

Gesture-based Bootstrapping for Egocentric Hand Segmentation

Anonymous CVPR submission

Paper ID 3420

Abstract

Accurately identifying hands in images is a key sub-task for human activity understanding with wearable first-person point-of-view cameras. Traditional hand segmentation approaches rely on a large corpus of manually labeled data to generate robust hand detectors. However, these approaches still face challenges as the appearance of the hand varies greatly across users, tasks, environments or illumination conditions. A key observation in the case of many wearable applications and interfaces is that, it is only necessary to accurately detect the user's hands in a specific situational context. Based on this observation, we introduce an interactive approach to learn a person-specific hand segmentation model that does not require any manually labeled training data. Our approach proceeds in two steps, an interactive bootstrapping step for identifying moving hand regions, followed by learning a personalized user specific hand appearance model. Concretely, our approach uses two convolutional neural networks: (1) a gesture network that uses pre-defined motion information to detect the hand region; and (2) an appearance network that learns a person specific model of the hand region based on the output of the gesture network. During training, to make the appearance network robust to errors in the gesture network, the loss function of the former network incorporates the confidence of the gesture network while learning. Experiments demonstrate the robustness of our approach with an F_1 score over 0.8 on all challenging datasets across a wide range of illumination and hand appearance variations, improving over a baseline approach by over 10%.

1. Introduction

We address the task of pixel-level hand detection using first person wearable cameras. From the perspective of human activity understanding, pixel-level detection of hand and arm regions is an important sub-task for higher levels of human activity understanding using a wearable camera. The exact shape and transformation of the hand holds important information about the task that is being performed.

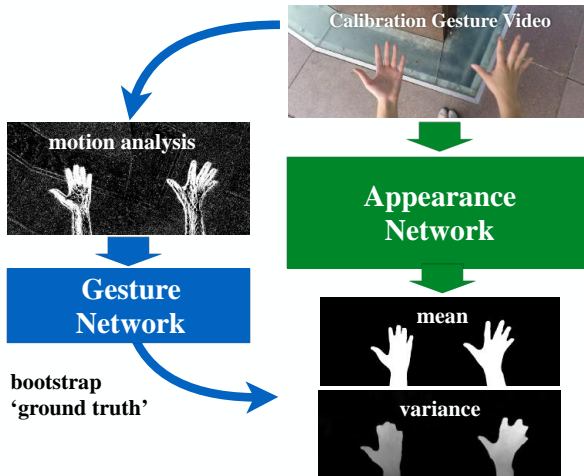


Figure 1: Our method bootstraps ground truth labels using motion analysis to train a personalized hand segmentation network.

In the context of virtual or augmented reality systems, the precise detection of hand regions is needed for realistic rendering and interactions with virtual objects. In the context of patient monitoring for neuromuscular rehabilitation, the exact hand shape (e.g. grasp or manipulation) is needed to accurately characterize the wearer's motor skills.

The task of building a model to detect hand regions with wearable cameras is daunting. First, the image of hands captured by first-person point-of-view cameras can exhibit very large variations in appearance. In general, there are no constraints on the environment in which the wearable camera might be used. The environment might include videos of a person playing a game in a dark room or playing outdoor sports under the bright sun. Training a single hand detector that can detect hands under all possible illumination conditions may be possible but would require a tremendous amount of labeled data. Second, the collection of pixel-level labels is labor intensive. Even with a large corpus of hand images from every possible scenario, labeling such data is currently not feasible. A single image of the hands at standard resolution can contain close to 1 million pixels that

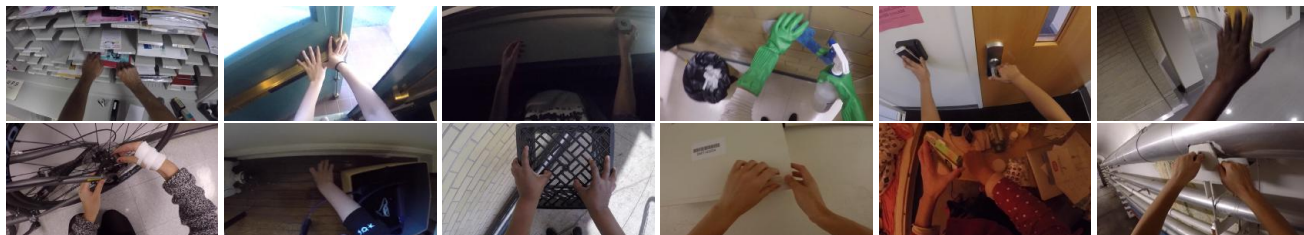


Figure 2: Sample images from our proposed dataset egocentric hand segmentation dataset. The dataset spans a diverse set of users across a variety of environments spanning multiple backgrounds and illumination variations.

must be manually labeled and can take several minutes to label. Based on these two challenges, it seems that training a hand detector that works for any person in any scenario will be quite challenging.

