Two-Stream CNNs for Gesture-Based Verification and Identification: Learning User Style

Jonathan Wu, Prakash Ishwar, Janusz Konrad*
Department of Electrical and Computer Engineering, Boston University
8 Saint Mary's Street, Boston, MA, 02215
[jonwu,pi,jkonrad]@bu.edu

Abstract

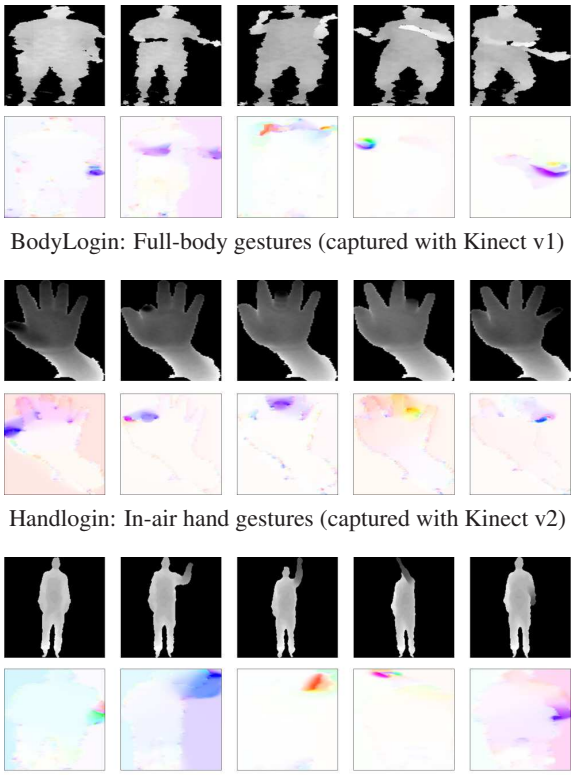
Recently, gestures have been proposed as an alternative biometric modality to traditional biometrics such as face, fingerprint, iris and gait. As a biometric, gesture is a short body motion that contains static anatomical information and changing behavioral (dynamic) information. We consider two types of gestures: full-body gestures, such as a wave of the arms, and hand gestures, such as a subtle curl of the fingers and palm. Most prior work in this area evaluates gestures in the context of a "password," where each user has a single, chosen gesture motion. Contrary to prior work, we instead aim to learn a user's gesture "style" from a set of training gestures. We use two-stream convolutional neural networks, a form of deep learning, to learn this gesture style. First, we evaluate the generalization performance during testing of our approach against gestures or users that have not been seen during training. Then, we study the importance of dynamics by suppressing dynamic information in training and testing. We find that we are able to outperform state-of-the-art methods in identification and verification for two biometrics-oriented gesture datasets for body and in-air hand gestures.

1. Introduction

Biometrics are a convenient alternative to traditional forms of access control such as passwords and pass-cards since they rely solely on user-specific traits. Unlike alphanumeric passwords, biometrics cannot be given or told to another person, and unlike pass-cards, are always "on-hand." Perhaps the most well-known biometrics with these properties are: fingerprint, face, speech, iris and, gait.

A gesture is a short, few seconds long, body motion that contains static anatomical information and changing behavioral (dynamic) information. We consider both full-body gestures, such as a wave of the arms, and hand gestures such as a subtle curl of the fingers and palm. For identification and verification, a user can chose a specific gesture as a "password."

In this work, rather than focusing on identifying a user performing a specific "password," we aim to identify a user across a *set* of gestures, in effect learning a user's *gesture style*. We focus on body- and hand-based gestures from



BodyLogin: Full-body gestures (captured with Kinect v1)

Handlogin: In-air hand gestures (captured with Kinect v2)

MSRAction3D: Full-body gestures (captured with Kinect v1)

Figure 1. Examples of normalized depth images and corresponding colored optical flow [14] for body and hand gestures captured using various depth sensors. Hue indicates optical flow orientation, and saturation indicates magnitude.

*This work was supported by the National Science Foundation (NSF) under award CNS-1228869.

depth maps acquired by Kinect sensors (v1 and v2) [2] (Fig.1).

Extensive literature exists for depth-based *gesture recognition* for body [15, 24, 31, 5] and hand [17, 10, 21] gestures. However, there are few works for user identification and verification based on gestures. Both body- and hand-based gesture biometrics have been investigated independently using primarily depth silhouette shape [27, 26] and skeletal features (pose estimates from depth maps) [12, 3, 30, 11]. In [26], a temporal hierarchy of depth-based silhouette covariances from hand gestures was used to authenticate users, whereas in [3] a dynamic time warping (DTW) algorithm applied to fingertip and palm coordinates (hand pose estimates), that were estimated from depth images, was used. Perhaps the work that is closest to the goal of this paper is [11], where action-specific metric learning from normalized joint positions of the body was used to predict identity from a pool of known actions. We differ from that work, in that we learn user identity directly from depth images, without the need to have pose estimates of body joint positions. We use depth maps and the associated optical flow, which can be useful in cases when skeletal pose estimation is not reliable or fully available (such as for hand poses).

This paper makes the following key contributions:

- Development of a two-stream convolutional neural network for user identification and verification based on body and hand gestures.
- Evaluation of the generalization performance for unseen gestures and users in the training set.
- Assessment of the value of dynamics for user identification and verification.
- A t-SNE-based assessment of the capacity of the studied methods to represent gestures independently of users (gesture recognition) or to represent users independently of gestures (user style in verification and identification).

