learning approaches have a few difficulties to overcome when applied to the fingerprint liveness detection problem. First, the number of training samples should be sufficiently large. Second, the computation time can be slow, depending on the design of the network. The deep learning method with aid of a GPU can be very fast, but the liveness detection is often times restricted to embedded systems or CPU-only systems. Furthermore, for applications with smartphones and tablets, a faster authentication is always more desirable.

The deep-CNN was successful in the liveness detection by applying a voting strategy [16]. A deep-CNN was trained for small textural patches of the fingerprint image, and the final liveness decision was obtained by voting counts from the patches from the same fingerprint. The need for large training samples are also addressed by pretraining or transfer learning approaches by Nogueira et al.'s implementation [4]. In their work, the features from AlexNet [17] and VGGNet [18] were used as the pretrained features for the liveness detection CNN. In [19], the CNN trained features are reduced by principal component analysis, and the extracted features are classified by the support vector machine. The Gram matrix from style transfer methods [20], [21] is also used to extract texture features in constructing liveness detection CNN [22].

Although transfer learning is an efficient approach to machine learning problems with a limited number of training samples, it has a disadvantage in that the computation time cannot be smaller than the pretrained networks. Accordingly, it was shown that the fingerprint liveness detection CNN can be trained using only a few thousand samples by defining a training cost over the 32 × 32 liveness map in our previous work [5]. The computation time was also reduced by employing wider strides and applying a separable convolution filter. Although [5] showed reasonable computation time at inference by enabling the resizing of the input image, the trade-off between accuracy and the computation time was unavoidable.

**TABLE 1.** **Liveness map CNN structure. All convolution layers use zero-padding. The filter parameters are denoted by $N \times C \times H \times W$ (number, channel, height, width) scheme.**

| layer | output size | stride | parameters |
|---|---|---|---|
| fingerprint image | $1 \times 512 \times 512$ | | |
| convolution | $1 \times 256 \times 256$ | $2 \times 2$ | $32 \times 1 \times 7 \times 7$ |
| max pool | $1 \times 128 \times 128$ | $2 \times 2$ | pool size $2 \times 2$ |
| relu | $1 \times 128 \times 128$ | $1 \times 1$ | leak 0 |
| convolution | $1 \times 128 \times 128$ | $1 \times 1$ | $32 \times 32 \times 3 \times 3$ |
| max pool | $1 \times 64 \times 64$ | $2 \times 2$ | pool size $2 \times 2$ |
| relu | $1 \times 64 \times 64$ | $1 \times 1$ | leak 0 |
| convolution | $1 \times 64 \times 64$ | $1 \times 1$ | $32 \times 32 \times 3 \times 3$ |
| max pool | $1 \times 32 \times 32$ | $2 \times 2$ | pool size $2 \times 2$ |
| relu | $1 \times 32 \times 32$ | $1 \times 1$ | leak 0 |
| convolution | $1 \times 32 \times 32$ | $1 \times 1$ | $32 \times 32 \times 3 \times 3$ |
| relu | $1 \times 32 \times 32$ | $1 \times 1$ | leak 0 |
| convolution | $1 \times 32 \times 32$ | $1 \times 1$ | $1 \times 32 \times 3 \times 3$ |
| mean square loss | $1 \times 32 \times 32$ | | |

## III. LIVENESS DETECTION USING TEMPLATE-PROBE CNN

The basic idea is to use the liveness maps from both template and probe fingerprints to predict the liveness of the probe fingerprint; see Fig. 2. The CNN for liveness map generation, which we denote as the liveness map CNN (LM-CNN), has been discussed in our previous work [5], with the structure summarized in Table 1. However, we made a few structural modifications as well as changes in the loss function, which improves the accuracy of the LM-CNN. The second CNN, which uses both liveness maps from template and probe fingerprints, is introduced next, where we attempted to design a fast and simple CNN. The second decision making CNN is denoted as the template-probe CNN (TP-CNN).

**FIGURE 3.** Preprocessed live and fake fingerprints shown left and right, respectively. The fingerprints are centered and cropped to 512 × 512 size. The grayscale values are reversed and min/max normalized to [0, 1] interval, but the grayscale shown here is over the [−1, 1] interval in consistency with generated liveness map. These two fingerprints are from the same finger. The live fingerprint is registered as the template fingerprint, and the fake fingerprint can be used to compromise the identification system.