But do we really need to train a hand detector that works for all people and under any scenario? Is there anything else we can do to bootstrap a hand detection without gathering so much data? When we consider many of the use cases of wearable cameras, we make the two observations that hold the key to a more scalable approach. First, we observe that in many cases the hand detector usually only needs to work for one person – the camera wearer. In some cases, it only needs to work in a limited environment, such as the home for rehabilitation or a single room for a VR application. In other words, it is completely acceptable to learn a personalized hand detector. Second, we make the observation that in applications such as VR and AR, the sensing of the hands is part of an interactive process. This also means, that it is plausible to assume a certain level of user interaction and cooperation to help generate a better hand detection algorithm.

We take advantage of the appropriateness of personalization and the possibility of interaction, and propose a two stage deep network for hand detection (see Figure 1). The detector will be optimized for the user and will not require any manual image labeling. The training process is as follows. (1) The user with the wearable camera is instructed to perform a simple hand gesture. (2) Using optical flow and background subtraction results as input, a deep ‘gesture’ network is learned to estimate a foreground map (pixel-level mask of the hand and arm). (3) Images of the hand during the gesture interaction along with foreground maps are used to train an person-specific deep ‘appearance’ network. To deal with possible errors in the output of the gesture network, the loss function of the appearance network is weighted by the confidence of the gesture network.

Contributions. (1) We propose human interaction as a means of bootstrapping a hand segmentation algorithm for wearable camera. (2) We show how the uncertainty of the foreground masks generated by the gesture network can be incorporated to adaptively weight the loss function of the

appearance network. (3) We present a novel dataset (Figure 2) with 5000 images of 10 users across 30 environments, along with the respective calibration gesture, as a benchmark dataset for egocentric hand segmentation.

2. Related Work

There have been two predominant approaches to pixel level hand detection with wearable cameras. In *appearance-based methods*, skin color and texture are modeled to separate hand regions from background [16, 21, 20, 5, 12, 17, 2, 14, 18, 3, 25, 26]. Betancourt et.al. [4] demonstrate that identifying the left and right hands and modeling hand-to-hand occlusions can improve the accuracy of hand segmentation. Recently, Zhu et.al. [27] propose a two-stage detector that first generates hand bounding box proposals that are evaluated by a multi-task convolutional neural network in the second stage. This category of approaches are robust to extreme motion and motion blur since they rely only on the appearance of the hand. At the same time, these approaches do not generalize well to significant differences in skin color across different individuals or when in the presence of environmental (illumination conditions) variations.

Motion-based methods are another category of approaches that exploit the motion of the hand for detection. Methods that perform background modeling [9, 1] can be used with limited camera motion that can accounted for by video stabilization methods. Sheikh et.al. [19] use optical flow and frame differencing for background subtraction for freely moving cameras. These motion based approaches are naturally more robust to appearance variations and do not require training; however, hand appearance features are typically still needed for segmentation. Motion-based approaches are prone to fail in the presence of extreme motion or when hand textures are washed out under high saturation conditions (*i.e.*, direct sunlight).

In general, motion based methods are more suitable for hand detection under limited hand motion and in the presence of high individual appearance variations while appearance based methods using skin color and textures exhibit better performance when hand motion is unconstrained.

Therefore, much of the work in hand detection from first-person vision has focused on extending traditional methods, with little focus on the potential to leverage human interaction as part of the bigger picture of an interactive wearable system. As one of the first examples of gesture-based calibration [13] showed that a robust color space mapping combined with optical flow and region growing can be used to generate hand region detection. Although the detection of hands using appearance and motion is unsupervised, it still makes prior assumptions about the color of the hands as a pre-processing step [23] and cannot handle large variations away from standard skin color. Furthermore, the appearance-based detection algorithm uses a parametric Gaussian mixture model whose parameters in practice require manual tuning as the environment changes. In contrast, we use a supervised motion-based hand segmentation network which only needs to be trained once and it much more robust in practice. Furthermore, our appearance-based detection algorithm is trained discriminatively and does not require manual tuning of parameters across different environments or users. Our method is truly automatic.

