Two-tiered facial verification for mobile devices

Rafael Padilha, Fernanda A. Andaló, Ricardo da S. Torres, Anderson Rocha and Jacques Wainer
Institute of Computing, University of Campinas (UNICAMP)
{rafael.padilha, feandalo, rtorres, anderson.rocha, wainer}@ic.unicamp.br

***Abstract*—Mobile devices had their popularity and affordability greatly increased in recent years. As a consequence of their ubiquity, these devices now carry all sorts of personal data that should be accessed only by their owner. Even though knowledge-based procedures are still the main methods to secure the owner's identity, recently biometric traits have been employed for a more secure and effortless authentication. In this work,[1] we propose a facial verification method optimized to the mobile environment. It consists of a two-tiered procedure that combines hand-crafted features and a new convolutional neural network – HF-CNN –, an architecture tweaked for mobile devices that processes encoded information of a pair of face images. We also propose a technique to adapt our method's acceptance cutoff to images with different characteristics than those present during training, by using the device owner's enrollment gallery. The proposed solution outperforms state-of-the-art face verification methods, while having a model 16 times smaller and 4 times faster when processing an image in recent smartphone models. Finally, we present a new dataset of selfie pictures – *RCD selfie dataset* – that hopefully will support future research in this scenario.**

I. Introduction

The need to secure one's identity is present in a variety of everyday activities [1], such as allowing or denying access to a requested service, a place, or sensitive information. Several systems use biometric traits to secure the identity of an individual [1]. Among the different biometric modalities, automated face recognition is a very important one. During the past years, it has gained more attention with improvements regarding quality, affordability, and ubiquity of image-capturing devices (surveillance cameras, mobile phone cameras), the many possible commercial uses, and the huge amount of images available online.

With mobile and wearable devices becoming cheaper and more popular, face authentication systems are being integrated into them. And although the capabilities of such devices have been growing in past years, it is necessary to bear in mind their limitations [2] when designing such systems. They have limited processing power that may not be sufficient to run many complex vision and pattern recognition algorithms, as well as a small memory space that may not be suitable to store several face images or features with high dimensionality.

In this work, we propose a two-tiered mobile face verification method: the first tier is fast, lightweight, and specifically tailored to the target user in order to attain high true positive rate; the second tier aims to eliminate false negatives by running a memory-efficient and fast CNN, optimized to distinguish identities by their latent characteristics. In the most common use case – the device user seeking authentication – the first tier alone can handle the authentication, saving up on computation and energy. Otherwise, the second tier can be triggered to confirm or reject the authentication attempt. Alongside with a deep learning approach, we use traditional techniques as complementary tools to extract specific information of the device owner.

Deep learning has been the state of the art for face recognition, surpassing traditional feature-engineered methods. However, complex deep networks are often computationally expensive, while traditional methods tend to be fast and memory-efficient. We capture their best characteristics, by showing how their fusion can lead to better accuracy. In addition to that, we introduce a new CNN architecture optimized with hybrid-input image representations, allowing it to learn whether a pair of face pictures share the same identity of not, i.e., the original face recognition problem is mapped onto a pairwise verification problem.

Another important aspect is that face verification is highly influenced by demographics [3], which often leads to a scenario where multiple models, trained on different demographics, are necessary. We instead present how to automatically select a decision cutoff to better adapt our method to unique characteristics of the users, eliminating the need to consider each race/ethnicity separately.

Finally, we present *RECOD Selfie Dataset*,[2] a public dataset collected during this research, composed of self-portrait (selfie) pictures with different acquisition conditions regarding illumination and head pose. The dataset comprises 56 identities and 2873 images, organized in $262,164$ pairs.

The remainder of this article is organized as follows. Section II outlines face verification methods in literature. Section III presents our two-tiered method, while Section IV details the proposed CNN architecture. Section V describes the datasets used in the experiments, highlighting the one specifically constructed for this work. Section VI presents the experimental results and Section VII summarizes and concludes this work.

II. Related Work

Traditional facial verification techniques were based on hand-crafted features designed using domain knowledge of the data to create representations of face images. Numerous features have been proposed to represent a face in different ways,

[1]This work relates to a MSc. thesis defended in September 1st 2017, with updated results.