**FIGURE 4.** 32 × 32 liveness map CNN responses are shown for the fingerprints from Fig. 3. The last image on the right is the response from the template-probe CNN using the aforementioned liveness maps. The images are shown in the [−1, 1] interval. The liveness map correctly depicts the live and fake fingerprint regions as well as the neutral background regions. The response map of the template-probe CNN comprises a simple uniform map of −1 values over a 32 × 32 area, correctly detecting the fake fingerprint of the second liveness map in reference to the first liveness map.

### A. LIVENESS MAP CNN (LM-CNN)

The fingerprint scanners used in this paper produce grayscale fingerprint images with a white background and a dark foreground. We reverse the color so that the background is zero black. The zero background is more consistent with the zero-padding procedure of the convolution layers. The grayscale values are rescaled using the min-max normalization, which is a typical preprocess for the deep learning of images. Then, the foreground is brought to the center by finding the weighted average position of the fingerprint foreground. Additionally, since the fingerprint image size varies from scanner to scanner, we crop or zero extend the image to 512 × 512 size. Here, rescaling the image might be detrimental to the process because textural information which can be critical to the liveness detection, may be lost during down or upscaling. Some examples of the resulting images after preprocessing are shown in Fig. 3.

Different scanners capture different features of fingerprints with varying sizes. For higher accuracy, the network must be trained individually for each scanner. Building spoof fingerprints is a time-consuming process, and for the LivDet [1], only approximately 1000 to 1500 spoof fingerprints are available for training for each scanner. The number of live fingerprints is also approximately 1000 to 1500. The training set size is much too small for typical deep learning classification networks, even with typical data augmentation comprising rotation and translation.

However, the main advantage of the LM-CNN is that with larger output map size, fewer fingerprints are required to train the network. The proposed network has a series of convolution, max-pool, and relu [23] layers that outputs a $1 \times 32 \times 32$ channel-height-width (CHW) liveness map. The ground truth is thus also a 1 channel 2D tensor of the same size with all values equal to $+1$ for live and $-1$ for spoof fingerprints. The loss function of the network is a simple square error function. The optimization procedure then tries to minimize the difference at each position of the $1 \times 32 \times 32$ map. Thus, the single output value of $1 \times 32 \times 32$ liveness map is a function of the receptive field of the fingerprint, which in turn is like a sample-wise loss function with receptive field, weights, and ground-truth value as the variables. If we use fully connected layers at the end with cross-entropy as the loss function, we would expect over fitting issues with only 2000 or so samples.

The number of convolution layers is maintained at 5, which is sufficiently deep without causing vanishing gradient issues. The details regarding parameters of the network are shown in Table 1. There are few notable differences from our previous liveness map CNN [5]. We abandon the separable filters because although the computation time is somewhat conserved, the main computation reduction comes from the 2 × 2 strides and 2 × 2 max-pooling in the earlier layers. The relatively large 7 × 7 filter is used in the beginning layer to cover the larger strides. Also since the input has only one channel, having a larger filter in the first layer does not increase the computation time as seriously as having larger filters in the latter layers. We also limited the number of filters to 32, mainly because we wanted to keep computation time within a few seconds in CPU. We used the conventional square loss function instead of evaluating the loss only over the map position, which contains the foreground fingerprint in its receptive field, as we did in [5]. The zero background that is present in both live and spoof images is automatically trained to associate with the neutral zero value in the liveness map. See Fig. 4 for the examples of the liveness map results.

We used the conventional stochastic gradient descent with a batch size of two during the optimization procedure. A smaller batch size is shown to converge at a lower cost value when using stochastic gradient descent. The learning rate has to be adjusted for different CNNs and datasets, in order to avoid possible gradient explosion or vanishing. However, we found that the typical learning rate value 0.01 was sufficient for this problem. A reasonable one or two day training time is allotted for each scanner which translated into around 512 epochs. The Xavier initialization [24] is used. Xavier initialization aims to have the same variance for output as input in a convolution layer, which allows for an initialization such that a reasonable variance can be found for the output of the initial network. In our previous study, we found that data