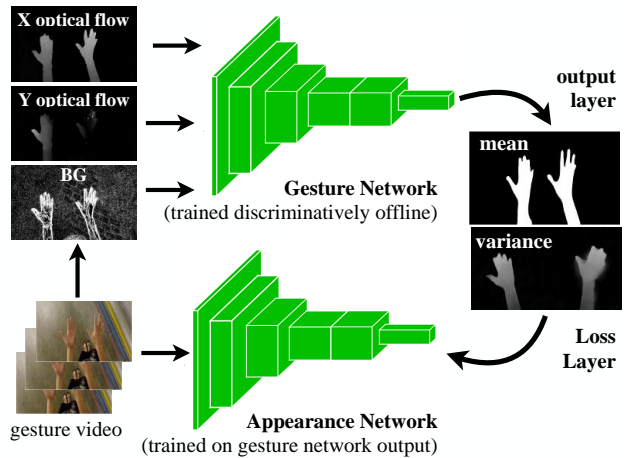
3. Our Approach

We focus on exploring the benefits of using cooperative human interaction to address the challenges of detecting hands across a diverse set of usage scenarios. In particular, we propose a two-step deep network to realize gesture-based bootstrapping for pixel-level hand and arm region segmentation. In the first step, we assume a predefined gestural interaction and only utilize motion information to achieve generalization across different users and varied environmental conditions. Specifically, we train gesture-based hand segmentation network by taking optical flow motion field and foreground probability map from background subtraction as input. Since we have strong priors over how the gesture will be performed, the trained detector is robust to any hand shape as well as hand appearance (e.g., the user can even be wearing a glove). In the second stage, we assume that the hand appearance of user will not change drastically over the duration of its use. The second appearance network utilizes the pixel-level notations from the output of the gesture network during the gesture demonstration. Those image frames are used to train a user specific customized hand region detector based only on appearance. An overview of our proposed interactive hand segmentation framework is illustrated in Figure 3.

3.1. Gesture-based Hand Segmentation

We utilize cooperative human interaction to build a personalized hand detection network. To facilitate this, we need a method for automatically gathering user specific “ground truth” segmentation masks of the user’s hands. In order to bootstrap the hand detection process, the user is

Appearance Network Training



Appearance Network Testing

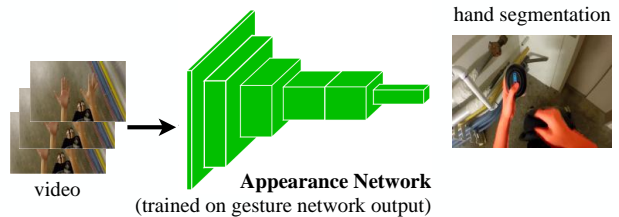


Figure 3: **Overview:** Our proposed two stage process for interactively learning personalized hand detection models. The person agnostic gesture network is designed for hand segmentation by analyzing the motion features of the interactive hand gesture. A person specific appearance network is then learned by bootstrapping the output of the gesture network as a proxy of the ground truth segmentation mask. The appearance network is designed to be robust to the errors of the gesture network by incorporating the confidence of the gesture network into its loss function. At test time the appearance network can then be used for hand segmentation across any unconstrained hand object interactions.

instructed to perform a simple hand gesture. To ensure that the bootstrapping process is successful, we consider free design aspects. (1) From the perspective of the user, the gesture should require the absolute minimum amount of effort to perform. (2) From a modeling point of view, the gesture should include both side of the hands in order to account for the variance in the user’s skin tone, as the palm (inner) surface is typically much lighter than the dorsal (outer) surface. (3) From an algorithmic perspective, the gesture should include motions that are easy for motion-based methods to analyze, e.g., not too fast, not too slow and does not cause the body to move too much. Based on these desiderata, the hand gestures are designed to include the following two steps: (1)

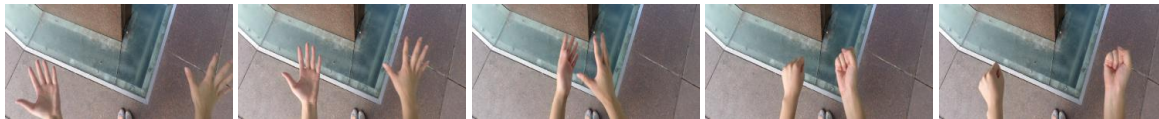


Figure 4: The sequence of our calibration gesture that we bootstrap to obtain ground truth hand segmentation masks to learn a user specific hand appearance network.

horizontal hand translation from side to middle with open palms, and (2) hand translation back from the center to side with closed fists. Figure 2 illustrates this calibration gesture sequence.

Stabilization: In order to segment the hand region and obtain precise motion information, the background in the frames should be steady. However, the videos that are captured are typically distorted and jittery due to the motion of a wide-angle camera mounted on a moving person. Therefore, we pre-process the gesture interaction videos. First we perform camera calibration to remove lens distortion, and then stabilize the videos to eliminate any unwanted vibration and camera motion. Camera stabilization proceeds in two steps. First, we utilize the method in [15] to eliminate of shaking. And second, we estimate the camera path by estimating the transformation between neighboring frames. This raw camera trajectory is smoothed out to estimate the transformation matrix that is applied to the frames. After getting rid of jittery motion, we apply perspective transformation on every frame in gesture interaction videos to align all frames to the first frame. We use the RANSAC [6] algorithm while estimating the transformation matrix to ignore the motion of the hands by treating hands as outliers. By matching the outliers ratio to the ratio of hand regions, we obtain robust a solution that works well across different backgrounds. The pre-processing steps are performed online and provide reliable input for further processing.

