Opisthenar: Hand Poses and Finger Tapping Recognition by Observing Back of Hand Using Embedded Wrist Camera

Hui-Shyong Yeo
School of Computer Science
University of St Andrews
Scotland, United Kingdom
hsy@st-andrews.ac.uk

Erwin Wu
School of Computing
Tokyo Institute of Technology
Tokyo, Japan
wu.e.aa@m.titech.ac.jp

Juyoung Lee
Graduate School of Culture Technology, KAIST
Daejeon, Republic of Korea
ejuyoung@kaist.ac.kr

Aaron Quigley
School of Computer Science
University of St Andrews
Scotland, United Kingdom
aquigley@st-andrews.ac.uk

Hideki Koike
School of Computing
Tokyo Institute of Technology
Tokyo, Japan
koike@c.titech.ac.jp

ABSTRACT

We introduce a vision-based technique to recognize static hand poses and dynamic finger tapping gestures. Our approach employs a camera on the wrist, with a view of the opisthenar (back of the hand) area. We envisage such cameras being included in a wrist-worn device such as a smartwatch, fitness tracker or wristband. Indeed, selected off-the-shelf smartwatches now incorporate a built-in camera on the side for photography purposes. However, in this configuration, the fingers are occluded from the view of the camera. The oblique angle and placement of the camera make typical vision-based techniques difficult to adopt. Our alternative approach observes small movements and changes in the shape, tendons, skin and bones on the opisthenar area. We train deep neural networks to recognize both hand poses and dynamic finger tapping gestures. While this is a challenging configuration for sensing, we tested the recognition with a real-time user test and achieved a high recognition rate of 89.4% (static poses) and 67.5% (dynamic gestures). Our results further demonstrate that our approach can generalize across sessions and to new users. Namely, users can remove and replace the wrist-worn device while new users can employ a previously trained system, to a certain degree. We conclude by demonstrating three applications and suggest future avenues of work based on sensing the back of the hand.

Author Keywords

Back of the hand; Opisthenar; Hand pose; Finger tapping;

CCS Concepts

•**Human-centered computing** → **Gestural input;**

UIST '19, October 20-23, 2019, New Orleans, LA, USA.

ACM ISBN 978-1-4503-6816-2/19/10 ...$15.00.
http://dx.doi.org/10.1145/3332165.3347867

INTRODUCTION

The ability to sense and recognize hand poses and gestures opens up the potential for new interactions with computing devices. The intent here is not to simply replicate existing gesture based sensing and interactions but instead explore sensing from a new perspective. Our goal is to determine hand pose and gesture in an orientation free manner, from a minimal wearable device without external sensing. Achieving this, affords us the opportunity to consider new forms of interaction in the one-handed, discreet and applications we demonstrate and along with proposing directions for future work.

Existing approaches to determining hand pose and gestures typically employ external sensing infrastructure, wearable gloves, wrist bands or arm bands. Such systems employ bespoke sensors, custom hardware or contrived sensor placement to realize the sensing. Our approach, by contrast, employs a camera to observe the back of the hand while different hand poses and gestures are performed. Such hand motions result in joint, tendon and bone movements, along with skin, blood vessel and shape deformations. To measure such changes, contact based approaches using strain gauges or optical sensors attached to the opisthenar area have been explored in [14, 25].

Our contact free approach employs a camera that can be embedded in many digital devices such as wrist worn wearable devices. Indeed, smartwatches with embedded cameras are now emerging for photography purposes. For example, Song et al. [23] exploited such a device to capture mid air gestures of the second hand interacting with a smartwatch. Here we seek to exploit such consumer grade cameras on smartwatches as shown in Figure 1 for our form of back of the hand sensing.

However, unlike some existing approaches which mount a vision sensor on the inner side of the wrist, our technique uses a camera on the outer side of the wrist, as we expect our technique to work on smartwatches with a built-in camera. This self-imposed constraint presents difficult challenges, such as a low angular view of the hand, proximity and self-occlusion

Figure 1. There are smartwatches (e.g., Zeblaze Thor 4 Dual shown here) with a built-in side camera. This can observe the back of hand area and can be leveraged to recognize hand poses and gestures.

which means most of the fingers are not visible to the camera especially when bended. However, as we will demonstrate, in spite of these challenges, we can recognize static hand poses and dynamic finger tapping gestures with a high degree of accuracy. This work makes the following contributions:

1. A novel vision-based technique to recognize hand poses and finger gestures using a camera placed on the outer wrist, which suffers from heavy occlusion of the fingers.

2. Applying modified two-stream convolutional networks with weighted motion images to improve the recognition of dynamic finger gestures and comparing different deep neural network architectures for our approach.

3. A thorough offline evaluation and a real-time user test in challenging conditions (removing and rewearing the device), which indicate generalizability across different rewearing locations and across different users to a certain degree.

4. Collecting and releasing a dataset of 1.38 million (training set) and 138k (test set) images and pretrained models.

RELATED WORK

Our related work spans multiple research areas including vision-based hand tracking, wearable sensors and activity recognition using deep neural networks.

Vision Based Approaches on Wearable Devices

Common approaches to determining hand pose and gestures on wearable devices employ vision sensors such as a camera or optical sensor on the inner side of the arm or wrist [11, 18, 28]. WristCam [28], Digits [11] and DigiTap [18] use camera to track hand poses or finger tapping actions. Digits [11] requires an IR laser line projector whereas DigiTap [18] requires a LED flash synced with an accelerometer to detect vibrations occurring during finger taps. In WatchSense [24], the authors created a compact wearable prototype, attached to a user's forearm, to detect finger interaction from the other hand. Closest to our work, Chen et al. [3] use elevated camera on the outer side of wrist to track 10 ASL hand poses. Closer to the hand, CyclopsRing [2] uses an ultra wide fish-eye camera, worn on a ring, to observe the inner palm and recognize hand gestures. Wrist-mounted cameras have also been explored in, for example, the recognition of daily activities [17]. However, many of these approaches, as noted, involve contrived and often impractical sensing arrangements.