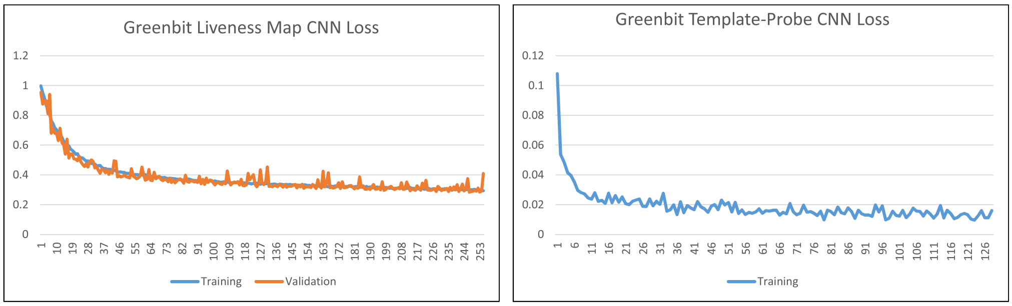



**FIGURE 5.** The mean squared loss is minimized for both LM-CNN and TP-CNN. For TP-CNN, the validation set was not used and the parameters with minimum training loss is chosen as the final network. 512 epochs are trained for LM-CNN in total, but only the first 256 epochs are shown. TP-CNN is trained for around 200 epochs, but 128 epochs are shown.

augmentation by image rotation adversely affects the accuracy of Digital Persona fingerprints which has a lower resolution than the other scanners. Nevertheless, we wanted to train the network so that it can work for both the original non-rotated fingerprints as well as for the augmented fingerprints. Thus, for 50% of the time, data are augmented by random $[-30°, 30°]$ rotation, max 8 pixel translation from center, and Gaussian noise. Smaller translation and rotation ranges are chosen in order to prevent fingerprints from going out of bound, and to maintain the natural orientation of scanned fingerprints. The loss minimization graph is shown in Fig. 5.

### B. TEMPLATE-PROBE CNN (TP-CNN)

The LM-CNN is able to obtain sufficiently accurate results by itself as demonstrated in our previous work. However, we want to improve the result by using the registered template fingerprint, which should already be in the identification system. The liveness map estimation basically indicates whether the receptive field of the fingerprint is likely to belong to a spoof or a live one. An average of a liveness map usually produces correct decisions, but there are outliers where live fingerprints have more spoof features. If we have live template fingerprints for comparison, we should expect improvements in the accuracy.

Multiple presses have been made from the same fingers in the LivDet 2015 dataset. Most of the fingerprints have multiple live presses that can be used as the template fingerprints. The fingerprints without the alternative live presses are not included in the training. The batch samples are produced by first randomly selecting probe fingerprints from the training set, and then randomly selecting from the available template fingerprints for each probe image. The probe and template images are each passed through the LM-CNN, and the resulting liveness maps are used as the input features for the TP-CNN. The ground truth for the network is the liveness of the probe fingerprint.

The input tensor size for the TP-CNN is $2 \times 32 \times 32$ CHW. The first channel is the liveness map of the template fingerprint. The second channel is the liveness map of the probe fingerprint. The TP-CNN is composed of a of series of convolution, max pool, and relu layers. The detailed

**TABLE 2.** Template-probe CNN structure. All convolution layers use zero-padding. The filter parameters are denoted by $N \times C \times H \times W$ (number, channel, height, width) scheme.

| layer | output size | stride | parameters |
|---|---|---|---|
| liveness map image | $2 \times 32 \times 32$ | | |
| convolution | $1 \times 32 \times 32$ | $1 \times 1$ | $32 \times 2 \times 3 \times 3$ |
| max pool | $1 \times 16 \times 16$ | $2 \times 2$ | pool size $2 \times 2$ |
| relu | $1 \times 16 \times 16$ | $1 \times 1$ | leak 0 |
| convolution | $1 \times 16 \times 16$ | $1 \times 1$ | $32 \times 32 \times 3 \times 3$ |
| max pool | $1 \times 8 \times 8$ | $2 \times 2$ | pool size $2 \times 2$ |
| relu | $1 \times 8 \times 8$ | $1 \times 1$ | leak 0 |
| convolution | $1 \times 8 \times 8$ | $1 \times 1$ | $32 \times 32 \times 3 \times 3$ |
| max pool | $1 \times 4 \times 4$ | $2 \times 2$ | pool size $2 \times 2$ |
| relu | $1 \times 4 \times 4$ | $1 \times 1$ | leak 0 |
| convolution | $1 \times 4 \times 4$ | $1 \times 1$ | $32 \times 32 \times 3 \times 3$ |
| relu | $1 \times 4 \times 4$ | $1 \times 1$ | leak 0 |
| convolution | $1 \times 4 \times 4$ | $1 \times 1$ | $1 \times 32 \times 3 \times 3$ |
| mean square loss | $1 \times 4 \times 4$ | | |

description of filters is shown in Table 2. The response tensor has size $1 \times 4 \times 4$, all having $+1$ values for live probe fingerprints and $-1$ for spoofs.