We validate our approach on two biometrics-oriented datasets (BodyLogin and HandLogin), and one gesture-centric dataset (MSRAction3D).

2. Convolutional Neural Networks

Deep convolutional neural networks (CNNs) have become very successful in vision tasks involving single still images. One of the contributions of this paper is in adapting CNNs to gesture-based biometrics where both static limb proportions as well as gesture dynamics come into play.

The goal of CNNs is to learn a large set of kernel weights optimized for a particular loss function. Within this domain, several single-image network architectures have been proposed, such as: AlexNet [9], GoogLeNet [22], and VGGNet [20]. These networks generally vary in the number of layers and the number and size of kernels.

In this paper, we analyze the biometric performance of gestures using AlexNet. AlexNet [9] is an eight-layer deep convolutional network consisting of five convolutional and three fully-connected layers (the last of which is a soft-max layer). We adapt this network to gesture sequences by using a variant of the two-stream convolutional network architecture proposed in [19]. Two-stream convolutional networks, as the name implies, train two separate convolutional networks: one for spatial information, and a second one for temporal information. Although such networks were originally intended for RGB images, we have adapted them to handle depth maps (Fig. 2).

The first network is a "spatial stream" convolutional network where a stream of T input depth map frames, extracted from the input video through subsampling, are mapped to a stream of T output feature vectors o_s by passing each frame, one-by-one, through the network (Fig. 2).

The second network is a "temporal stream" convolutional network that takes a sequence of T colored optical flow frames (corresponding to the T spatial-stream input frames) as input. Optical flow [14] is computed for each pair of consecutive depth map images (depth map values are treated as luminance values). The computed optical flow vectors are mapped into polar coordinates and then converted to hue, based on the angle, and saturation, based on the magnitude, with a fixed brightness (Fig. 1). Much like the first network, this stream of T input optical flow frames is mapped to a stream of T output feature vectors o_t by passing every colored optical flow frame, one-by-one, through the temporal-stream network.

A simple convex combination of the outputs of both networks is used to yield a single output o_c which is used for performance evaluation:

$$o_c = w_s o_s + w_t o_t,$$

where $w_s \geq 0$ is the spatial-stream network weight, $w_t \geq 0$ is the temporal-stream network weight, $w_s + w_t = 1$, and o_s and o_t are the respective network outputs. When $w_s = 1, w_t = 0$, only information from the spatial-stream network is used, and when $w_s = 0, w_t = 1$, only information from the temporal-stream network is used. We will report results for various combinations of (w_s, w_t) weights.

2.1. CNNs for Identification and Verification

Identification: The use of this network for *closed-set identification*, i.e., given a gesture, *identify* a user from a set of known users, is straightforward. During training (see Section 2.2), gesture sequences are broken up into single frames to be trained standalone. During testing, we take