[2]To be published soon in http://dx.doi.org/10.6084/m9.figshare.5427142

such as by capturing geometrical information [4], holistic characteristics [5], local appearance of facial traits [6] and texture features [7]. In our solution, we consider two hand-crafted methods: Histogram of Oriented Gradients (HOG) [8] and Local Region Principal Component Analysis (LRPCA) [9].

Data-driven methods, such as CNNs, differ from feature-engineered ones by introducing the ability to build complex concepts out of simpler ones, without depending on domain-knowledge modeling. Fuelled by recent data availability, complex deep architectures started to achieve lower error rates than engineered descriptors [10] on image recognition tasks.

For face verification, we can highlight three CNNs. DeepFace [11] is an architecture with locally-connected layers, allowing the network to learn distinct features for different spatial positions. FaceNet [12] learns a direct embedding from face images to a low-dimensional Euclidean space by triplet loss optimization. VGGFace [13] is a 16-layer CNN with 140 million parameters, trained for face verification with 2.6 million pictures using a triplet-loss approach similar to [12]. VGGFace is the basis for our architecture (Section IV).

Despite achieving impressive results, these networks are not suitable to mobile devices due to their high number of parameters and operations performed. With this in mind, several approaches were designed to simplify and/or speed-up existing architectures. A popular strategy is to compress and prune a CNN model in a lossy process, decreasing model size while trying to maintain accuracy [14]–[17]. Unfortunately, these approaches are not always supported by current deep learning frameworks or may even require specialized hardware [16].

Another line of research aims at designing compact and efficient network architectures, already tweaked to the limitations of low-powered devices.

Recently, MobileNet [18] was designed as a flexible architecture with two hyperparameters that limit the width and spatial resolution of each layer. The authors analyze how these hyperparameters affect accuracy, the number of parameters, and the number of multiply-add operations performed; and apply different network setups in several recognition problems.

A CNN relevant to our work is *SqueezeNet* [19]. To constrain the number of parameters, the authors propose to replace most of its 3×3 filters with 1×1 and decrease the number of channels of the input map. Each block of the SqueezeNet, named as Fire module, consists of two convolutional layers: a *squeeze* layer with 1×1 convolutional filters and an *expand* layer with both 1×1 and 3×3 convolutional filters. In addition to stacked Fire modules, SqueezeNet replaced traditional fully-connected layers with global average pooling, which impels correspondence between each class and the feature maps of the last few convolutional layers without the parameter overhead of fully-connected layers.

III. Two-tiered Face Verification

Our face verification method consists of a two-tiered solution tailored to the mobile environment (Figure 1). The first tier attains high true positive rate, favoring the common case

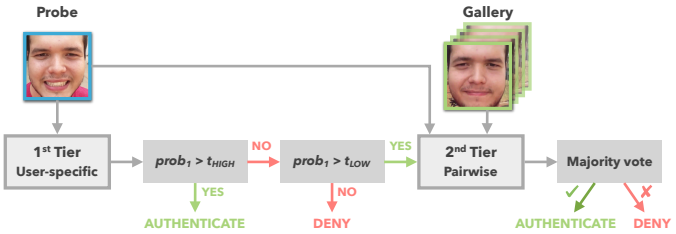


Fig. 1. Outline of the proposed 2-tiered solution for face verification. The 1st tier consists of a set of fast user-specific classifiers, while the 2nd tier uses a fusion of deep and hand-crafted features to identify if a pair of images belongs to the same identity (pairwise verification).

(the owner seeking authentication). If the first tier's confidence is low, the second tier is triggered for further verification.

In the first tier, we use a set of user-specific classifiers to identify the owner in self-portrait pictures. These classifiers are trained using two hand-crafted features: HOG and LRPCA. Since both features are fast to extract and the classifiers are trained with gallery images (i.e., pictures depicting the target user obtained during an enrollment phase), the method attains high true positive rate without consuming too much computational time.