The training parameters are kept similar to the LM-CNN with a learning rate of 0.01. Xavier initialization is used again, and the same rotation, translation, and Gaussian noise data augmentation is performed for both template and probe fingerprints before the LM-CNN. However, the TP-CNN was trained without the validation set for around 200 epochs. Considering the complexity of the network and the available data augmentation through template image selection, we felt that the chances of overfitting were minuscule. Additionally, the 200 epochs is probably excessive, and the minimization is mostly attenuated after 10 epochs. The loss minimization graph is shown in Fig. 5.

## IV. EVALUATION

The datasets from the LivDet 2015 are used for the test. The LivDet 2017 competition is completed; however, the test sets are not available to the public as the majority of 2017 datasets are being used in the 2019 competition.

The 2015 dataset has fingerprints from four scanners: GreenBit, Digital-Persona, Biometrika, and Cross-Match. Examples of each fingerprint scanners are shown in Fig. 6.

**FIGURE 6.** Examples of live fingerprints for Digital Persona, Crossmatch, Biometrika and Greenbit are shown in order from left to right.

**TABLE 3.** Accuracy of liveness map CNN (LM-CNN) and proposed template-probe CNN (TP-CNN) over the LivDet 2015 dataset. In the LivDet 2015 test set, a large portion of the test fingerprints did not have a matching live fingerprint that could be used as the template. In this evaluation, the accuracy was measured for test fingerprints with at least one matching live fingerprint.

| Scanner<br>Image Size<br>Tested Fingerprints / Test Set Size | Digital Persona<br>252 × 324<br>2140/2500 | Crossmatch<br>800 × 750<br>2933/2948 | Biometrika<br>1000 × 1000<br>2140/2500 | Green Bit<br>500 × 500<br>2140/2500 | Avg. Accuracy |
|---|---|---|---|---|---|
| LM-CNN | 90.93 | 99.15 | 96.17 | **97.24** | 95.87 |
| TP-CNN | **93.50** | **99.55** | **97.57** | 96.54 | **96.79** |

**TABLE 4.** Accuracy of proposed methods compared with previous state-of-the-art CNN methods as well as few of the LivDet 2015 competition methods are shown here. All the LivDet 2015 test sets are evaluated. LM-CNNp is the liveness map CNN from our previous work [5]. For TP/LM-CNN, the TP-CNN is used for the test fingerprint with matching live fingerprint; otherwise, LM-CNN is used to classify the liveness.

| Scanner<br>Image Size<br>Matching Fingerprints / Test Set Size | Digital Persona<br>252 × 324<br>2140/2500 | Crossmatch<br>800 × 750<br>2933/2948 | Biometrika<br>1000 × 1000<br>2140/2500 | Green Bit<br>500 × 500<br>2140/2500 | Avg. Accuracy |
|---|---|---|---|---|---|
| hbirkholz [1] | 88.00 | 89.93 | 93.40 | 91.36 | 90.67 |
| titanz [1] | 89.04 | 91.62 | 92.36 | 91.76 | 91.20 |
| jinglian [1] | 88.16 | 94.34 | 94.08 | 94.44 | 92.76 |
| unina [1] | 85.44 | 96.00 | 95.20 | 95.80 | 93.11 |
| CNN-VGG [4] | 93.72 | 98.10 | 94.36 | 95.40 | 95.40 |
| Gram-128-CNN [23] | 91.5 | **99.73** | 95.90 | **98.65** | 96.45 |
| LM-CNNp [5] | 90.50 | 98.60 | 95.80 | 96.20 | 95.28 |
| Proposed LM-CNN | 91.92 | 99.15 | 96.72 | 96.96 | 96.19 |
| Proposed TP/LM-CNN | **94.12** | 99.56 | **97.92** | 96.36 | **96.99** |

Each scanner has a total of 2000 fingerprints for training. Half of the training set is composed of live fingerprints, and the other half is composed of spoof fingerprints synthetically produced with materials such as wood-glue, gelatin, latex, and ecoflex. Each scanner also has a separate test set with approximately 2500 to 3000 fingerprints. The test set has around 1000 live and 1000 spoof fingerprints of same materials as the training set as well as an additional 500 spoof fingerprints made with new materials not found in the training set.