Sensor Based Approaches on Wearable Devices

Another line of inquiry is based on wearable devices such as a glove, armband or wristband. Glove type devices (e.g., PowerGlove) are common and allow the accurate tracking of hand and finger joints, albeit at the cost of requiring a user to wear a cumbersome device. Other common approaches use various type of sensors such as EMG [21], FSR [4], NIR [15], IMU [13] and EIT [31] in wrist and armband devices. For example, Myo is a consumer device for sensing 5 hand gestures using Electromyography (EMG) sensors worn on the arm. Saponas et al. [21] also use EMG to enable always-available input. GestureWrist [20] recognizes hand gestures by measuring changes in wrist shape with capacitive sensing. Tomo [31] employs Electrical Impedance Tomography (EIT) to recognize gross hand gestures and thumb-to-finger pinches. WristFlex [4] utilizes force sensitive resistors (FSR) on the wrist band to detect 5 finger pinch poses. ViBand [13] and Serendipity [29] sense active hand gestures using inertial measurement unit (IMU).

Closer to what is proposed here are the approaches introduced in BackHand [14] and Behind the Palm [25]. Both require the attachment of sensors directly on the back of the hand to measure tendon movements and skin deformation. [14] uses strain gauges mounted on flexible sticky pads while [25] uses photo reflective sensors. In [14], a per-user accuracy of 95.8% using 10-fold cross-validation was reported, where each fold only contains data from a single trial of a single participant. In [25], 99.5% accuracy was reported. However, these evaluations are based on a single session without removing and rewearing the device, which raises the question of its usability in practical settings. More importantly, the result cannot generalize across users, with 27.4% for leave-one-user-out accuracy [14]. In our work, we show that our approach can not only generalize well across sessions (remove and rewear) but can also generalize somewhat across unseen users (79.8% leave-one-user-out).

Deep Neural Networks and Deep Learning

Deep neural networks are frequently used in image classification tasks, including hand pose and gesture estimation. Convolutional neural networks (CNN) have demonstrated the ability to learn the components (e.g., edges, lines, curves, shapes, etc.) of an image, so it can easily tell the difference between two static hand pose images. However, 2D CNN deals with each image in training, therefore, it cannot capture sequence or temporal information, and thus is not suitable for detecting dynamic gestures such as tapping or flicking the fingers. In motion recognition, there is existing work that adds a recurrent neural network (RNN) such as the long short-term memory (LSTM) layer after the CNN to learn the temporal features, which is called the Long-term Recurrent Convolutional Networks (LRCN) [5]. This approach has been shown to have high accuracy on some human action datasets such as the UCF-101. Another approach is to extract temporal features by employing a two-stream method [6, 22]. This involves inputting both the raw image and a stack of motion images (optical flow is often used) as two streams into the neural network that fuses their result. Other more complex networks such as 3D CNN [10] might also be suitable in learning temporal motions, but their model is heavy weight and requires more computation, hence it might not be suitable for real-time tasks on a smartwatch.

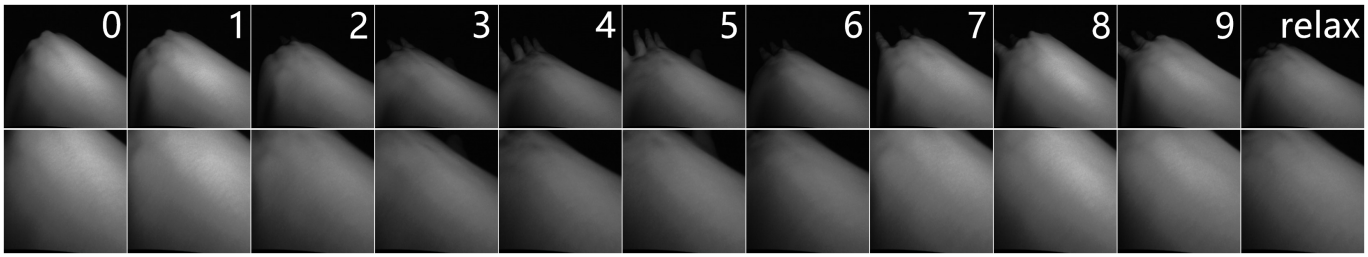

Figure 2. Example hand images (single frame for each pose) from one participant, from left to right: numbers in American Sign Language. Top row shows images using 299x299 pixels, bottom row shows cropped images using 175x175 pixels. Note that some fingertips are visible to the camera when they are fully extended (e.g., pose 3, 4, 5), but in other poses such as one finger pointing, the index finger is fully occluded even if it is fully extended.

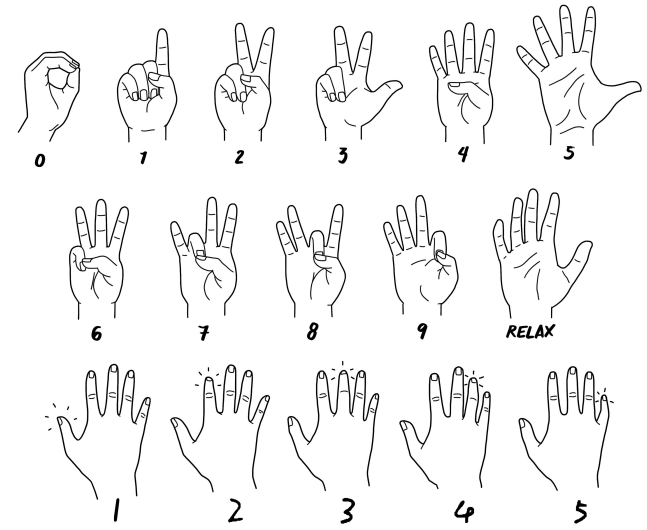

Figure 3. Top 2 rows: 10 numbers in American Sign Language used in our study. Bottom row: 5 finger tapping gestures used in our study.