As stated above, the primary motivation of the gesture interaction step is to automatically generate ‘ground truth’ segmentation masks that will be used by the appearance network downstream. Since we have no prior information about the environment or appearance of the hands, we use a motion-based approach to segment moving hand regions as we are certain that hands are the only moving object following the path in accordance to the gesture instructions given to the user. First we obtain a coarse segmentation of the hands using two types of motion cues: (1) Optical flow using Horn and Shunck optical flow [10] to extract dense motion flow fields based on the brightness constancy constraint. Regions of high motion magnitude correspond to hand regions when the hand is moving., and (2) background subtraction [8] that enables statistical estimation of image backgrounds by applying bayesian inference to model a pixels probability of being a hand.

Gesture Network: Although the two methods described

above generally achieve reliable results under ideal conditions, each method suffers from its own limitations. To overcome these drawbacks and refine the segmentation masks, we propose a *Gesture Network* based on [24] (a semantic segmentation convolution neural network). The Gesture Network fuses the coarse segmentation results of optical flow field and background subtraction to estimate the hand region segmentation map. This network is trained on manually annotated pixel-level hand masks from a video dataset of calibration gestures performed across many different environments. This labeling process only needs to be performed once since the trained network is not dependent on the environment or appearance of the hand. The gesture network outputs a heat map over all pixels representing the probability of each pixel belonging to the hand region. It is important to emphasize here that the accuracy of this first stage is of utmost importance as it serves as the proxy ‘ground truth’ on which the second stage appearance network will be trained.

3.1.1 Characterizing Uncertainty

To shield the appearance network from the errors made by the gesture network we estimate the uncertainty in the output of the gesture network. To quantify the confidence/uncertainty at each pixel of the hand detection heat map results, we use dropout [22] at test time in the gesture network. Specifically, we apply dropout to various layers of the Gesture Network during training. Instead of calculating the ‘ensemble average’ of the neuron activations at test time, we also apply dropout at test time running the same input through the Gesture Network multiple times. By observing the changes in estimates of each pixel over time, we can obtain a Bayesian estimate of the covariance [7]. Figure 5 gives a visual example of the mean and variance of all pixels in a single image. The figure shows high variance in the space between fingers and near the contour of the hand, which is reasonable as those parts are often misclassified.

3.2. Appearance-based Hand Segmentation

Using the ‘ground truth’ output of the bootstrapped motion-based gesture network, we are now ready to train an appearance-based hand segmentation network which will be able to detect hands under any type of motion. Recall that we aim to train an appearance-based hand detector that



Figure 5: Visualization of output mask and variance over 100 samples. From left to right is original image, expectation mask, and variance mask

works for a specific user (not all possible users). Accompanied with ‘ground truth’ pixel notations from output of the gesture network, our appearance network takes the frames in gestural interaction video as input, and learns customized features of one particular user’s hands. Similar to the gesture network we use a modified version of ParseNet[24] as our appearance network to output a two class probability map.

In order to utilize the uncertainty of the gesture network, we design a weighted loss function using the precision matrix (inverse of the covariance matrix).

$$L = (\sigma(x) - t)^T \Sigma^{-1} (\sigma(x) - t)$$

where L is the loss function, Σ^{-1} represents the precision matrix, t is pixel notation in output of gestural network, x is output of appearance network, and $\sigma(x)$ is sigmoid function $\sigma(x) = \frac{1}{1+e^{-\alpha x}}$.

The partial derivative of loss function with respect to x is

$$\frac{\partial L}{\partial x} = \alpha \sigma(x) (1 - \sigma(x)) (\sigma(x) - t)^T \Sigma^{-1}$$

We know that the sigmoid function saturates fast. When $\sigma(x)$ is close to 1, the term $(1 - \sigma(x))$ will be close to 0, resulting in the magnitude of derivative term very small. The model is difficult to train under this condition. Therefore, we defined a soft sigmoid function $\sigma(x) = \frac{1}{1+e^{-\alpha x}}$, where α is a number less than 1. This soft version of the sigmoid function efficiently eliminates saturation, and also helps to speed up the training process.

In general, deep convolutional neural networks require a large amount of data to train the model. In our case, a typical gesture demonstration takes about 5 seconds and we extract about 180 consecutive frames from the center of the temporal window. The temporal window can easily be detected as the user is instructed to remove their hands from the view of the camera before and after the gesture motion. To address the limited amount of training data, we apply two strategies. First, we aggressively regularize the network through the use of early dropout during training. This is in contrast to standard ParseNet training which usually includes dropout layers only after the fully convolutional layers. By dropping out neurons in the convolutional layers, our appearance network is able to achieve optimal performance. Second, we implement data augmentation to

prevent over-fitting. Since the background during the gesture demonstration is limited to one scene, we also explore background augmentation by including other images of the surrounding environment.

4. Experimental Results

4.1. Personalized Hand Detection Dataset

Since existing egocentric activity datasets [5] [1] do not contain any consistent gestural inputs, we created a new dataset that is more appropriate for our interactive scenario. The dataset includes 10 users in 30 different environmental conditions (Figure 2). In each scenario, there are videos for training with gestural demonstrations and test videos including a wide range of various body and hand motions. Our dataset contains large variation over hand appearance and illumination conditions. Furthermore, the dataset also includes hand object interaction under extreme conditions, *e.g.* hands in very dim light, tasks with bandaged hands. To train our appearance-based baseline models, we utilize two publicly available benchmark datasets, CMU-EDSH [5] and GTEA [1] as the images are taken from first-person point-of-view and include interactions between hands and objects in varied forms.