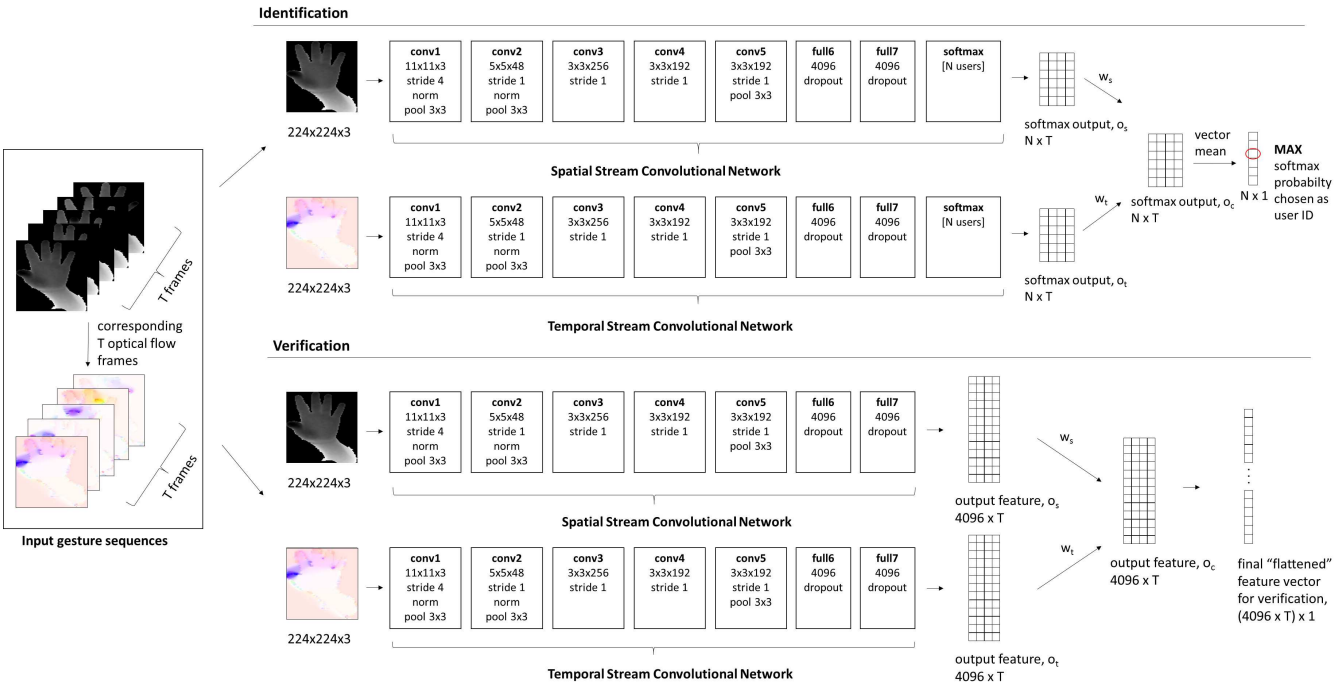


Figure 2. A visualization of how we use a deep network for user identification and verification. In identification (top), we fully fine-tune a network, using gesture depth map frames and optical flow. In verification (bottom), we borrow weights from an identification network, and use the outputs of the fully-connected layer as the verification features.

the mean of the soft-max probability outputs o_c across T frames (Fig. 2). Recall that o_c is a weighted combination of the softmax probabilities for an input across two networks. This yields a single soft-max probability vector of length N (given N users to identify), and the component with the largest probability identifies the user. Although not the main focus of this paper, gesture recognition uses the same structure where N is the number of gestures rather than the number of users to identity.

Verification: In *verification*[1] (given a gesture, is a user who (s)he claims to be?), we propose using the output features from the "full7" layer of a network trained for identification (Fig. 2). This avoids having to train a separate verification network for each user which is very expensive computationally. In addition, there are simply not enough training samples for each positive class represented by an authentic user to fully train a network. In this approach, for T frames that are uniformly sampled from a gesture sequence, two features of dimension $4096 \times T$ (the length of the last fully connected layer) are extracted yielding o_s and o_t, whose linear combination gives o_c. Since there is no built-in classification in this approach (no softmax layer), we use these features as inputs to a two-class classification algorithm for verification, e.g., based on nearest-neighbor or SVM. The intuition behind this idea is that, given enough users to identify, the network will naturally learn a user-separating feature space which can be leveraged for verification.

We discuss the parameters and training of all the elements of our networks in the next section.

2.2. Network Implementation Details

Typically, there are not enough training samples in gesture datasets to train all the weights of a deep convolutional network from scratch. Therefore, we follow the common practice to "pre-train" the network [4, 8] using weights from another network with sufficient data and then fine-tune those weights for new data. In our case, we use the dataset from ImageNet [18] (a network with a softmax loss function to classify RGB images into 1000 classes) to initialize the weights in our 5 convolutional layers (conv1 to conv5). Although our modality is different, as we use depth images and colored optical flow (instead of RGB), initializing with ImageNet weights is still effective. Our fully-connected layers are trained from scratch, with weights initialized to be zero-mean Gaussian with a small standard deviation of 0.001. In all our networks, we use a batch size of 256 images. For the spatial-stream network, we start with a learning rate of 0.003, decreasing this rate by one-tenth every 3,000 iterations until a total of 12,000 iterations are completed. For the temporal-stream network, we start with a learning rate of 0.001, decreasing this rate by one-tenth every 1,000 iterations until a total of 6,000 iterations are completed. The dropout value is set to 0.5 in the fully-connected layers of both networks.

[1] Verification is also called authentication.

We implement, in entirety, all our networks using Caffe [7, 25] on a single Titan Z GPU.

3. Gesture Datasets

We evaluate our method on 3 publicly available datasets. Two of these datasets were designed for user verification and identification (collected with the intention of maximizing the number of users). The third one was designed for gesture recognition (collected with the intention of maximizing the number of gesture/action types).

HandLogin [26] is a dataset containing in-air hand gesture sequences of 21 users, each performing 4 different gestures that are recorded by a Kinect v2 sensor. These gestures are: compass (move open hand in multiple directions), piano (move fingers as if playing piano), push (move open hand towards and away from the sensor), and flipping fist (twist and curl hand into a fist). Each user performed 10 samples of each gesture.

BodyLogin [27, 30, 28] is a full body multi-view dataset containing gesture sequences of 40 users performing 5 different gestures that are recorded by Kinect v1 sensors. Four of these gestures are predefined: S gesture (user draws an "S" shape with both arms), left-right (user reaches right shoulder with left hand, and then left shoulder with right hand), double-handed arch (user moves both arms in an upwards arch), and balancing (user performs a complex balancing gesture involving arms and legs). The fifth gesture is created by the user (user-defined). Each user performed each gesture about 20 times under varying degradations, such as carrying a bag, wearing a coat, passage of time, and also under spoof attacks. In this study, we train and test with samples across all degradations, and only from the center camera viewpoint.

MSRAction3D [13, 24] is a full-body, single-view dataset containing motion sequences of 10 users, performing 20 different actions in front of a Kinect v1 sensor. Each subject performs each action 2 or 3 times, with a total of 567 depth map sequences. Actions in this dataset are quite varied, for example: arm waves, hammer motions, catches, punches, symbol drawings, kicks, tennis swings, golf swings, and jogging. Although in [13], the actions are split into 3 subsets for evaluation, we instead evaluate all the actions at once in all our experiments, which is a more difficult scenario.

The depth data from all datasets are first background subtracted (background frames are given) and then normalized and resized using bicubic interpolation to 224×224 pixels as shown in Fig. 1.

4. Performance Evaluation

We evaluate performance for two access control scenarios [6]: closed-set identification and verification.