DESIGN AND IMPLEMENTATION

The opisthenar area has the dorsal venous network - a web of veins and tendons, which exhibit changes due to hand and finger movement. Indeed, there have been systems proposed which use a camera to observe this area for authentication purposes [19, 30]. The work in [14] and [25] further demonstrate the potential of recognizing hand poses through measuring skin and tendon movement in this area. Inspired by their work, we aim to achieve a similar goal but with a more practical and constrained setting - using a camera worn on the outer wrist, such as those embedded on the side of a smartwatch.

In this work we consider both static hand poses and dynamic finger tapping gestures. In particular, we consider 10 numbers in American Sign Language (ASL) for static poses (Figure 3) following previous work [14, 25]. We also added 5 individual finger tapping action [9] for dynamic gestures, as these are particularly challenging in our constrained setting.

In addition, we can also recognize wrist whirling gestures [7], where we have trained a separate model for recognizing 4 directions (NSEW) and the recognition was very robust. This can be used as a gesture delimiter to indicate the start of a gesture, or to switch into an active mode to recognize hand poses, thus avoiding false positives during fast movement.

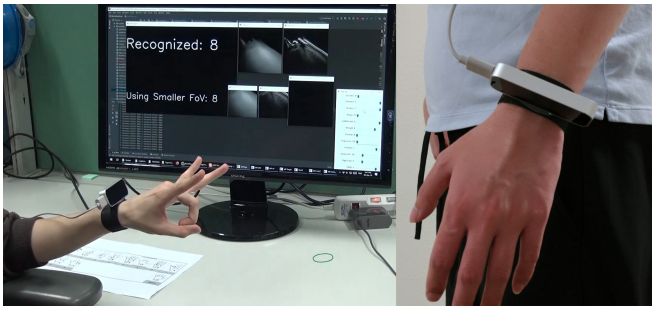

Figure 4. Participants wear the device on the wrist with a Velcro strap.

Hardware

We use a single infrared camera with active infrared light source for easy removal of the background. In our prototype, we re-purposed a Leap Motion device. Note that we do not use any finger tracking capabilities of the Leap Motion SDK, but merely treat it as single camera. In particular, we extract the infrared images from the left camera only

The Leap Motion imaging sensor (Aptina MT9V024, global shutter) has a native resolution of 752x480, covering a 135 degree field-of-view [16]. The resolution is cropped to 640x480 (VGA) and then further down-sampled vertically to 640x240 via hardware binning. The pixel size is 6 microns, hence the effective pixel resolution is quite low. Furthermore, we do not use the full field of view (FoV) of the camera. We crop the center area roughly covering the opisthenar area to emulate a different FoV of a smartwatch's camera for testing.

On the Leap Motion device there are three infrared LEDs with a wavelength of 850nm, which is outside the visible light spectrum. IR-pass filters on the lens block out other wavelengths and aid hand segmentation. We lowered the camera's exposure time to 20ms (also lowered the intensity of IR emitters, as they are physically wired to the camera's exposure pin), enabled high dynamic range (HDR), and adjusted the digital and analog gain to 20 and 5, respectively.

Image Preprocessing

We first rectify the images to remove distortion caused by the fish-eye lens and crop the center 299x299 pixels. As can be

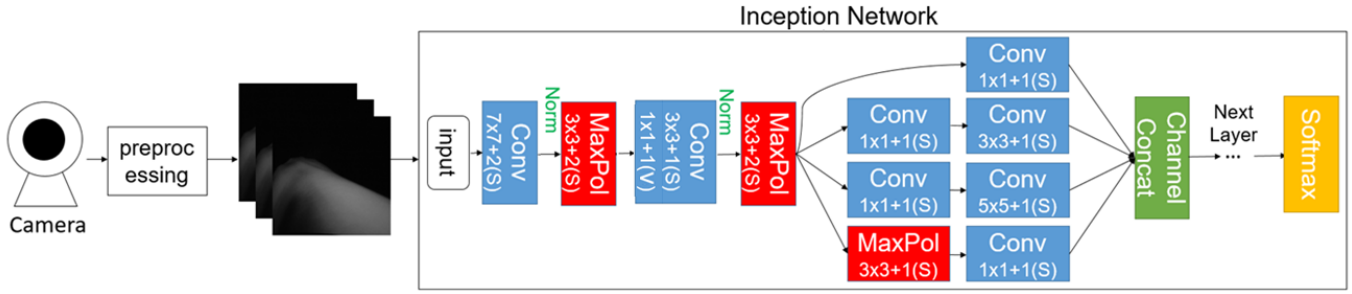

Figure 5. Our system pipeline and neural network architecture. After images are extracted from the camera, they went through image preprocessing such as image normalizing before passed to the input of an Inception network and go through multiple layers before reaching the last softmax layer.

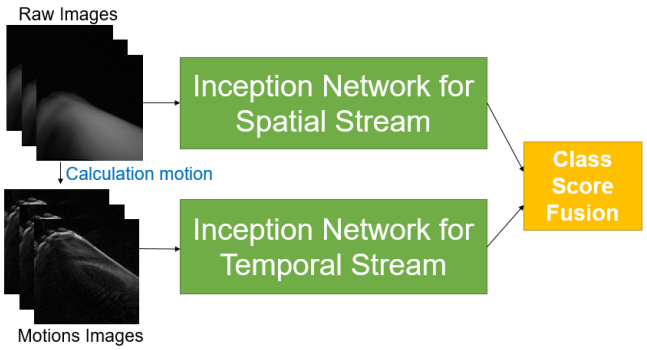

Figure 6. Two-stream neural network using both raw images and processed motion images as input.

seen in Figure 2 (top), in certain hand poses, some fingertips are visible to the camera when they are fully extended but in other poses, such as one finger pointing, the index finger is fully occluded even though it is fully extended. We further crop the center 175x175 pixels (Figure 2 bottom) to better emulate a more challenging setting (e.g., smaller FoV). In this tighter crop, no fingers can be seen by the camera.