The next subsection will evaluate the incremental accuracy improvement of using the additional TP-CNN. Although the LM-CNN by itself is able to obtain high accuracy, using the additional template fingerprint information which should be readily available in the identification system, allows for consistently higher accuracy. In the second subsection, we evaluate the proposed approach with previous spoof detection methods, which only use the probe fingerprint information.

### A. TEMPLATE-PROBE CNN ACCURACY

Since the ground truth of live and spoof values were set to $+1$ and $-1$, respectively, we average the response of the TP-CNN. The probe fingerprint is classified as spoof if the average is less than zero; otherwise, it is classified as a live fingerprint. The same classification decision scheme is used in the TP-CNN. In this evaluation, the threshold for the decision is set to zero in consistency with the ground-truth definition. However, the decision threshold can be adjusted to add bias toward live or spoof classification depending upon the security demands of an application.

Around 85% of the fingerprints in the test set have at least one live matching fingerprint that can be used as the template fingerprint. For this subsection, we evaluated the fingerprints with templates only. The template fingerprints are chosen randomly from five to ten template fingerprint candidates. Overall, the accuracy using the TP-CNN is shown to be higher than liveness detection using only the probe fingerprint. See Table 3 for the comparison results. The only

**TABLE 5.** The millisecond (ms) computation times using CPU and GPU are shown for LM-CNN and TP-CNN.

| | 3.60 GHz CPU | Nvidia 1080Ti GPU |
|---|---|---|
| LM-CNN | 1696 ms | 24 ms |
| TP-CNN | 16 ms | 1 ms |

accuracy decrease is shown in the scanner Green Bit, where the LM-CNN accuracy of 97.24% is reduced to 96.54%. We observed greatest increase in the accuracy in lower resolution Digital Persona scanner from 90.93% to 93.50%.

The statistical significance tests are calculated for the changes in the accuracies between LM-CNN and TP-CNN. Using the proportion test, the p-value for Digital Persona data is 0.002. The increase of accuracy is highest for Digital Persona set. The Crossmatch set's accuracy is pretty much saturated and showed smallest changes, with p-value of 0.073. Biometrika set's p-value is 0.011. For Green Bit set, the accuracy actually decreased with p-value of 0.218. Combining the all four sets, we have 9353 fingerprints among which 8993 is correctly classified using LM-CNN and 9075 samples correctly classified using TP-CNN. The p-value for all test set is 0.001.

The computation time is shown in Table 5. All four scanner fingerprints are cropped or extended to the same $512 \times 512$ size, making the operating time the same for all scanners. Since the TP-CNN takes the liveness maps of both template and probe fingerprints, we might expect the computation time to be at least doubled. However, in practice, the liveness map can be precomputed for the template fingerprints. Thus, the proposed approach requires only the additional computation time for the TP-CNN which is computed in far shorter milliseconds due to the many times smaller $32 \times 32$ size. An average 24 millisecond (ms) computation time is measured for the LM-CNN, and 1 ms is measured for TP-CNN using an Nvidia 1080TI GPU. The CPU only operation can increase the computation time to 1.696 seconds for the LM-CNN, while the TP-CNN only adds 16 ms.

### *B. COMPARISON TO PREVIOUS METHODS*

The accuracy comparison with previous liveness detection methods is summarized in Table 4. The previous methods were evaluated for all of the test set. However, as previously mentioned, some of the test fingerprints did not contain template live fingerprints. To obtain the accuracy on the entire dataset even for fingerprints without the template fingerprint, we made following adjustments to the proposed approach. First, if the test fingerprint has the matching template fingerprint, the liveness maps of both are determined, and the TP-CNN is used to make the final liveness detection. Second, if the test fingerprint does not have the template fingerprint, the liveness decision is made using only the liveness map CNN result for the test fingerprint. This evaluation approach allowed us to evaluate all the test sets and make a fair comparison with the previous methods.