In *closed-set identification*, given a query gesture sequence, an identity is predicted from a pool of known users. The performance measure we use for identification is the correct classification error (CCE), which is the rate at which users are incorrectly identified.

In *verification*, given a query gesture sequence and claimed identity, the claim is either verified or rejected. If the query is sufficiently close in distance to a known, enrolled gesture sequence of the claimed identity, it will be accepted as that user; otherwise, it will be rejected. An error in verification results from either a false acceptance or a false rejection. The false acceptance rate (FAR) is the rate at which *unauthorized* users are accepted and is a measure of security. The false rejection rate (FRR) is the rate at which *authorized* users are denied access and is a measure of convenience. There exists a trade-off between FAR and FRR which is controlled by a threshold on acceptance distance (between the query and closest enrolled gesture). A popular metric that captures this trade-off with a single scalar is the equal error rate (EER) which is the FAR (or FRR) for the threshold when FAR and FRR are equal.

In our verification experiments, we use the ℓ_2 distance between the features of gesture sequences (flattened vectors of length $4096 \times T, T = 50$). If the distance between a query sample and its nearest-neighbor enrolled sample of the claimed identity is below a threshold, it is accepted; otherwise, it is rejected. In this paper, we report the EER for our verification experiments. Additional details on this can be found in [26].

5. Results and Discussion

In all of our experiments, we benchmark against reimplemented depth silhouette covariance features as proposed in [26]. This method is not based on convolutional neural networks.

User Identification: We attempt to identify a user across a whole pool of possible gestures. We test performance both when a gesture has been seen by the system and also when it has not. The latter case evaluates how well our learned model *generalizes* to gestures that have not been part of the training set. If it performs well, our model would have, in effect, learned a specific "style" with which a user performs gestures, not just the specific gestures a user performs.

Results for both the BodyLogin and Handlogin datasets are shown in Table 1. The first row of this table ("All / All") refers to a scenario when the network has been trained with samples from all gestures. In this row, we split the dataset into one half for training and the other half for testing, where each half contains samples from all gestures. The remaining rows in the table are for scenarios when the network has been trained on some gestures while tested on a different unseen gesture. For example, for "All but Fist / Fist" the network has been trained on "Compass," "Piano"

Table 1. User identification results for BodyLogin and HandLogin.

Dataset	*Scenario*	**User Identification CCE (%)**					
		Weighted Convnets (w_s, w_t)					Baseline
	Training / Testing Gestures	⟵ Spatial				Temporal ⟶	Wu [26]
		$(1,0)$	$(\frac{2}{3},\frac{1}{3})$	$(\frac{1}{2},\frac{1}{2})$	$(\frac{1}{3},\frac{2}{3})$	$(0,1)$	
HandLogin (21 users, 4 gestures)	1. All / All	0.24%	0.24%	**0.24%**	0.71%	4.05%	6.43%
	2. All but Compass / Compass	**2.38%**	2.86%	4.76%	8.57%	36.19%	82.38%
	3. All but Piano / Piano	1.91%	**0.48%**	1.43%	1.91%	12.86%	68.10%
	4. All but Push / Push	**44.29%**	49.05%	54.29%	67.62%	77.14%	79.52%
	5. All but Fist / Fist	16.67%	**15.71%**	17.14%	20.00%	31.43%	72.38%
BodyLogin (40 users, 5 gestures)	1. All / All	0.05%	0.05%	0.05%	**0.05%**	5.01%	1.15%
	2. All but S / S	**0.75%**	1.00%	1.25%	1.75%	16.75%	75.75%
	3. All but Left-Right / Left-Right	**0.88%**	1.25%	1.50%	1.88%	11.50%	80.88%
	4. All but 2-Handed Arch / 2-Handed Arch	0.13%	0.13%	**0.13%**	0.38%	6.25%	74.50%
	5. All but Balancing / Balancing	**9.26%**	10.01%	13.27%	19.52%	45.06%	77.97%
	6. All but User Defined / User Defined	**5.28%**	5.53%	6.16%	8.54%	22.49%	71.61%

and "Push" but tested on "Fist." In Table 2, we report results for user identification on the MSRAction3D dataset. Here, we train only on one sample of each action, and test on the remaining 1-2 samples. This is the same as the row ("All / All") in Table 1, where we train with samples from all gestures. In addition to our silhouette covariance benchmark from [26], we also compare to the reported user identification results from [11], which uses skeletal joint estimates and a distance metric based on skeletal coordinates to determine user identity.

Table 2. User identification on MSRAction3D. [11] performs user identification on skeletal pose estimates derived from depth maps.

Dataset	**User Identification CCE (%)**				
	Weighted Convnets (w_s, w_t)			Baselines	
	⟵ Spatial		Temporal ⟶	Wu [26]	[11]
	$(1,0)$	$(\frac{1}{2},\frac{1}{2})$	$(0,1)$		
MSR	0.0%	**0.0%**	0.53%	13.6%	7.0%

Suppression of Dynamics in User Identification: In order to understand the impact of dynamics in our deep network representation, we studied the effect of "removing" it. Although a similar study was done in [29], that was based on skeletal pose estimates. Our study is based on depth maps. We consider *both* the input to the temporal-stream network, as well as the input to the spatial-stream network as containing full dynamic information. To suppress the impact of dynamics, we remove the temporal network completely, and use only the first 3 depth map frames (out of typically hundreds of frames, spanning the time duration of less than a tenth of a second) as input to the spatial stream network.