Through our design and testing process, we found that a 2D Convolutional Neural Network (CNN) works well for static hand poses but not for dynamic finger gestures. Therefore, we improve our network architecture by trying i) Long-term Recurrent Convolutional Networks (LRCN) and later we settled at ii) Two-Stream Convolutional Network (TS-CNN) architecture. These are state-of-the-art action recognition methods.

With two-stream networks, the first stream typically uses a color or gray scale image while the second stream uses an optical flow image [22]. However, in our images, optical flow methods fail to track enough feature points, thus making the second stream input to the network weak. This is largely due to the relatively smooth skin surface and dark background.

Therefore, we instead use amplified weighted motion images as the input for second stream, akin to motion-history images (MHI) [1]. This is the absolute difference between the current frame and the weighted sum of past frames (alpha = 0.1), and each pixel intensity is multiplied by factor of 10. This was done to amplify the tiny skin deformation and tendon movements, as can be seen in Figure 6 (bottom left).

Deep Neural Network Architecture

We implemented our deep neural network using Keras with a Tensorflow backend. For static hand pose recognition, we use an Inception v1 [26] architecture (Figure 5), with softmax function at the last layer for 11 outputs. The inputs to the network are normalized single channel infrared images. For dynamic gestures we use a LRCN architecture and a TS-CNN with the Inception [26] architecture in each stream (Figure 6). For most of the model training, we use an Adam optimizer with a learning rate of 0.0001 and trained the network for 20 epochs, unless otherwise specified. The trained models are saved and later loaded during real-time user testing.

EVALUATION

We evaluated our system using a real-time user test, followed by offline cross-evaluation. Data was collected separately over 2 days from 10 participants (3 females, mean age = 25) we recruited from our department. The entire process took about 2 hours for each participant and they were compensated $20 for their time. On the first day we collected training data from the users and proceed to train a personalized model for each user, whereas on the second day we conducted a real-time user test, while the data are also saved as test set for later analysis. The test set is never used in the neural network training.

Procedure

Our participants wear a Velcro strap with the camera attached on the outer wrist of their watch-wearing hand (Figure 4), simulating a wrist-worn device with an embedded camera. After each session of data collection, we remove the device from the participant's wrist and put it back on a slightly different location. This is to sample more variance and noise in the data, to ensure the neural network can generalize across sessions.

We collect two types of data, one for static hand poses and one for dynamic finger gestures. There are 11 static hand poses consisting of 10 numbers in the American Sign Language, plus a relax pose (Figure 3 top), following the work in BackHand [14]. We add 6 dynamic gestures consisting of individual finger tapping actions, plus a relax pose (Figure 3 bottom).

On the first day, data was collected for 5 sessions covering all poses and gestures. For each trial within a session, we recorded 600 frames at roughly 30 fps. For static poses there are 10 participants x 5 sessions x 11 poses x 600 frames =

330k images. For dynamic gestures there are 360k images as we saved both gray-scale and the computed motion images.

For static hand poses we ask the participants to first move their hand in mid air through random positions and orientations, and then rest their elbow on the chair's arm rest while tilting their wrist in different directions. For dynamic gestures, we ask the participants to rest their elbow on the chair's arm rest and perform the finger tapping action continuously in the air.

On the second day, we ask participants to perform each of the static hand poses and finger tapping gestures in a randomized order, for 10 sessions. After each session we remove the strap and assist the participant to put it back on. In each trial, the participants perform the task and the author presses the space bar once to save 30 frames, which lasts about 2 seconds. After each trial, the result was displayed on screen, but only for the first 5 sessions and no feedback for the last 5 sessions. The data is also saved for offline analysis. There are 33k (static) and 36k (dynamic) images saved for this test set. Data and trained models can be found online at: https://github.com/tcboy88/opisthenar.

RESULTS AND ANALYSIS

We conducted real-time user test which yields more realistic results, that might be expected in real-world conditions. We also conducted cross-validation using leave-one-participant-out (user-independent) and leave-one-session-out (both user-dependent and user-independent). The results are shown in Tables 1 and 2, along with confusion matrices in Figures 7 and 8 and recognition rates by participant in Figure 9.

Static Hand Poses

In the real-time test we saved the result of 30 frames and selected the top 1 prediction with the highest occurrence. As shown in Table 1, the average accuracy for each participant (personalized) is 89.4% (SD: 9.0%). Using a smaller crop (175x175), the accuracy dropped to 51.7% (SD: 8.8%). The confusion matrix can be seen in Figure 7.

We also trained a generalized model using all 10 participants' training data. We then evaluated with the frames collected during the real-time test, which were not seen by the network before. This is almost equivalent to the real-time test, except without filtering with highest occurrence), and the accuracy is 88.0% (normal FoV) and 65.0% (smaller FoV).

We also conducted leave-one-user-out cross-validation, with results of 79.8% (evaluate on test set) and 71.6% (evaluate on train set, since one participant data is left out, it is fine to evaluate on train set as the trained model has never see this data before). For leave-one-session-out cross-validation, the results are 76.8% (using individual participant data and averaging the result) and 88.5% (using all 10 participants' data).

Dynamic Finger Tapping Gestures

In dynamic finger tapping recognition, we tried three different neural network architectures and finally settled on the one that yields the best result. We use 8:2 train/test split here to quickly experiment with different architectures. Initially, we used a similar network architecture as in static pose recognition (Inception) to recognize the dynamic motion on a per frame basis.