Additionally for the comparative evaluation, we included the results from the LM-CNN only. The modified version of our previous liveness map approach is able to obtain a very high accuracy compared to previous methods. However, the proposed TP-CNN is shown to have slightly higher accuracy overall.

## V. CONCLUSION

In this paper, we introduced a template-probe CNN (TP-CNN) approach to detect the liveness of fingerprints. We use the template (registered) fingerprints that are already in the identification system to better detect spoof probe (test) fingerprints. The liveness maps are calculated for both template and probe fingerprints, of which the template fingerprints' liveness map can be calculated once during the registration stage. Using the liveness map from both of the fingerprints as the input features, the proposed TP-CNN is able to increase the accuracy of the liveness detection. Due to the relatively small size of the liveness map and TP-CNN, there was only a minuscule increase in the computation time. The accuracy comparison with previous methods also showed favourable results.

## REFERENCES

[1] L. Ghiani, D. A. Yambay, V. Mura, G. L. Marcialis, F. Roli, and S. A. Schuckers, "Review of the fingerprint liveness detection (LivDet) competition series: 2009 to 2015," *Image Vis. Comput.*, vol. 58, pp. 110–128, Feb. 2017.

[2] V. Mura, G. Orrù, R. Casula, A. Sibiriu, G. Loi, P. Tuveri, L. Ghiani, and G. L. Marcialis, "LivDet 2017 fingerprint liveness detection competition 2017," in *Proc. Int. Conf. Biometrics (ICB)*, Feb. 2018, pp. 297–302.

[3] I. Chingovska, A. Anjos, and S. Marcel, "Anti-spoofing in action: Joint operation with a verification system," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2013, pp. 98–104.

[4] R. F. Nogueira, R. de Alencar Lotufo, and R. C. Machado, "Fingerprint liveness detection using convolutional neural networks," *IEEE Trans. Inf. Forensics Security*, vol. 11, no. 6, pp. 1206–1213, Jun. 2016.

[5] H. Y. Jung and Y. S. Heo, "Fingerprint liveness map construction using convolutional neural network," *Electron. Lett.*, vol. 54, no. 9, pp. 564–566, May 2018.

[6] *LivDet—Liveness Detection Competitions*. Accessed: Aug. 2019. [Online]. Available: http://livdet.org

[7] A. Abhyankar and S. Schuckers, "Fingerprint liveness detection using local ridge frequencies and multiresolution texture analysis techniques," in *Proc. Int. Conf. Image Process. (ICIP)*, Oct. 2006, pp. 321–324.

[8] L. Ghiani, A. Hadid, G. L. Marcialis, and F. Roli, "Fingerprint liveness detection using binarized statistical image features," in *Proc. IEEE 6th Int. Conf. Biometrics, Theory, Appl. Syst. (BTAS)*, Sep. 2013, pp. 1–6.

[9] C. Gottschlich, E. Marasco, A. Y. Yang, and B. Cukic, "Fingerprint liveness detection based on histograms of invariant gradients," in *Proc. IEEE Int. Joint Conf. Biometrics*, Sep. 2014, pp. 1–7.

[10] A. F. Sequeira and J. S. Cardoso, "Fingerprint liveness detection in the presence of capable intruders," *Sensors*, vol. 15, no. 6, pp. 14615–14638, Jun. 2015.

[11] Y. Jiang and X. Liu, "Uniform local binary pattern for fingerprint liveness detection in the Gaussian pyramid," *J. Elect. Comput. Eng.*, vol. 2018, Jan. 2018, Art. no. 1539298.

[12] A. Toosi, A. Bottino, S. Cumani, P. Negri, and P. L. Sottile, "Feature fusion for fingerprint liveness detection: A comparative study," *IEEE Access*, vol. 5, pp. 23695–23709, 2017.

[13] G. L. Marcialis, A. Lewicke, B. Tan, P. Coli, D. Grimberg, A. Congiu, A. Tidu, F. Roli, and S. Schuckers, "First international fingerprint liveness detection competition–LivDet 2009," in *Proc. Int. Conf. Image Anal. Process.* Berlin, Germany: Springer, 2009, pp. 12–23.