In Table 3, we show the empirical performance of dynamics suppression for our two-stream approach as well as for the approach in [26] which we have reimplemented for this experiment.

Table 3. Results for the suppression of dynamics in user identification: only first 3 frames of each depth map sequence are used for training and testing, and the temporal stream is disabled $(w_s = 1, w_t = 0)$.

Dataset	*Scenario*	**User Ident. CCE (%)**	
	Data Used	Spatial	Wu [26]
HandLogin	All frames	**0.24%**	6.43%
	No dynamics	**1.90%**	9.29%
BodyLogin	All frames	**0.05%**	1.15%
	No dynamics	**1.00%**	32.60%

User Verification: Here, we attempt to verify a user's query gesture and claimed identity against a pool of known gestures (all gestures of the claimed identity). As it is impractical to train a deep network for each user, we instead train an identification network first and use it as a *feature extractor* for verification (see Section 2). In our experiments, we "leave-out" one-fourth of the user pool for testing, and train an identification network (for feature extraction) on the remaining three-fourths. For BodyLogin, this is leave-10-persons-out and for HandLogin this is leave-5-persons-out cross-validation. In the benchmark verification method, we use covariance features from the test samples. We report these results averaged across 4 "leave-out" folds for verification in Table 4 for Bodylogin and HandLogin.

Gesture Recognition: Here, we attempt to recognize the

Table 4. User verification results for BodyLogin and HandLogin.

Dataset	*Scenario*	**Verification EER (%)**					
		Weighted Convnets (w_s, w_t)					Baseline
	Users	⟵ Spatial				Temporal ⟶	Wu [26]
		$(1,0)$	$(\frac{2}{3},\frac{1}{3})$	$(\frac{1}{2},\frac{1}{2})$	$(\frac{1}{3},\frac{2}{3})$	$(0,1)$	
HandLogin	Leave 5 persons out	2.52%	**2.20%**	2.71%	4.09%	6.50%	11.45%
BodyLogin	Leave 10 persons out	2.76%	2.45%	**1.99%**	3.07%	8.29%	3.46%

gesture type performed across a pool of users. While in user identification we are trying to learn the user identity irrespective of which gestures the user performs, in gesture recognition we are trying to learn the gesture irrespective of the users who perform them. Similar to how we "leave-out" gestures in our user identification experiments, we "leave-out" users in our gesture recognition experiments. Specifically, we "leave-out" half of the user pool for testing, and train a gesture recognition network on the remaining half. For MSRAction3D, we employ the cross-validation approach of leaving 5 persons out as done in [16], and in BodyLogin[2] and Handlogin, we perform leave-20-persons-out, and leave-10-persons-out (half of each dataset population), respectively. We report results for gesture recognition in Table 5.

Table 5. Gesture recognition results. For each dataset, we perform leave-$(N/2)$-persons-out cross-validation, where N is equal to the total number of users in the dataset.

Dataset	**Gesture Recognition CCE (%)**			
	Weighted Convnets (w_s, w_t)			Baseline
	⟵ Spatial		Temporal ⟶	Wu [26]
	$(1,0)$	$(\frac{1}{2},\frac{1}{2})$	$(0,1)$	
HandLogin	15.00%	6.82%	10.91%	**0.91%**
BodyLogin	21.10%	**15.09%**	20.35%	15.44%
MSRAction3D	44.36%	36.00%	40.36%	**25.45%**

Discussion: The above results demonstrate a significant decrease in error when using deep networks compared to benchmark methods in user identification (all 3 datasets) and verification (HandLogin and BodyLogin).[3] This decrease is most striking in identification, when we test gestures that have not been used in training the network. In stark contrast to the CNN features proposed in our work, the covariance features proposed in [26] do not generalize well *across* gestures, i.e., when gestures that are not part of the training set appear in the test set. This can be seen most clearly by examining the CCE values for the "Compass" gesture in Table 1. The CCE for covariance features is as high as 82.38% while it is only 2.38% for our CNN features.

This cross-gesture generalization capacity of CNNs is also observed in the t-SNE embeddings [23] of the "full7" layer outputs for Handlogin (Fig. 3), BodyLogin (Fig. 4), and MSRAction3D (Fig. 5) datasets. Part (a) of each figure shows the feature embedding for our baseline, which favors clustering by gesture type. Parts (b), (c), and (d) show the feature embeddings for our convolutional networks. In part (b), the pre-trained embedding from ImageNet tends to favor clustering points by gesture type. After fine-tuning for identification in part (c), we see clustering by user identity. This reveals that it is very beneficial to fine-tune our networks from the pre-trained weights in order to cluster by user. Fine-tuning for gesture recognition, shown in part (d), causes even more compact clustering by gesture type than in part (b). Note that in the t-SNE plots of the "full7" layer outputs after fine-tuning for identification (part (c)) users tend to cluster together whereas gesture types are mixed within each cluster. However, in the corresponding t-SNE plots of the covariance features (part (a)), gesture types tend to cluster together with users being mixed within each cluster.