4.2. Gesture Network Evaluation

4.2.1 Ablative Analysis

As discussed in Section 3.1, the video input of the gesture network is designed to capture the motion information of a predetermined gesture. In this experiment, we seek to understand which motion cues are most useful for detecting hand regions during a gesture demonstration. We perform ablative analysis across three input types: (1) background subtraction, (2) dense optical flow in the x direction and (3) dense optical flow in the y direction.

We performed five experiments using different combination of the three motion-based input types. Table 1 contains the F1 scores with respect to different combination of input types. The results show that best results are obtain when all

Input combination	F1 score
Bg-sub	0.924
Opt	0.919
Bg-sub + Opt x-direction	0.928
Bg-sub + Opt y-direction	0.931
Bg-sub + Opt x-direction + Opt y-direction	0.946

Table 1: Hand segmentation performance with different input combination, where Bg-sub represents foreground probability map calculated from background subtraction, and Opt represents optical flow



Figure 6: Visualization of input for gesture network

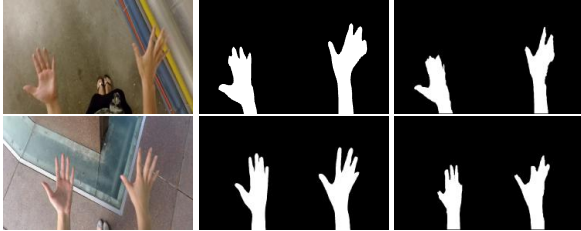


Figure 7: Visualization of output from proposed method on gestural interaction step and [13]. The first column is the frame image, the second is the output of our proposed method and the third column is the result of method in [13]

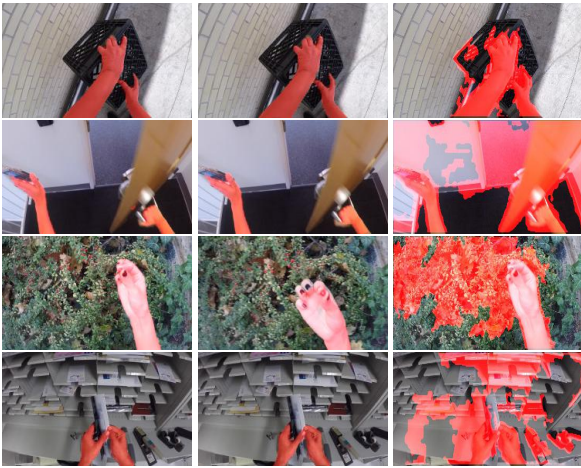


Figure 8: Output visualization of ParseNet, FCN, and Per-Pix hand detector from left to right

features are used, namely, background subtraction and optical flow in both directions yield the best gesture-based hand detection performance.

Figure 6 shows sample images of the foreground probability map computed from background subtraction and the optical flow field in the x and y direction, respectively. As one would expect, the optical flow works well to segment hand regions when the hands are in motion and the background subtraction is able to find hand contours even when there is very little motion.

Method	All	Test1	Test 2	Test 3
Proposed	0.946	0.938	0.941	0.952
[13]	0.853	0.858	0.842	0.857

Table 2: F1 score comparison against [13]

Method	Test1	Test2	Test3	Test4
ParseNet [24]	0.886	0.845	0.890	0.882
FCN [11]	0.872	0.852	0.889	0.865
PerPix [5]	0.453	0.665	0.144	0.564

Table 3: F1 score of Hand segmentation with ParseNet, FCN, and PerPix hand detector in [5]

4.2.2 Comparative Analysis

We now compare our performance to a state-of-the-art method [13]. As mentioned above [13] detects hand regions in an unsupervised manner based on both skin color statistics and motion information from optical flow. Table 2 illustrates F1 scores both averaged on all test sets (750 images in total) and independently on three videos (150 images for each) with varied hand appearances and illumination conditions. We also visualize output of motion-based segmentation for comparison in Figure 7.

4.2.3 Implementation Details

Due to limited images available to train the gesture network, we apply data augmentation on the input images. We implement data augmentation with three parts: (1) Crop: We randomly crop images to size 304×824 , and resize it back to 380×1030 . (2) Rotation: Considering the user’s head angle while performing gestural interaction task, we apply rotation with angle -10° and $+10^\circ$ on input images. (3) Flip: We flip images on horizontal direction.

We implement our gesture network based on architecture of ParseNet. We adopt fine tuning with pretrained model on VGG-16 as reasonable initialization. The size of network input is $K \times C \times 380 \times 1030$, where K is batch size and always set to 1, C is input channels which is related to the combination of input fields for the network.