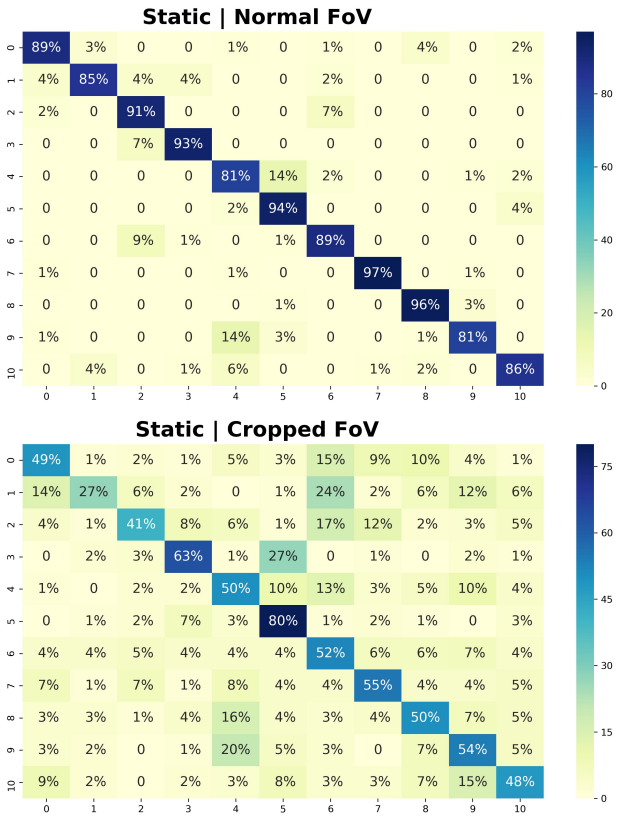

Figure 7. Confusion matrix for static poses recognition of real-time test. Top figure is using image with 299x299 pixels whereas bottom figure is using cropped image with 175x175 pixels.

However, the recognition rate was low at 39.6% (personalized) and 32.3% (user-independent, trained on all participants' data), as shown in Table 2. This is as expected, as a 2D CNN does not take into account the temporal information presented.

Next, we explored the long-term recurrent convolutional network (LRCN) [5] method, where the accuracy improved slightly to 39.7% (personalized) and 36.4% (user-independent). From frame to frame, we can observe that there are only very small changes on the back of the hand, which cannot be easily learned by an LSTM unit, we hypothesize.

Finally, we tried the two-stream networks architecture inspired by [22]. The train/test results (on Table 2) improved greatly to

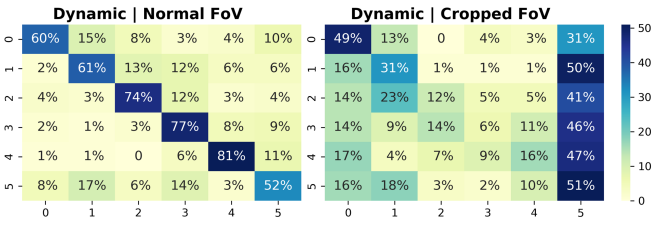

Figure 8. Confusion matrix for dynamic gestures recognition of real-time test. Left figure is using image with 299x299 pixels whereas right figure is using cropped image with 175x175 pixels.

Gesture	FoV	Real-Time Test Individual	Offline Test Individual	Offline Test General	Leave-1-participant Test set	Leave-1-participant Train set	Leave-1-session Individual	Leave-1-session General
Static	Normal	89.4%	88.6%	88.0%	79.8%	71.6%	76.8%	88.5%
Static	Cropped	51.7%	51.7%	65.0%	44.3%	35.5%	44.5%	53.9%
Dynamic (Two-Stream)	Normal	67.5%	62.3%	67.5%	54.3%	63.3%	-	-
Dynamic (Two-Stream)	Cropped	27.5%	46.9%	45.0%	44.4%	50.2%	-	-

Table 1. Left: Result of real-time user tests. "Individual" shows the average score on the personalized model test on each user; "General" shows the score of the general model tested on all users. Middle left: Result of offline tests. Middle right: Result of Leave-1-participant-out cross-validation. Test on test set and test on train set. Right: Result of Leave-1-session-out cross-validation.

Method	Mean of Individual.	Std.	General
Inception	39.6%	11.0	32.3%
Long-term RCN	39.7%	6.0	36.4%
Two-Stream	70.1%	5.8	69.7%

Table 2. Accuracy on dynamic finger tapping gestures using 3 different action recognition methods. Images using normal FoV, 8:2 train/test split using training data only.

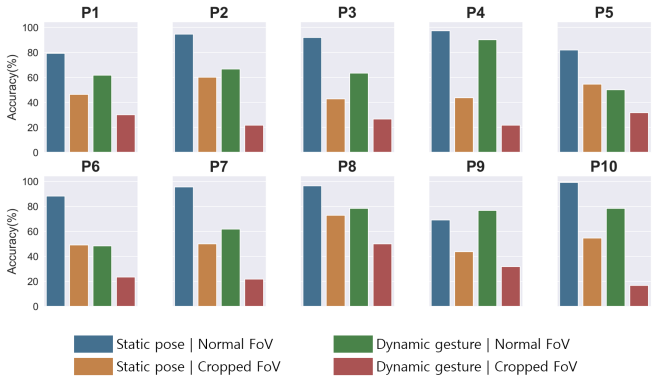

Figure 9. Real-time test results: recognition rate by participant, static hand pose vs. dynamic finger gesture, normal Fov vs. cropped FoV.