As most authentication attempts are made by the device owner, our method must be fast and accurate for these frequent cases; however, it is acceptable to take more time to deny an intruder. We translate these ideas by having a fast first tier, whose confidence score is tested against two thresholds: t_{low} and t_{high}. In case the score surpasses t_{high}, the user is automatically authenticated; if the score is in between the thresholds, the solution follows to the second tier; whereas if the score is below t_{low}, access is denied.

The second tier consists of a group of *pairwise classifiers*, trained to assess if a pair of faces belongs to the same identity or not. We use two classifiers trained on HOG and LRPCA features separately, considering pairs of images; and also a CNN trained with *hybrid images*, of which the input channels represent a pair of people.

To better adapt to each user's unique face characteristics, we propose a method to automatically learn the decision cutoff of the second tier on-the-fly, based on the gallery images.

A. Enrollment and Pre-processing

During enrollment, a gallery $G = \{g_1, \ldots, g_n\}$ of n selfies from the device owner are acquired with the mobile front camera, in different *views*, i.e., similar photos with small variations in head pose, facial expression, and illumination conditions. Similarly, a negative gallery $O = \{o_1, \ldots, o_n\}$ composed of n face images depicting other people rather than the device owner is embedded into the device beforehand. Each image is converted to grayscale, cropped and aligned so that the distance of the center of the eyes to the image boundaries is the same for all enrollment images [9].

During verification, a selfie from the person seeking authentication is acquired and normalized in the same way, generating a probe image p.

B. First tier: user-specific verification

According to [20], it is possible to improve facial recognition by learning specific user characteristics. In order to do this, we use the gallery G as *positive examples*, while the negative gallery O is used as *negative examples*.

All images are scaled to 128×128 pixels, and HOG and LRPCA features are extracted from them. We train two Logistic Regression (LogReg) classifiers on top of HOG and LRPCA features separately.

During test, probe p has its features extracted and tested against the LogReg models trained with HOG and LRPCA features, resulting in two probabilities $prob_1^{HOG}$ and $prob_1^{PCA}$, respectively, which are averaged to produce the final score $prob_1$ for this tier.

C. Second tier: pairwise verification

The next step in our method seeks to determine if two images belong to the same individual. By slightly changing our target problem – from user-specific to pairwise verification – we wish to capture complementary characteristics that, along with the first tier, improve overall accuracy.

Since the training process does not require pictures of the device owner, it can be performed outside the mobile device, allowing more complex and powerful models. This tier considers both hand-crafted and data-driven approaches:

1) Multiview hand-crafted classifiers: During authentication, we leverage from multiple views of the gallery by comparing a probe to as many gallery images as possible, in what we refer to as *multiview* approach. As the gallery increases in size and diversity, our method will have more information to authenticate. Considering this, we build pairs of images, consisting of the probe p and each gallery image $g_i \in G$, and extract features for these pairs.

To construct a feature vector $F_{pair}(a, b)$ for a pair of face images a and b, feature vectors $F(a)$ and $F(b)$ are first extracted for each image of the pair. As a way to combine both feature vectors and emphasize the relationship between them, we concatenate the modulus of the difference and the element-wise product of $F(a)$ and $F(b)$ [21].

During training, considering a dataset of face images D, we compute a set of pair feature vectors $S_{train} = \{F_{pair}(x, y) | x, y \in D, x \neq y\}$. If images x and y depict the same person, then $F_{pair}(x, y)$ is labeled as *positive*, and *negative* otherwise.

The set S_{train} of pair feature vectors is used as input to a LogReg classifier, to learn a model able to predict the probability of the input being *positive*.

During test, we compute $F_{pair}(p, g_i)$ between probe p and each gallery image $g_i \in G$, $i = 1, \ldots, n$. Each pair feature vector is tested against the LogReg model, yielding a probability $prob_{g_i}$ of the respective pair of images p and g_i depicting the same person. The final *multiview* probability is the average of all $prob_{g_i}$.

We train a LogReg model on top of HOG features and another one on top of LRPCA features. In an authentication attempt, a *multiview* probability will be yielded for each type of feature – referred to as $prob_2^{HOG}$ and $prob_2^{PCA}$ – which are combined with the output from our data-driven classifier to compose the 2nd tier final score.