There are cases where our network does not generalize well across gestures, e.g., the "Push" gesture. We posit that this lower performance occurs because the trained gestures are significantly different in form and dynamics from the other gestures. The "Push" gesture contains variations in *scale* whereas the other gestures do not. The "Fist" gesture contains motion that completely occludes the shape of the hand, which is not in the other gestures. The "Balancing" gesture includes leg movements, not so for other gestures. For the most part, this type of result is to be expected. It will always be difficult to generalize to a completely unknown gesture that has little-to-no shared components with training gestures.

For identification on MSRAction3D, we get 0% classification error. Although seemingly surprising, this result might be attributed to the dataset collection procedure. In

[2]Of the 5 gesture classes in BodyLogin, 4 gesture classes are shared across users, and 1 is not, being user-defined. This means that in leave-persons-out gesture recognition, the fifth gesture class will not have samples of its gesture type in training. As a result, the fifth gesture class is expected to act as a "reject"/"not gestures 1 - 4" category for gesture recognition.

[3]Due to the general lack of per-user samples in MSRAction3D (as it is a gesture-centric dataset), we do not report results for verification, and leave-gesture-out experiments for identification.

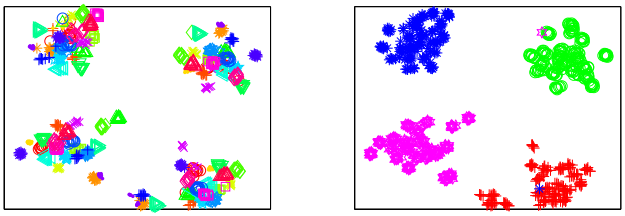
MSRAction3D, gesture samples from a user are extracted by partitioning one long continuous video into multiple sample parts. While not an issue for gesture recognition (as the same user will never be in *both* training and test sets due to "leave-persons-out" testing), this can result in biases for user recognition. This bias stems from almost identical, partially shared body postures across samples, which the deep network learns very well. The aforementioned issue is avoided in BodyLogin and HandLogin, as there is a "reset" procedure between samples, since samples are *not* recorded from one long continous sequence (users leave and re-enter the room between samples).

For verification, the differences are far less dramatic, but CNN features still yield a decent decrease in EER. In both scenarios, the smaller the value, the better the performance (we want small EER and CCE).

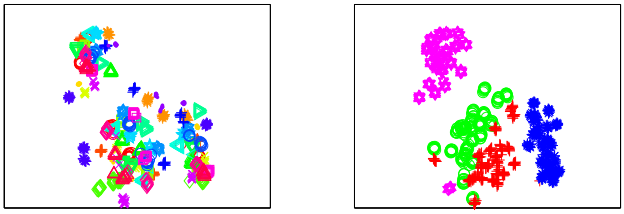
Across all our results, the temporal stream is complementary to the spatial stream for user identification, verification, and even gesture recognition. That is, having a temporal stream weight $w_t \neq 0$, will not degrade performance. The only exception to this, is when *information* is not seen in the training phase such as in leave-gesture-out results for user identification in Table 1. The reduced performance due to the inclusion of the temporal stream is not entirely surprising, as there are body/hand motions in testing that have not been seen in training (unseen optical flow vectors). As a result, this ends up generalizing poorly, whereas the static poses from the spatial network still fare quite well. Across all experimental results, a simplistic weighted average of $(\frac{1}{2}, \frac{1}{2})$ is perhaps the best option.

Our experiments involving dynamics suppression in user identification (Table 3) confirm that motion plays a crucial role; it can reduce the mis-identification rate from 1 error in 100 attempts to 1 error in 2,000 attempts (for BodyLogin). This conclusion is consistent across both methods we evaluate.

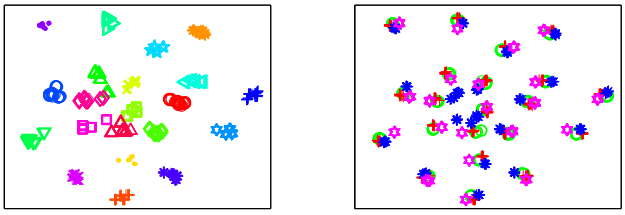
In gesture recognition, our deep learning approach slightly outperforms the non-CNN approach on BodyLogin, but is outperformed on the other datasets. We speculate that this is due to the size of the dataset. Notably, BodyLogin is our largest dataset with the most samples (≈4,000 gesture sequences, ≈150 frames each), and can beat our baseline. This is larger than both HandLogin (≈840 gesture sequences, ≈150 frames each) and MSRAction3D (≈600 gesture sequences, ≈35 frames each) combined, both of which underperform in gesture recognition. As the CNN approach outperforms the baseline in all other experiments, this perhaps suggests that with fewer samples it is easier to discriminate between users, than it is to discriminate between gestures. Overall, we believe that on larger datasets such as BodyLogin, deep learning will likely outperform the baseline.