70.1% (personalized) and 69.7% (user-independent). Therefore, we chose this two-stream method for our real-time user test, where the average accuracy for each participant (personalized) is 67.5% (SD: 12.6%). Using a smaller crop (175x175), the real-time user test accuracy dropped to 27.5% (SD: 8.9%). The confusion matrix can be seen in Figure 8 (left: normal FoV, right: cropped FoV).

For the generalized model using all 10 participants' training data and test on real-time data, the accuracy is 67.5% (normal FoV) and 45.0% (smaller FoV). For leave-one-user-out cross-validation, the results are 54.3% (test on test set) and 63.3% (test on the training set). Due to the less encouraging result, we do not further test the leave-session results.

DISCUSSION

Static Hand Poses As shown in our results, our technique can recognize static hand poses with high accuracy in both real-time user test and offline analysis. Although the accuracy dropped when using a smaller FoV that observes only a small part of the opisthenar, the result remains promising. Therefore, it can be inferred that the neural network learns features not only from the veins and tendons, but also from the metacarpal, knuckles and fingertips when they are visible to the camera. Indeed, the tiny changes on the back of hand were captured by the camera and fed into the neural network, even though the imaging sensor used in our prototype has a limited resolution of 0.36 megapixel. We posit that using a higher resolution camera should result in a better recognition rate. In fact, most built-in cameras on smartwatches (e.g., Zeblaze) have a resolution of 2 to 5 megapixels, which is an order of magnitude higher that what is used here.

In our leave-one-user-out test, our strong user-independent results show that a deep neural network can learn useful features across many users and generalized it to new users. Compared to relevant work, BackHand [14] has only 27.4% accuracy for leave-one-user-out vs. our 79.8% (test on test set) and 71.6% (test on train set). In case of a new user, the trained model can also be used for transfer-learning or fine-tuning. Hence, a new user only requires minimal training to create a personalized model that works well. Interestingly, the user-independent result (65.0%) on a smaller crop is actually better than its personalized model (51.7%). It might be that the network is able to learn useful features from many users and improves the recognition rate when testing on single user.

In our leave-one-session-out test, the user-dependent accuracy is slightly lower than the real-time accuracy as we intentionally ask users to remove/rewear 5 times at quite different locations. This is done to capture more variety of data and camera angles. Indeed, the higher real-time result shows that it has captured usage variety and works well even for 10 times remove/rewearing. In addition, the general result is better, even in this session-fold split, because more data from diverse users help in training deep neural network and user generalization. For example, in certain session a user wore the watch at this position, and it generalizes to another user in another session.

Our technique is also robust to i) removal and rewearing of device ii) hand movement and wrist tilting. This is because during the data collection stage we collected data with these considerations in mind, where we ask the participants to move their arm in mid air and to tilt their wrist if different directions. Relevant work was evaluated in single session and has not shown any of these robustness characteristics.

Dynamic Finger Gestures Our system can also recognize dynamic gestures with 67.5% recognition accuracy but the rates were still less than desirable for real-world deployment. We hypothesize that this is, in part, due to how we collected the data and tested it. For simplicity, we asked participants to keep performing the finger tapping action while we recorded

the data for 600 frames. Therefore, the labelled data contains a mixture of no gesture, down gesture and up gesture, each with different magnitude and velocity. A better approach might be to synchronize and only record the start and end of a gesture, e.g., by pressing a key or with another hand tracking system.

On an actual smartwatch platform, we can also utilize the built-in accelerometer for hybrid sensing. For example, relevant work [13, 29] have demonstrated finger tapping action recognition using just the accelerometer. Furthermore, we can combine with orientation sensing to extend the gestures set, e.g., in WristFlex [4] the authors doubled the gestures set.

When using smaller crop, the index, middle and ring finger tapping actions are typically incorrectly recognized, as shown in the confusion matrix (Figure 8 right). According to human hand anatomy, these fingers are highly correlated and tend to move together, thus it might be difficult to differentiate which finger tapping action by only looking at a small patch of the opisthenar area. Further work might explore sensor fusion (IMU) to better separate these finger tapping actions.

APPLICATIONS

We created three simple applications (Figure 10) to demonstrate the potential and usefulness of our technique (please also refer to our video figure), including:

· Smartwatch control using both static and dynamic gestures
· A phone dialing interface controlled by ASL number poses
· A piano playing app which recognizes finger tapping.

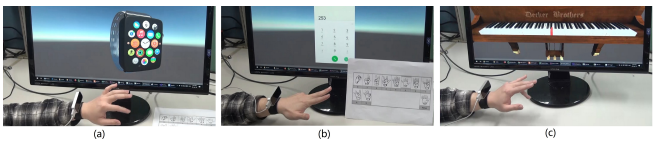

Figure 10. Demo applications: (a) smartwatch control, (b) phone dialing and (c) interactive piano controlled by individual finger tapping.

LIMITATIONS

For quick prototyping, we used a standalone infrared camera connected to a computer. We aim to port the system onto a smartwatch with a built-in RGB camera. This will present other challenges such as i) lighting and background noises ii) lower camera placement and limited field of view and iii) limited processing power. Here we suggest potential solutions.

First, the three channel RGB images might provide more useful information for deep neural network to learn from than the single channel that we used here. Second, we measured the center point of a Leap Motion and a smartwatch camera to the skin, which is 1.5cm and 1.1cm, respectively, hence a small difference. The accuracy using a smaller FoV is lower but is promising for further investigation in future work. Third, as technology advances, processing power advances so we foresee such methods becoming feasible in the near future. Mobile system on chip (SoC) nowadays has dedicated Neural Processing Unit (NPU) to accelerate deep learning and inference. In addition, there are efficient deep learning models (e.g., MobileNets [8], EfficientNet [27]) that can run on a resource constrained devices, while maintaining power efficiency.