2) Data-driven classifier: We propose a novel convolutional neural network architecture – referred to as Hybrid-Fire CNN, or HF-CNN – which is described in more details in Section IV. This data-driven approach yields the probability $prob_2^{HF}$ of an input hybrid image representing the same person.

3) 2nd tier fusion and decision: Each pairwise classifier of the 2nd tier has a decision threshold associated with it. In case the classifier's score is above its threshold t, the pair is considered to depict the same person in both images.

Considering the three classifier outputs – $prob_2^{HOG}$, $prob_2^{PCA}$, and $prob_2^{HF}$ – we use majority vote to determine if an authentication is successful or not. However, defining each t below which the authentication fails is not a trivial task. Therefore, we propose a technique to adapt the 2nd tier methods to the owner's unique characteristics.

Given gallery G from a specific user and negative gallery O, we create two equal-size sets of tuples, P and N. Each *positive tuple* in P is created by randomly sampling without replacement l images from G, with $l < n$, whereas a *negative tuple* in N comprises one image from O and $l - 1$ random images from G. Within each tuple, the first image is considered as the probe and the rest as the gallery. For each classifier of the 2nd tier, we input each tuple, register their probability and perform a grid search on the decision cutoff value t, for $0 < t < 1$, to maximize, for that particular user, a desired metric (e.g., accuracy, TPR and TNR).

IV. HF-CNN and Hybrid Images

Our initial explorations with a data-driven method were done with VGGFace network [13]. Despite its impressive results in facial recognition, it is not suitable for the mobile scenario due to its approximately 15 million multiply-add operations and 134 million parameters. Aiming at reducing both, we made several architectural adjustments.

We removed the last six convolutional and fully-connected layers (*conv5-1* to *FC-8*). Although responsible for only 10% of multiply-add operations, they account for 95% of the total parameters. Besides that, they are responsible for learning most high-level concepts related to the target identities of VGGFace, which are not appropriate for our task of identity-independent face verification.

We replaced these layers with 8 Fire modules [19] with 64/256/256 filters in their *squeeze1x1 / expand1x1 / expand3x3* layers. After the last Fire, a convolutional layer with 1×1 filters, followed by a global average pooling and softmax activation outputs the probability $prob_2^{HF}$.

As we are now dealing with a binary problem (face verification), we still need to modify the expected input of the network so that it represents the probe and gallery images. For this, we propose the concept of *hybrid image*, which combines information regarding the probe and the gallery. We aimed for a representation to capture the most relevant characteristics from gallery face images while also attenuating small

variations present on them (e.g., facial expression, makeup, hairstyles). In this vein, for a probe p and a gallery G, a hybrid image is constructed by stacking, as two channels, p and the average image $\overline{g}$ of the images in G.

The proposed CNN, which is referred to as Hybrid-Fire CNN, or HF-CNN, is trained [3] with hybrid images. Differently from VGGFace, which works with RGB images, HF-CNN accepts only two-channel images. To account for this difference, *conv1_1* layer was modified accordingly, while also mapping the shape of its output feature maps to the expected input of *conv1_2*. This allows us to adapt the pre-trained weights of VGGFace to detect low-level patterns in hybrid images.

To reduce the number of operations performed by HF-CNN, rather than directly altering the network architecture and discarding the pre-trained weights, we feed smaller hybrid images to our network. By reducing the image dimensions by half (from 224×224 to 112×112), the internal maps also shrink by the same ratio, thus decreasing the amount of performed multiply-add operations.

V. Datasets

In this work, we present the *RECOD Selfie Dataset* (RCD), a public dataset[4] created for this research, with videos from 56 identities recording self-portrait videos of approximately 30 seconds, using mobile front cameras. For each identity, two videos were recorded: one indoors with artificial light and the other outdoors with direct sunlight and occasional cloud shadows. Each participant was instructed to slowly rotate around his/her own axis during the capture, further increasing variability from one frame to another. While rotating, the person could act naturally, with spontaneous face expressions, and moderately changing head pose and the angle of the phone in relation to the face. Most of the videos were recorded with 1080×1920 resolution, while a minority has 480×640.