4.3. Appearance Network Evaluation

4.3.1 Comparative Analysis

Inspired by state-of-art performance of convolutional neural network on image segmentation, we first explore performance of Convolutional Neural Network on hand segmentation from first-person point-of-view. The training set includes 2717 images with per-pixel hand mask from two public datasets, EDSH and GTEA.

We run experiments with ParseNet [24], Fully convolutional neural network (FCN) [11], and Perpих hand detector in [5]. We report our result on four test sets with different skin appearance and environments.

Table 3 illuminates F1 scores of ParseNet [24] and FCN [11] trained on 2717 images of datasets EDSH and GTEA. We also visualize the output of three methods in Figure 10. The result shows that convolutional neural network yields high accuracy on the task of hand segmentation in egocentric vision.

We design the network architecture of the appearance network based on the protocol of ParseNet. The network is initialized with the pretrained model from VGG-16. The appearance network takes frames from the gesture interaction video as input. For the loss layer, the pixel-wise segmentation masks are used in conjunction with the precision matrix (described in Section 3.1.1).

4.3.2 Data Augmentation Analysis

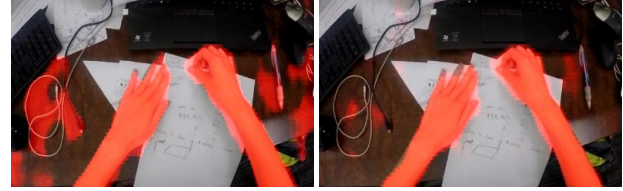
We implement three data augmentation strategies. *Transformation augmentation*: We apply cropping, rotation, flipping in this part. For crop, we randomly crop images to size 304×824 , and resize it back to 380×1030 . To enhance model ability of learning hands in different direction, we randomly apply rotation transformation with angle -30° , -15° , $+15^\circ$, $+30^\circ$. As for flipping, we randomly mirror images on horizontal direction.

Due to single background in gestural interaction video, we implement background augmentation by including 30 images without hand. We collect images by instructing user to look around before performing gesture. The background images enable the model to segment hand robustly. We visualize two output images before and after adding background images in Figure 9.

In order to enhance the model to adapt to illumination variation, we apply brightness augmentation on all images. We first calibrate the brightness level by transforming the images from the RGB color model to HSV color space where the value (V) represents brightness. We scale the value at different scales to simulate images in different illumination conditions. We analyze the performance of different data augmentation strategies on one test set with 177 images in Table 4.

4.3.3 Impact of Dropout on Training

As described in [22], dropout is a powerful technique to avoid over-fitting. By randomly removing units with certain ratio during training, the network achieve better performance as well as robustness to small amount of data. In our case, the number of input images is around 150. Therefore, we implement a dropout layer on early stages to tackle the problem of limited training images. We show the result on



(a) before augmentation (b) after augmentation

Figure 9: Background Augmentation Results.

Data Augmentation strategy	F1 score
No augmentation	0.834
Brightness	0.852
Environment	0.867
Brightness + Transformation	0.854
Environment + Transformation	0.865
Brightness + Transformation + Environment	0.869

Table 4: Performance evaluation of data augmentation strategy

Vid.	1	2	3	4	5	6	7
F1	0.823	0.844	0.842	0.868	0.860	0.876	0.875

Table 5: Performance of multiple gestural interaction videos as input

one test set with 177 images. The training set is augmented with brightness and transformation.

4.3.4 Data Dependency Analysis

Since one gesture interaction video only provides around 150 images, more videos will potentially increase data diversity to benefit the model. We show performance of increasing gestural interaction video numbers in Table 5. We observe that, with more calibration videos, the accuracy of the model increases.

4.3.5 Impact of Modeling Uncertainty

We defined the loss function of the appearance network in Section 3.2 that takes into account the precision matrix of the gesture network. In our case, the precision matrix can be interpreted in two forms. When the matrix is the identity matrix, the appearance network does not consider uncertainty in the gesture network. If we consider the variance if each pixel, the matrix is diagonal. The F1 score without considering uncertainty is 0.842, while the score considering uncertainty is 0.859.