We suspect that the accuracy of our method might differ for people with different hand size or shape (such as people with smaller hands tend to have higher accuracy in the study). Hence, more evaluations should be conducted to better understand how the physical characteristics of the hand will affect the recognition. Nonetheless, in practice there can always be a quick data collection session for an individual to fine tune the pretrained model to work better for that individual.

FUTURE WORK

More diverse gestures and participants In future work, we would like to explore more gestures (e.g., 25 hand activities in [12]) and the upper bound number of gestures without sacrificing accuracy. We would like to collect more data from diverse participants with different skin tones, or with tattoos, hairy hands to improve our model generalizability to more people. We would also like to explore the possibility of full articulated hand pose estimation.

Hand to hand and hand to object interaction We suggest the ability to determine static poses and dynamic gestures opens up the potential to explore other forms of interaction. Namely, when the hand intersects with another hand, or object, there are a range of interactions including, finger movement, touch, pinching or grasping. Consider, for example, in hand to hand interaction, the opisthenar area can act as a touchpad operated by the fingers of a second hand [24]. Such interactions can be extended to explore finger to finger interactions (such as clasping), or pinching or natural two-handed grasping actions. In this way, the dexterity of two hands can allow for both hands (while sensing only one) to afford new eyes free and potential discreet forms of input (e.g., a subtle directional stroke from one hand onto one's opposite wrist).

Similarly, by tracking the movements of the back of the hand we might consider what objects the hand is interacting with. Consider movements of fingers while typing on a keyboard, tapping a surface, holding a phone or grasping a mug. A hold and release action might trigger an event while moving from a pinch gesture to a firm grasp of an object might change the context of the current interaction. Likewise, proximate surfaces can now extend the range of inputs possible, without requiring instrumentation of the environment.

CONCLUSION

In this exploration, we proposed a vision-based approach for static hand poses and dynamic finger tapping gestures recognition, by observing the changes on the back of the hand using a camera that can be embedded in wrist-worn devices. Through our extensive evaluation, the results show that it is accurate for static poses, and could generalize across sessions and across users. The accuracy of dynamic finger tapping recognition is lower and requires further improvement. Finally, we envision how new forms of interaction might be enabled with such a technique incorporated into wrist-worn devices.

ACKNOWLEDGEMENTS

This work was supported by JST CREST, under Grant No. JPMJCR17A3, Japan. We also gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan V GPU used in this research.

REFERENCES

[1] A. F. Bobick and J. W. Davis. 2001. The recognition of human movement using temporal templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 23, 3 (March 2001), 257–267. DOI: http://dx.doi.org/10.1109/34.910878

[2] Liwei Chan, Yi-Ling Chen, Chi-Hao Hsieh, Rong-Hao Liang, and Bing-Yu Chen. 2015. CyclopsRing: Enabling Whole-Hand and Context-Aware Interactions Through a Fisheye Ring. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology (UIST '15)*. ACM, New York, NY, USA, 549–556. DOI: http://dx.doi.org/10.1145/2807442.2807450

[3] Feiyu Chen, Jia Deng, Zhibo Pang, Majid Baghaei Nejad, Huayong Yang, and Geng Yang. 2018. Finger Angle-Based Hand Gesture Recognition for Smart Infrastructure Using Wearable Wrist-Worn Camera. *Applied Sciences* 8, 3 (2018). DOI: http://dx.doi.org/10.3390/app8030369

[4] Artem Dementyev and Joseph A. Paradiso. 2014. WristFlex: Low-power Gesture Input with Wrist-worn Pressure Sensors. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology (UIST '14)*. ACM, New York, NY, USA, 161–166. DOI: http://dx.doi.org/10.1145/2642918.2647396

[5] J. Donahue, L. A. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, T. Darrell, and K. Saenko. 2015. Long-term recurrent convolutional networks for visual recognition and description. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2625–2634. DOI: http://dx.doi.org/10.1109/CVPR.2015.7298878

[6] H. Gammulle, S. Denman, S. Sridharan, and C. Fookes. 2017. Two Stream LSTM: A Deep Fusion Framework for Human Action Recognition. In *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*. 177–186. DOI:http://dx.doi.org/10.1109/WACV.2017.27

[7] Jun Gong, Xing-Dong Yang, and Pourang Irani. 2016. WristWhirl: One-handed Continuous Smartwatch Input Using Wrist Gestures. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology (UIST '16)*. ACM, New York, NY, USA, 861–872. DOI: http://dx.doi.org/10.1145/2984511.2984563

[8] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. 2017. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861* (2017).

[9] Y. Jang, S. Noh, H. J. Chang, T. Kim, and W. Woo. 2015. 3D Finger CAPE: Clicking Action and Position Estimation under Self-Occlusions in Egocentric Viewpoint. *IEEE Transactions on Visualization and Computer Graphics* 21, 4 (April 2015), 501–510. DOI: http://dx.doi.org/10.1109/TVCG.2015.2391860

[10] S. Ji, W. Xu, M. Yang, and K. Yu. 2013. 3D Convolutional Neural Networks for Human Action Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35, 1 (Jan 2013), 221–231. DOI: http://dx.doi.org/10.1109/TPAMI.2012.59

[11] David Kim, Otmar Hilliges, Shahram Izadi, Alex D. Butler, Jiawen Chen, Iason Oikonomidis, and Patrick Olivier. 2012. Digits: Freehand 3D Interactions Anywhere Using a Wrist-worn Gloveless Sensor. In *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology (UIST '12)*. ACM, New York, NY, USA, 167–176. DOI: http://dx.doi.org/10.1145/2380116.2380139

[12] Gierad Laput and Chris Harrison. 2019. Sensing Fine-Grained Hand Activity with Smartwatches. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI '19)*. ACM, New York, NY, USA, Article 338, 13 pages. DOI: http://dx.doi.org/10.1145/3290605.3300568