Besides RCD, we also consider *Unicamp Video-Based Attack Database* [22] (UVAD), *Oulu-NPU database* [23], [24] (OULU) and *Motorola Selfie Dataset* (MOT)[5]. Examples of images from each dataset are presented in Figure 2.

Combining all datasets, we end up with 564 identities, totaling $27,817$ images in a wide range of illumination, background, hairstyle, facial pose, and expression. We built pairs with pictures pertaining to the same identity (*positive pairs*) and the same number of randomly selected pairs of images from distinct identities (*negative pairs*); totaling 5.5+ million pairs for analysis.

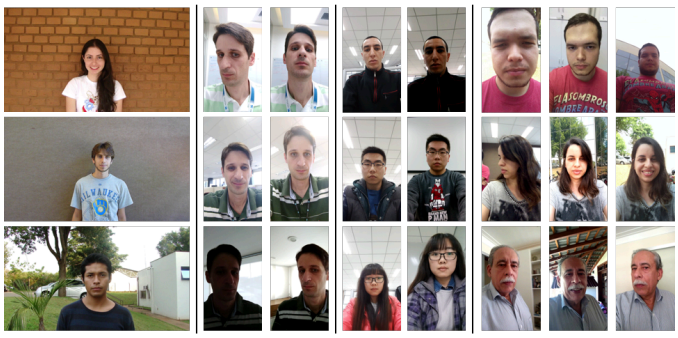
VI. Experimental Results

The proposed 2-tiered verification method is a combination of a series of complementary techniques. In this section, we assess the impact of the individual components, not only to evaluate the performance of the solution, but also as a way to examine the possibility of integrating them into other methods.

[3]Trained on Caffe framework. http://caffe.berkeleyvision.org

[4]Link: http://dx.doi.org/10.6084/m9.figshare.5427142.

[5]Private dataset, created in cooperation with Motorola LLC, consisting of videos from 49 identities, captured in the same setup of RCD.



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

Fig. 2. Samples from **(a)** UVAD, **(b)** MOT, **(c)** OULU, and **(d)** RCD datasets.

A. Experimental setup

We organized the datasets into random identity-disjoint train, validation, and test sets. MOT, UVAD and OULU-Train (20 identities of OULU) were used for training; RCD-Validation (14 identities of RCD) was used for validation; RCD-Test and OULU-Test (remaining 42 identities from RCD and 15 from OULU, respectively) were used as test sets.

We randomly sampled a negative gallery O from the images of MOT, UVAD, and OULU-Train. For each constructed pair of RCD-Test and OULU-Test, we consider the first image of the pair as probe p and the person depicted in the second image as the user. We randomly sampled images of the user from RCD-Test and OULU-Test to construct gallery G. Both G and O have 10 images. Before training, pairs from RCD-Validation were used to find the best hyperparameters.

B. 2-tiered solution experiments

Table I presents results for each tier alone, their individual techniques and the whole 2-tiered solution. In this experiment, we arbitrarily fixed $t_{high} = 0.7$ and $t_{low} = 0.5$.

Note the impact of each tier in the complete solution. The 1st tier achieves a higher true positive rate, when compared to the 2nd tier, i.e., 1st tier is tailored to solve the common case – the device owner seeking authentication. On the other hand, the 2nd tier achieves a higher true negative rate, being capable of eliminating 1st tier's false negatives. This is especially evident when analyzing the performance on OULU-Test.

While we used fixed t_{high} and t_{low} in the last experiment, these thresholds provide a simple way to balance the trade-off between speed, TPR, and TNR. For example, by increasing t_{high} it is possible to be stricter when the 1st tier authenticates a probe. This increases false rejection (by lowering TPR), but decreases false acceptance (by increasing TNR).

Besides security, there is also an efficiency aspect related to the selection of these thresholds. They control how many images are sent to the 2nd tier and how many are authenticated or denied by the 1st tier, which directly relates to the overall speed of the solution. In a modern smartphone, feature extraction using HOG and LRPCA takes less than 1 ms, while forwarding an image through HF-CNN takes approximately 1 s. Consequently, it is faster to have most attempts solved by

TABLE I

PERFORMANCE OF THE COMPLETE 2-TIERED VERIFICATION METHOD, 1ST AND 2ND TIER SEPARATELY, AS WELL AS EACH OF THEIR INNER METHODS.