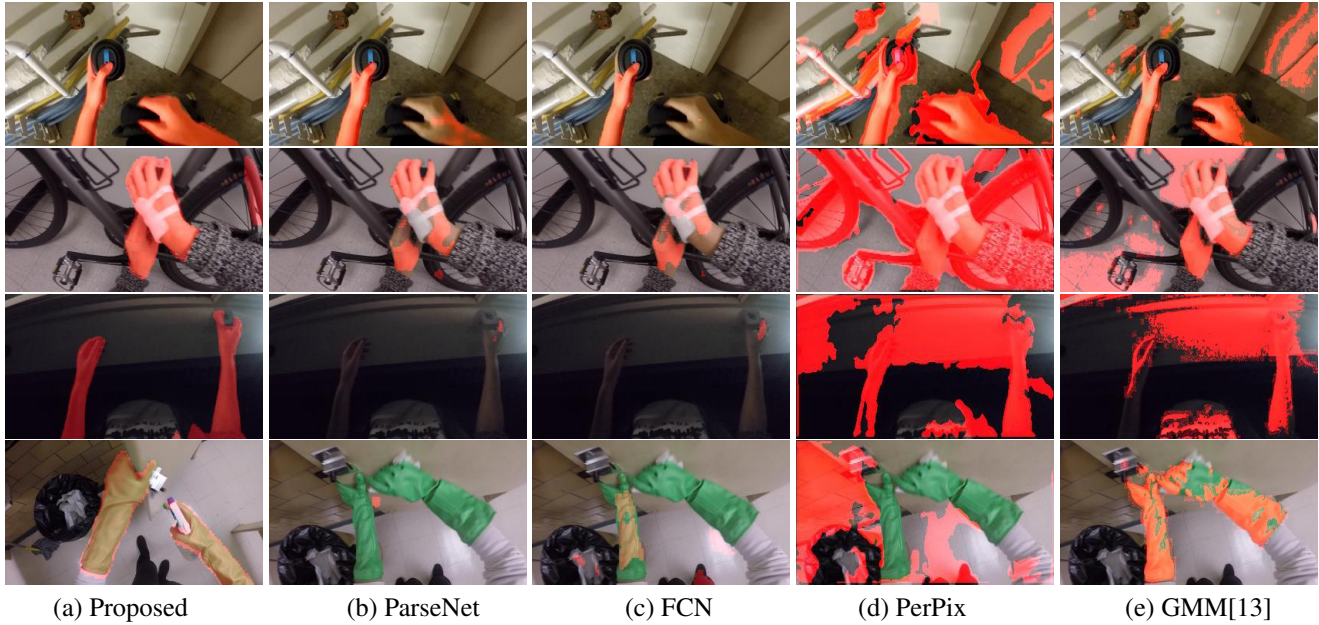


Figure 10: Output visualization of proposed method and baselines

Method	test1	test2	test3	test4
Proposed	0.853	0.803	0.842	0.897
ParseNet [24]	0.852	0.598	0.203	0.01
FCN [11]	0.845	0.572	0.213	0.172
PerPix [5]	0.661	0.625	0.003	0.002
GMM [13]	0.736	0.213	0.326	0.732

Table 6: F1 score of hand segmentation performance against baseline method

4.3.6 Performance on Challenging Datasets

We present a performance comparison of our proposed method and against several baselines in Table 6. We report results for four challenging test sets, each including 50 images, *e.g.*, dark environment, hand with bandage, tasks with gloves. The visualization of the output is illustrated in Figure 10. On the test videos under normal conditions, a standard convolutional neural network achieves remarkable performance on hand segmentation. However, under extreme conditions *e.g.* work in very dark room or action with gloves, the pretrained models will fail due to a large difference between the training distribution and the test distribution. In contrast, our personalized detector performs well with the help of gestural bootstrapping.

5. Conclusion

We have developed a two stage network, where the first step utilizes human interaction to realize gesture-based

bootstrapping to generate target labels. In the second stage, an appearance-based network uses those target, along with uncertainty estimates, to train a discriminative appearance-based hand segmentation. Our results show that by training a personalized hand detector for each individual user, the model is more robust and yields higher performance, even in the most challenging environments.

References

- [1] J. M. R. Alireza Fathi, Xiaofeng Ren. Learning to recognize objects in egocentric activities., 2011. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR).
- [2] A. A. Argyros and M. I. Lourakis. Real-time tracking of multiple skin-colored objects with a possibly moving camera. In *European Conference on Computer Vision*, pages 368–379. Springer, 2004.
- [3] A. Betancourt, M. M. López, C. S. Regazzoni, and M. Rauterberg. A sequential classifier for hand detection in the framework of egocentric vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 586–591, 2014.
- [4] A. Betancourt, P. Morerio, E. Barakova, L. Marcenaro, M. Rauterberg, and C. Regazzoni. Left/right hand segmentation in egocentric videos. *Computer Vision and Image Understanding*, 2016.
- [5] C. Li and K. M. Kitani. Pixel-level hand detection in egocentric videos. pages 3570–3577, 2013. In *Computer Vision and Pattern Recognition (CVPR)*, 2013 IEEE International Conference.
- [6] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image