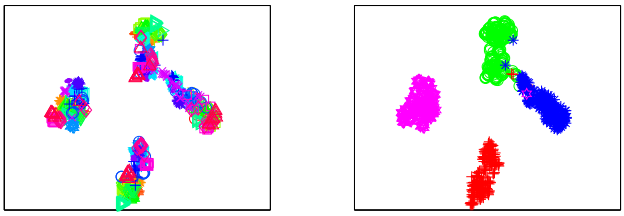
(a) HandLogin silhouette covariance features [26]



(b) HandLogin pre-trained "full7" features (no fine tuning)



(c) HandLogin *user identification* fine-tuned "full7" features

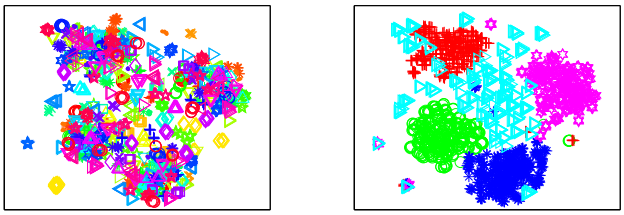


(d) HandLogin *gesture recognition* fine-tuned "full7" features

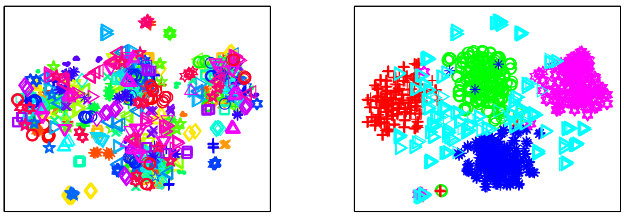
Figure 3. 2-D t-SNE embeddings of features for the HandLogin dataset. Left-column plots are color-coded by user, whereas those in the right column are color-coded by gesture type. A single marker represents a single gesture sequence. These figures show the t-SNE embeddings of the last fully-connected layer's output from our convolutional networks (before and after fine-tuning), and those from our baseline, silhouette-covariance features.

6. Conclusions

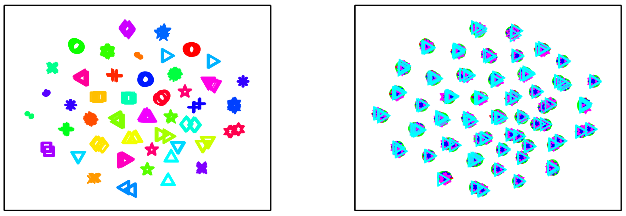
This is the first work to investigate the use of two-stream convolutional networks for learning user-specific gesture "styles". Most prior works assume a single gesture password per user and perform poorly when gesture types that are not encountered in the training set appear during testing. The proposed CNN-based features are able to effectively generalize across multiple types of gestures performed by the same user by implicitly learning a representation that depends only on the intrinsic "style" of each user as opposed to the specific gesture as we demonstrated across multiple



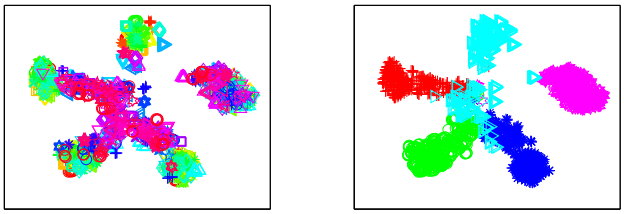
(a) BodyLogin silhouette covariance features [26]



(b) BodyLogin pre-trained "full7" features (no fine tuning)

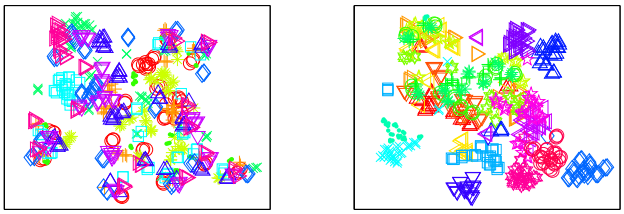


(c) BodyLogin *user identification* fine-tuned "full7" features

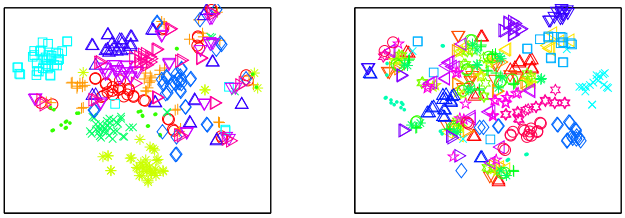


(d) BodyLogin *gesture recognition* fine-tuned "full7" features

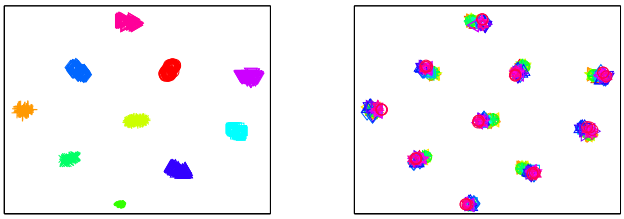
Figure 4. 2-D t-SNE embeddings of features for the BodyLogin dataset. For additional information, please see Figure 3. The cyan marker denotes user-defined gestures where any motion is allowed; it is not expected to cluster tightly.



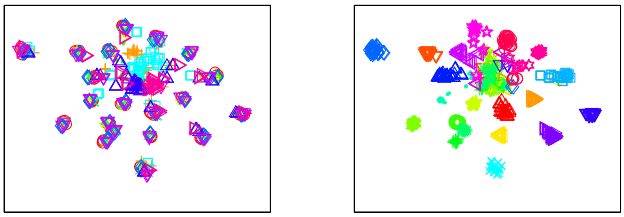
(a) MSRAction3D silhouette covariance features [26]



(b) MSRAction3D pre-trained "full7" features (no fine tuning)



(c) MSRAction3D *user identification* fine-tuned "full7" features



(d) MSRAction3D *gesture recognition* fine-tuned "full7" features

Figure 5. 2-D t-SNE embeddings of features for the MSRAction3D dataset. For additional information, please see Figure 3.

datasets.