	RCD-Test		OULU-Test	
Method	TPR %	TNR %	TPR %	TNR %
US-HOG	96.1	97.4	99.8	77.0
US-LRPCA	93.4	94.9	99.8	82.6
1st tier only	96.1	97.9	99.8	82.5
Pairwise HOG	85.6	94.5	81.5	90.2
Pairwise LRPCA	95.4	89.0	99.9	87.9
HF-CNN	91.0	97.4	96.6	94.4
2nd tier only	93.7	97.6	96.9	94.6
2-tiered method	94.1	99.5	99.7	94.5

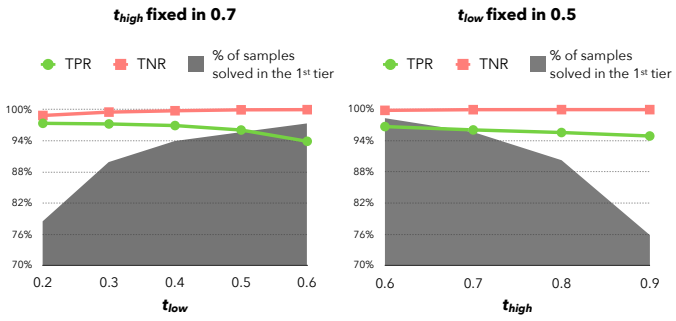


Fig. 3. Threshold (t_{high} and t_{low}) selection exploration for the complete 2-tiered method in RCD-Validation.

the 1st tier, while the next tier only processes those near the 1st tier's decision frontier. Figure 3 presents some threshold setups and the corresponding results for the whole solution in RCD-Validation. We also show the percentage of samples processed by each tier.

For the 2nd tier, the decision cutoff learning was proposed as a way to adapt the methods to images with different characteristics than the ones present during training, while also incorporating information about the device owner. For these experiments, the 2-tiered solution was evaluated in a cross-dataset scenario. Training was done with MOT and UVAD, keeping OULU-Train out, while RCD-Test and OULU-Test were used for testing. For the decision cutoff learning, both P and N sets have 100 tuples, with $l = 7$, and we selected the cutoff value of each user that maximized accuracy, maintaining $TPR > 0.9$ and $TNR > 0.99$.

Table II shows each tier's components separately and their fusion to achieve the tier performance, with and without the decision cutoff learning. When not using OULU for training, HOG and LRPCA are considerably affected. However, cutoff learning is able to improve the complete solution for OULU-Test. For RCD, the cutoff learning has negatively impacted individual methods, but this was lessened by the fusion of all components and the combination with the 1st tier.

TABLE II

METHODS WITH AND WITHOUT DECISION CUTOFF LEARNING.

		RCD-Test		OULU-Test	
Cutoff Learning	**Method**	TPR %	TNR %	TPR %	TNR %
✗	pairwise classifier w/ HOG features	84.3	96.3	93.8	62.4
	pairwise classifier w/ LRPCA features	95.9	88.8	100.0	27.3
	HF-CNN	93.9	95.5	97.2	72.0
	2nd tier only	94.3	97.5	98.2	58.8
	2-tiered method	94.3	99.4	99.7	85.6
✓	pairwise classifier w/ HOG features	86.7	93.1	98.5	91.3
	pairwise classifier w/ LRPCA features	88.9	94.5	99.3	87.8
	HF-CNN	92.5	96.6	97.2	84.9
	2nd tier only	92.4	97.9	99.5	92.4
	2-tiered method	93.3	99.3	99.7	94.0

C. Comparison with existing methods

Several methods in the literature have approached the facial recognition task. However, only few have focused on the mobile environment, where it is necessary to ponder other factors besides accuracy. For this comparison, we have considered:

- **VGGFace:** The output of layer *FC-7* is used as feature vector for an image. The pair feature vector consists of the concatenation of absolute difference and element-wise multiplication of the feature vectors of individual images. A LogReg trained with such pair of feature vectors determines the verification outcome.
- **Fine-tuned VGGFace:** VGGFace fine-tuned with hybrid images, using the same protocol of HF-CNN.
- **ResFace101**: ResNet-101 network fine-tuned for face recognition with CASIA [25], following the data augmentation described in [26]. For the verification task, we considered the same steps performed for VGGFace.
- **Fine-tuned SqueezeNet**: SqueezeNet [19] fine-tuned for face verification. We used the same method described in [27], as well as the provided model and weights.