864	analysis and automated cartography. <i>Communications of the</i>	networks from overfitting. <i>J. Mach. Learn. Res.</i> , 15(1):1929–	918
865	<i>ACM</i> , 24(6):381–395, 1981.	1958, Jan. 2014.	919
866	[7] Y. Gal and Z. Ghahramani. Dropout as a bayesian approx-	[23] J. Van De Weijer, C. Schmid, J. Verbeek, and D. Larlus.	920
867	imation: Representing model uncertainty in deep learning.	Learning color names for real-world applications. <i>IEEE</i>	921
868	<i>arXiv preprint arXiv:1506.02142</i> , 2015.	<i>Transactions on Image Processing</i> , 18(7):1512–1523, 2009.	922
869	[8] M. A. G. K. Godbehere, A. B. Visual tracking of human	[24] A. R. W. Liu and A. C. Berg. Parsenet: Looking wider to see	923
870	visitors under variable-lighting conditions for a responsive	better, 2015. <i>arXiv e-prints</i> , arXiv:1506.04579 [cs.CV].	924
871	audio art installation, 2012. In <i>ACC</i> .	[25] X. Zhu, X. Jia, and K.-Y. K. Wong. Pixel-level hand detec-	925
872	[9] E. Hayman and J.-O. Eklundh. Statistical background sub-	tion with shape-aware structured forests. In <i>Asian Confer-</i>	926
873	traction for a mobile observer. In <i>Computer Vision, 2003.</i>	<i>ence on Computer Vision</i> , pages 64–78. Springer, 2014.	927
874	<i>Proceedings. Ninth IEEE International Conference on</i> , pages	[26] X. Zhu, X. Jia, and K.-Y. K. Wong. Structured forests for	928
875	67–74. IEEE, 2003.	pixel-level hand detection and hand part labelling. <i>Computer</i>	929
876	[10] B. K. P. Horn and B. G. Schunck. Determining optical flow.	<i>Vision and Image Understanding</i> , 141:95–107, 2015.	930
877	<i>ARTIFICIAL INTELLIGENCE</i> , 17:185–203, 1981.	[27] X. Zhu, W. Liu, X. Jia, and K.-Y. K. Wong. A two-stage	931
878	[11] T. D. Jonathan Long, Evan Shelhamer. Fully convolu-	detector for hand detection in ego-centric videos. In <i>2016</i>	932
879	tional networks for semantic segmentation, 2015. The IEEE	<i>IEEE Winter Conference on Applications of Computer Vision</i>	933
880	Conference on Computer Vision and Pattern Recognition	(WACV), pages 1–8. IEEE, 2016.	934
881	(CVPR).		935
882	[12] P. Kakumanu, S. Makrogiannis, and N. Bourbakis. A sur-		936
883	vey of skin-color modeling and detection methods. <i>Pattern</i>		937
884	<i>recognition</i> , 40(3):1106–1122, 2007.		938
885	[13] L. Q. K. S. B. E. B. R. Kumar, J. On-the-fly hand detec-		939
886	tion training with application in egocentric action recogni-		940
887	tion. 2015. In <i>Com-puter Vision and Pattern Recognition</i>		941
888	<i>Workshop (CVPRW)</i> .		942
889	[14] C. Li and K. M. Kitani. Model recommendation with virtual		943
890	probes for egocentric hand detection. In <i>Proceedings of the</i>		944
891	<i>IEEE International Conference on Computer Vision</i> , pages		945
892	2624–2631, 2013.		946
893	[15] V. K. Matthias Grundmann and I. Essa. Auto-directed video		947
894	stabilization with robust 11 optimal camera paths., 2011.		948
895	In <i>Computer Vision and Pattern Recognition (CVPR)</i> , 2011		949
896	IEEE Conference on.		950
897	[16] M.J.Jones and J.M.Rehg. Statistical color models with appli-		951
898	cation to skin detection. <i>International Journal of Computer</i>		952
899	<i>Vision</i> , 46(1):81–96, 2002.		953
900	[17] P. Pérez, C. Hue, J. Vermaak, and M. Gangnet. Color-based		954
901	probabilistic tracking. In <i>European Conference on Computer</i>		955
902	<i>Vision</i> , pages 661–675. Springer, 2002.		956
903	[18] G. Serra, M. Camurri, L. Baraldi, M. Benedetti, and R. Cuc-		957
904	chiara. Hand segmentation for gesture recognition in ego-		958
905	vision. In <i>Proceedings of the 3rd ACM international work-</i>		959
906	<i>shop on Interactive multimedia on mobile & portable de-</i>		960
907	<i>vices</i> , pages 31–36. ACM, 2013.		961
908	[19] Y. Sheikh, O. Javed, and T. Kanade. Background subtraction		962
909	for freely moving cameras. In <i>2009 IEEE 12th International</i>		963
910	<i>Conference on Computer Vision</i> , pages 1219–1225. IEEE,		964
911	2009.		965
912	[20] L. Sigal, S. Sclaroff, and V. Athitsos. Skin color-based		966
913	video segmentation under time-varying illumination. <i>IEEE</i>		967
914	<i>Transactions on Pattern Analysis and Machine Intelligence</i> ,		968
915	26(7):862–877, 2004.		969
916	[21] A. S.L.Phung and D.Chai. Skin segmentation using color		970
917	pixel classification: analysis and comparison. <i>Pattern Anal-</i>		971
	<i>ysis and Machine Intelligence</i> , 27(1):148–154, 2005.		
	[22] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and		
	R. Salakhutdinov. Dropout: A simple way to prevent neural		