A key practical outcome of this approach is that for verification and identification there is no need to retrain a CNN as long as users do not use dramatically different gestures. With some degradation in performance, a similar new gesture can still be used for convenience. Additional information and resources for this work are available at [1].

References

[1] DeepLogin. http://vip.bu.edu/projects/hcis/deep-login.

[2] Kinect for Windows. http://www.microsoft.com/en-us/kinectforwindows/, 2014.

[3] M. Aumi and S. Kratz. Airauth: evaluating in-air hand gestures for authentication. In *Proceedings of the 16th International Conference on Human-Computer Interaction with Mobile Devices & Services*, pages 309–318. ACM, 2014.

[4] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. In *Proceedings of The 31st International Conference on Machine Learning*, pages 647–655, 2014.

[5] M. Hussein, M. Torki, M. Gowayyed, and M. El-Saban. Human action recognition using a temporal hierarchy of covariance descriptors on 3d joint locations. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence*, pages 2466–2472. AAAI Press, 2013.

[6] A. Jain, A. Ross, and K. Nandakumar. *Introduction to Biometrics*. Springer, 2011.

[7] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of*

the ACM International Conference on Multimedia, pages 675–678. ACM, 2014.

[8] S. Karayev, M. Trentacoste, H. Han, A. Agarwala, T. Darrell, A. Hertzmann, and H. Winnemoeller. Recognizing image style. In *Proceedings of the British Machine Vision Conference. BMVA Press*, 2014.

[9] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 1097–1105, 2012.

[10] A. Kurakin, Z. Zhang, and Z. Liu. A real time system for dynamic hand gesture recognition with a depth sensor. In *Signal Processing Conference (EUSIPCO), 2012 Proceedings of the 20th European*, pages 1975–1979. IEEE, 2012.

[11] I. Kviatkovsky, I. Shimshoni, and E. Rivlin. Person identification from action styles. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 84–92, 2015.

[12] K. Lai, J. Konrad, and P. Ishwar. Towards gesture-based user authentication. In *Advanced Video and Signal-Based Surveillance (AVSS), 2012 IEEE Ninth International Conference on*, pages 282 –287, Sept. 2012.

[13] W. Li, Z. Zhang, and Z. Liu. Action recognition based on a bag of 3d points. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on*, pages 9–14, 2010.

[14] C. Liu. *Beyond pixels: exploring new representations and applications for motion analysis*. PhD thesis, Citeseer, 2009.

[15] L. Miranda, T. Vieira, D. Martinez, T. Lewiner, A. Vieira, and M. Campos. Real-time gesture recognition from depth data through key poses learning and decision forests. In *25th SIBGRAPI Conference on Graphics, Patterns and Images*, pages 268–275. IEEE, 2012.

[16] O. Oreifej and Z. Liu. Hon4d: Histogram of oriented 4d normals for activity recognition from depth sequences. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 716–723, 2013.

[17] Z. Ren, J. Yuan, and Z. Zhang. Robust hand gesture recognition based on finger-earth mover's distance with a commodity depth camera. In *Proceedings of the 19th ACM International Conference on Multimedia*, pages 1093–1096. ACM, 2011.

[18] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

[19] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in Neural Information Processing Systems*, pages 568–576, 2014.

[20] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[21] J. Suarez and R. Murphy. Hand gesture recognition with depth images: A review. In *RO-MAN, 2012 IEEE*, pages 411–417, 2012.

[22] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015.

[23] L. Van der Maaten and G. Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(2579-2605):85, 2008.

[24] J. Wang, Z. Liu, Y. Wu, and J. Yuan. Mining actionlet ensemble for action recognition with depth cameras. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1290–1297, 2012.

[25] L. Wang, Y. Xiong, Z. Wang, and Y. Qiao. Towards good practices for very deep two-stream convnets. *arXiv preprint arXiv:1507.02159*, 2015.

[26] J. Wu, J. Christianson, J. Konrad, and P. Ishwar. Leveraging shape and depth in user authentication from in-air hand gestures. In *Image Processing (ICIP), 2015 IEEE International Conference on*, pages 3195–3199. IEEE, 2015.

[27] J. Wu, P. Ishwar, and J. Konrad. Silhouettes versus skeletons in gesture-based authentication with kinect. In *Proceedings of the IEEE Conference on Advanced Video and Signal-Based Surveillance (AVSS)*, 2014.

[28] J. Wu, P. Ishwar, and J. Konrad. The value of posture, build and dynamics in gesture-based user authentication. In *Biometrics (IJCB), 2014 IEEE International Joint Conference on*, pages 1–8. IEEE, 2014.

[29] J. Wu, P. Ishwar, and J. Konrad. The value of posture, build and dynamics in gesture-based user authentication. In *Proceedings of the IEEE International Joint Conference on Biometrics (IJCB)*, 2014.

[30] J. Wu, J. Konrad, and P. Ishwar. The value of multiple viewpoints in gesture-based user authentication. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 90–97, 2014.

[31] L. Xia, C. Chen, and J. Aggarwal. View invariant human action recognition using histograms of 3d joints. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 20–27, 2012.