Table III presents results for the considered methods. Our method outperforms or compares to the other solutions.

We also compared state-of-the-art CNNs with HF-CNN. Table IV presents the analysis regarding number of operations and parameters, and time and memory consumption. In comparison with VGGFace, we have significantly reduced the number of parameters and performed operations. HF-CNN is still behind some architectures tweaked for efficiency, such as MobileNet and SqueezeNet. however they were not proposed for face verification.

VII. CONCLUSION

In this work, we have proposed a 2-tiered method for facial verification optimized for the mobile environment.

TABLE III
COMPARISON OF THE PROPOSED 2-TIERED METHOD WITH METHODS PROPOSED FOR FACE RECOGNITION IN THE LITERATURE.

	RCD-Test			OULU-Test		
Method	$ACC\ \%$	$TPR\ \%$	$TNR\ \%$	$ACC\ \%$	$TPR\ \%$	$TNR\ \%$
2-tiered method	96.8	94.1	99.5	97.1	99.7	94.5
VGGFace	96.9	97.0	96.9	89.0	88.1	89.8
Fine-tuned VGGFace	92.8	87.7	97.9	94.8	91.7	97.9
ResFace101	93.8	95.6	91.9	91.2	90.7	91.6
Fine-tuned SqueezeNet	72.1	72.8	71.4	67.5	67.5	67.5

TABLE IV
COMPUTATIONAL TIME AND MEMORY ANALYSIS FOR CNNS IN *Device A* (MOTOROLA MOTO G5, 2GB RAM, ANDROID 7.0) AND *Device B* (MOTOROLA MOTO Z, 3GB RAM, ANDROID 7.1.1).

		Forward pass time (s)			
Architecture	**Million multiply-add**	***Dev A***	***Dev B***	**Thousand parameters**	**Model size (MB)**
HF-CNN	$3,572$	1.15	0.80	$9,208$	35
VGGFace	$15,468$	10.27	3.03	$134,263$	553
ResFace101	$7,610$	4.31	2.35	$64,060$	256.7
MobileNet	574	1.08	0.34	$4,230$	16
SqueezeNet	388	0.23	0.21	$1,230$	4.7
GoogLeNet	$1,600$	0.97	0.89	$6,990$	51

The proposed Hybrid-Fire CNN (HF-CNN), inspired by VGGFace [13] and SqueezeNet [19], was able to outperform VGGFace, but with a model 16 times smaller and 4 times faster. HF-CNN uses hybrid images to combine the information of a probe and gallery images, as a way to limit the necessity of multiple forward passes. Hybrid images can also be viewed, parallel to siamese networks, as a simple way to adapt a multiclass formulation such as face identification to a binary verification formulation.

In addition, we have also collected a new dataset of selfie pictures, with varying capture conditions regarding illumination, head pose, background, and facial expression.

This work resulted in a USPTO patent application, filed on March 12, 2018

ACKNOWLEDGMENT

The authors would like to thank Motorola LLC for the financial support.

REFERENCES

[1] A. K. Jain, R. Bolle, and S. Pankanti, *Biometrics: personal identification in networked society*. Springer US, 2006.

[2] K. Choi, K.-A. Toh, and H. Byun, "Realtime training on mobile devices for face recognition applications," *Pattern Recognition*, vol. 44, no. 2, pp. 386–400, 2011.

[3] B. F. Klare, M. J. Burge, J. C. Klontz, R. W. V. Bruegge, and A. K. Jain, "Face recognition performance: Role of demographic information," *IEEE Trans. Inf. Forensics Security*, vol. 7, no. 6, pp. 1789–1801, 2012.