[14] D. Yambay, L. Ghiani, P. Denti, G. L. Marcialis, F. Roli, and S. Schuckers, ''LivDet 2011—Fingerprint liveness detection competition 2011,'' in *Proc. Int. Conf. Biometrics (ICB)*, Mar. 2012, pp. 208–215.

[15] L. Ghiani, D. Yambay, V. Mura, S. Tocco, G. L. Marcialis, F. Roli, and S. Schuckcrs, ''LivDet 2013 fingerprint liveness detection competition 2013,'' in *Proc. Int. Conf. Biometrics (ICB)*, Jun. 2013, pp. 1–6.

[16] C. Wang, K. Li, Z. Wu, and Q. Zhao, ''A DCNN based fingerprint liveness detection algorithm with voting strategy,'' in *Proc. Chin. Conf. Biometric Recognit.* Springer, 2015, pp. 241–249.

[17] A. Krizhevsky, I. Sutskever, and G. E. Hinton, ''ImageNet classification with deep convolutional neural networks,'' in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[18] K. Simonyan and A. Zisserman, ''Very deep convolutional networks for large-scale image recognition,'' 2014, *arXiv:1409.1556*. [Online]. Available: https://arxiv.org/abs/1409.1556

[19] C. Yuan, X. Li, Q. Wu, J. Li, and X. Sun, ''Fingerprint liveness detection from different fingerprint materials using convolutional neural network and principal component analysis,'' *Comput., Mater. Continua*, vol. 53, no. 4, pp. 357–371, 2017.

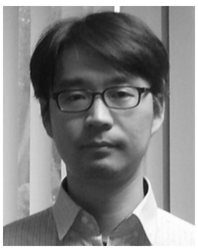
[20] L. A. Gatys, S. A. Ecker, and M. Bethge, ''A neural algorithm of artistic style,'' 2015, *arXiv:1508.06576*. [Online]. Available: https://arxiv.org/abs/1508.06576

[21] L. A. Gatys, M. Bethge, A. Hertzmann, and E. Shechtman, ''Preserving color in neural artistic style transfer,'' 2016, *arXiv:1606.05897*. [Online]. Available: https://arxiv.org/abs/1606.05897

[22] E. Park, X. Cui, W. Kim, and H. Kim, ''End-to-end fingerprints liveness detection using convolutional networks with gram module,'' 2018, *arXiv:1803.07830*. [Online]. Available: https://arxiv.org/abs/1803.07830

[23] V. Nair and E. Geoffrey Hinton, ''Rectified linear units improve restricted boltzmann machines,'' in *Proc. Int. Conf. Mach. Learn.(ICML)*, 2010, pp. 807–814.

[24] X. Glorot and Y. Bengio, ''Understanding the difficulty of training deep feedforward neural networks,'' in *Proc. 13th Int. Conf. Artif. Intell. Statist.*, Mar. 2010, pp. 249–256.

**HO YUB JUNG** received the B.S. degree electrical engineering from The University of Texas at Austin, in 2002, and the M.S. and Ph.D. degrees in electrical engineering and computer science from Seoul National University, in 2006 and 2012, respectively. From 2012 to 2014, he was with the Digital Media and Communications Research and Development Center, Samsung Electronics. From 2014 to 2016, he was a Faculty Member with the Hankuk University of Foreign Studies. Since 2017, he has been an Assistant Professor with the Department of Computer Engineering, Chosun University. His research interests include computer vision, machine learning, and medical imaging.

**YONG SEOK HEO** received the B.S. degree in electrical engineering and the M.S. and the Ph.D. degrees in electrical engineering and computer science from Seoul National University, South Korea, in 2005, 2007, and 2012, respectively. From 2012 to 2014, he was with the Digital Media and Communications Research and Development Center, Samsung Electronics. He is currently with the Department of Electrical and Computer Engineering, Ajou University, as an Associate Professor. His research interests include segmentation, stereo matching, 3D reconstruction, and computational photography.

**SOOCHAHN LEE** received the B.S. and Ph.D. degrees from Seoul National University, Seoul, South Korea, in 2004 and 2011, respectively. From 2011 to 2014, he was with the Digital Media and Communications Research and Development Center, Samsung Electronics. From 2014 to 2019, he was an Assistant Professor with Soonchunhyang University, Asan, South Korea. Since 2019, he has been an Assistant Professor with the School of Electrical Engineering, Kookmin University, Seoul. His research interests include computer vision and medical image analysis.