[13] Gierad Laput, Robert Xiao, and Chris Harrison. 2016. ViBand: High-Fidelity Bio-Acoustic Sensing Using Commodity Smartwatch Accelerometers. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology (UIST '16)*. ACM, New York, NY, USA, 321–333. DOI: http://dx.doi.org/10.1145/2984511.2984582

[14] Jhe-Wei Lin, Chiuan Wang, Yi Yao Huang, Kuan-Ting Chou, Hsuan-Yu Chen, Wei-Luan Tseng, and Mike Y. Chen. 2015. BackHand: Sensing Hand Gestures via Back of the Hand. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology (UIST '15)*. ACM, New York, NY, USA, 557–564. DOI: http://dx.doi.org/10.1145/2807442.2807462

[15] Jess McIntosh, Asier Marzo, and Mike Fraser. 2017. SensIR: Detecting Hand Gestures with a Wearable Bracelet Using Infrared Transmission and Reflection. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology (UIST '17)*. ACM, New York, NY, USA, 593–597. DOI: http://dx.doi.org/10.1145/3126594.3126604

[16] Leap Motion. 2019. LeapUVC Documentation. (2019). https://github.com/leapmotion/leapuvc/blob/master/LeapUVC-Manual.pdf.

[17] K. Ohnishi, A. Kanehira, A. Kanezaki, and T. Harada. 2016. Recognizing Activities of Daily Living with a Wrist-Mounted Camera. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 3103–3111. DOI: http://dx.doi.org/10.1109/CVPR.2016.338

[18] Manuel Prätorius, Dimitar Valkov, Ulrich Burgbacher, and Klaus Hinrichs. 2014. DigiTap: An Eyes-free VR/AR Symbolic Input Device. In *Proceedings of the 20th ACM Symposium on Virtual Reality Software and Technology (VRST '14)*. ACM, New York, NY, USA, 9–18. DOI:http://dx.doi.org/10.1145/2671015.2671029

[19] Raf Ramakers, Davy Vanacken, Kris Luyten, Karin Coninx, and Johannes Schöning. 2012. Carpus: A Non-intrusive User Identification Technique for Interactive Surfaces. In *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology (UIST '12)*. ACM, New York, NY, USA, 35–44. DOI:http://dx.doi.org/10.1145/2380116.2380123

[20] J. Rekimoto. 2001. GestureWrist and GesturePad: unobtrusive wearable interaction devices. In *Proceedings Fifth International Symposium on Wearable Computers*. 21–27. DOI: http://dx.doi.org/10.1109/ISWC.2001.962092

[21] T. Scott Saponas, Desney S. Tan, Dan Morris, Ravin Balakrishnan, Jim Turner, and James A. Landay. 2009. Enabling Always-available Input with Muscle-computer Interfaces. In *Proceedings of the 22Nd Annual ACM Symposium on User Interface Software and Technology (UIST '09)*. ACM, New York, NY, USA, 167–176. DOI: http://dx.doi.org/10.1145/1622176.1622208

[22] Karen Simonyan and Andrew Zisserman. 2014. Two-stream Convolutional Networks for Action Recognition in Videos. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 1 (NIPS'14)*. MIT Press, Cambridge, MA, USA, 568–576. http://dl.acm.org/citation.cfm?id=2968826.2968890

[23] Jie Song, Gábor Sörös, Fabrizio Pece, Sean Ryan Fanello, Shahram Izadi, Cem Keskin, and Otmar Hilliges. 2014. In-air Gestures Around Unmodified Mobile Devices. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology (UIST '14)*. ACM, New York, NY, USA, 319–329. DOI: http://dx.doi.org/10.1145/2642918.2647373

[24] Srinath Sridhar, Anders Markussen, Antti Oulasvirta, Christian Theobalt, and Sebastian Boring. 2017. WatchSense: On- and Above-Skin Input Sensing Through a Wearable Depth Sensor. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17)*. ACM, New York, NY, USA, 3891–3902. DOI: http://dx.doi.org/10.1145/3025453.3026005

[25] Y. Sugiura, F. Nakamura, W. Kawai, T. Kikuchi, and M. Sugimoto. 2017. Behind the palm: Hand gesture recognition through measuring skin deformation on back of hand by using optical sensors. In *2017 56th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE)*. 1082–1087. DOI: http://dx.doi.org/10.23919/SICE.2017.8105457

[26] C. Szegedy, Wei Liu, Yangqing Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. 2015. Going deeper with convolutions. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 1–9. DOI: http://dx.doi.org/10.1109/CVPR.2015.7298594

[27] Mingxing Tan and Quoc V Le. 2019. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. *arXiv preprint arXiv:1905.11946* (2019).

[28] Andrew Vardy, John Robinson, and Li-Te Cheng. 1999. The WristCam As Input Device. In *Proceedings of the 3rd IEEE International Symposium on Wearable Computers (ISWC '99)*. IEEE Computer Society, Washington, DC, USA, 199–. http://dl.acm.org/citation.cfm?id=519309.856464

[29] Hongyi Wen, Julian Ramos Rojas, and Anind K. Dey. 2016. Serendipity: Finger Gesture Recognition Using an Off-the-Shelf Smartwatch. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. ACM, New York, NY, USA, 3847–3851. DOI: http://dx.doi.org/10.1145/2858036.2858466

[30] In-kuk Yun, Je-In Yu, and Dae-Kwang Jung. 2016. Wearable device and method of operating the same. (Feb. 4 2016). US Patent App. 14/812,436.

[31] Yang Zhang and Chris Harrison. 2015. Tomo: Wearable, Low-Cost Electrical Impedance Tomography for Hand Gesture Recognition. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology (UIST '15)*. ACM, New York, NY, USA, 167–173. DOI: http://dx.doi.org/10.1145/2807442.2807480