[4] T. Kanade, "Picture processing system by computer complex and recognition of human faces," Ph.D. dissertation, Kyoto University, 1973.

[5] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 7, pp. 711–720, 1997.

[6] L. Wiskott, J.-M. Fellous, N. Kuiger, and C. V. D. Malsburg, "Face recognition by elastic bunch graph matching," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 7, pp. 775–779, 1997.

[7] T. Ahonen, A. Hadid, and M. Pietikainen, "Face description with local binary patterns: Application to face recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 12, pp. 2037–2041, 2006.

[8] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *IEEE Conf. Comput. Vision Pattern Recognition*, 2005, pp. 886–893.

[9] P. J. Phillips, J. R. Beveridge, B. A. Draper, G. Givens, A. J. O'Toole, D. S. Bolme, J. Dunlop, Y. M. Lui, H. Sahibzada, and S. Weimer, "An introduction to the good, the bad, & the ugly face recognition challenge problem," in *IEEE Int. Conf. Automatic Face Gesture Recognition*, 2011, pp. 346–353.

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

[11] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "DeepFace: Closing the gap to human-level performance in face verification," in *IEEE Conf. Comput. Vision Pattern Recognition*, 2014, pp. 1701–1708.

[12] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," in *IEEE Conf. Comput. Vision Pattern Recognition*, 2015, pp. 815–823.

[13] O. M. Parkhi, A. Vedaldi, and A. Zisserman, "Deep face recognition," in *Brit. Machine Vision Conf.*, 2015.

[14] E. L. Denton, W. Zaremba, J. Bruna, Y. LeCun, and R. Fergus, "Exploiting linear structure within convolutional networks for efficient evaluation," in *Advances Neural Inform. Process. Syst.*, 2014, pp. 1269–1277.

[15] S. Han, J. Pool, J. Tran, and W. Dally, "Learning both weights and connections for efficient neural network," in *Advances Neural Inform. Process. Syst.*, 2015, pp. 1135–1143.

[16] S. Han, H. Mao, and W. Dally, "Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding," *arXiv preprint arXiv:1510.00149*, 2015.

[17] H. Li, A. Kadav, I. Durdanovic, H. Samet, and H. P. Graf, "Pruning filters for efficient convnets," *arXiv preprint arXiv:1608.08710*, 2016.

[18] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "MobileNets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[19] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5mb model size," *arXiv preprint arXiv:1602.07360*, 2016.

[20] G. Chiachia, A. X. Falcao, N. Pinto, A. Rocha, and D. Cox, "Learning person-specific representations from faces in the wild," *IEEE Trans. Inf. Forensics Security*, vol. 9, no. 12, pp. 2089–2099, 2014.

[21] F. Song and X. Tan, "Learning one-shot exemplar SVM from the web for face verification," in *Asian Conf. Comput. Vision*, 2014, pp. 408–422.

[22] A. Pinto, W. R. Schwartz, H. Pedrini, and A. Rocha, "Using visual rhythms for detecting video-based facial spoof attacks," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 5, pp. 1025–1038, 2015.

[23] Z. Boulkenafet, J. Komulainen, L. Li, X. Feng, and A. Hadid, "OULU-NPU: A mobile face presentation attack database with real-world variations," in *IEEE Int. Joint Conf. Biometrics*, 2017.

[24] Z. Boulkenafet, J. Komulainen *et al.*, "A competition on generalized software-based face presentation attack detection in mobile scenarios," in *IEEE Int. Joint Conf. Biometrics*, 2017.

[25] D. Yi, Z. Lei, S. Liao, and S. Z. Li, "Learning face representation from scratch," *arXiv preprint arXiv:1411.7923*, 2014.

[26] I. Masi, A. Tran, T. Hassner, J. T. Leksut, and G. Medioni, "Do we really need to collect millions of faces for effective face recognition?" in *Europ. Conf. Comput. Vision*, 2016.

[27] K. Grm, V. Štruc, A. Artiges, M. Caron, and H. K. Ekenel, "Strengths and weaknesses of deep learning models for face recognition against image degradations," *IET Biometrics*, vol. 7, no. 1, pp. 81–89, 2018.