Recognition from Hand Cameras: A Revisit with Deep Learning

Cheng-Sheng Chan, Shou-Zhong Chen, Pei-Xuan Xie, Chiung-Chih Chang, and Min Sun(✉)

Departmant of Electrical Engineering, National Tsing Hua University, Hsinchu, Taiwan
{s104061526,s104061545}@m104.nthu.edu.tw,
{s101061230,s101060006}@m101.nthu.edu.tw, sunmin@ee.nthu.edu.tw

Abstract. We revisit the study of a wrist-mounted camera system (referred to as HandCam) for recognizing activities of hands. HandCam has two unique properties as compared to egocentric systems (referred to as HeadCam): (1) it avoids the need to detect hands; (2) it more consistently observes the activities of hands. By taking advantage of these properties, we propose a deep-learning-based method to recognize hand states (free vs. active hands, hand gestures, object categories), and discover object categories. Moreover, we propose a novel two-streams deep network to further take advantage of both HandCam and HeadCam. We have collected a new synchronized HandCam and HeadCam dataset with 20 videos captured in three scenes for hand states recognition. Experiments show that our HandCam system consistently outperforms a deep-learning-based HeadCam method (with estimated manipulation regions) and a dense-trajectory-based HeadCam method in all tasks. We also show that HandCam videos captured by different users can be easily aligned to improve free vs. active recognition accuracy (3.3 % improvement) in across-scenes use case. Moreover, we observe that finetuning Convolutional Neural Network consistently improves accuracy. Finally, our novel two-streams deep network combining HandCam and HeadCam achieves the best performance in four out of five tasks. With more data, we believe a joint HandCam and HeadCam system can robustly log hand states in daily life.

Keywords: Activity recognition · Wearable camera

1 Introduction

Recently, the technological advance of wearable devices has led to significant interests in recognizing human behaviors in daily life (i.e., uninstrumented environment). Among many devices, egocentric camera systems have drawn significant attention, since the camera is aligned with the wearer's field-of-view, it naturally captures what a person sees. These systems have shown great potential in recognizing daily activities (e.g., making meals, watching TV, etc.) [1], estimating hand poses [2,3], generating how-to videos [4], etc.

B. Leibe et al. (Eds.): ECCV 2016, Part IV, LNCS 9908, pp. 505–521, 2016.
DOI: 10.1007/978-3-319-46493-0_31

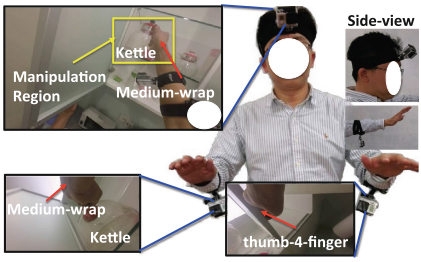

Fig. 1. Illustration of our wearable camera system: consisting of three wide-angle cameras, two mounted on the left and right wrists to capture hands (referred to as HandCam) and one mounted on the head (referred to as HeadCam). We use our HandCam system to robustly recognize object categories (e.g., kettle) and hand gestures (see red arrows for illustration). (Color figure online)

Despite many advantages of egocentric camera systems, there exists two main issues which are much less discussed [2]. Firstly, hand localization is not solved especially for passive camera systems. Even for active camera systems like Kinect, hand localization is challenging when two hands are interacting or a hand is interacting with an object. Secondly, the limited field-of-view of an egocentric camera implies that hands will inevitably move outside the images sometimes. On the other hand, cameras have been mounted on other locations to avoid similar issues. In project Digit [5], a camera is mounted on a user's wrist to always observe the user's hand pose. This allows a user to issue gesture commands at any time. Similarly, a camera has been mounted on a robot arm for accurately picking up an object [6]. In fact, the seminal work [7] has conducted simulation on 3D model of the human body to analyze the effects of field of view and body motion, when cameras are mounted at different locations. Hence, we argue that egocentric camera system might not be the best wearable system for recognizing human behaviors.

We revisit the wrist-mounted camera system (similar to [8]) to capture activities of both hands (Fig. 1). We name our system "HandCam" which is very different from egocentric systems with cameras on head or chest (e.g., [2,4,9]). By wearing cameras on wrists, HandCam directly recognizes the states of hands (e.g., object: kettle; gesture: medium-wrap in Fig. 1). It avoids the needs to detect hands and infer manipulation regions as required in classical egocentric systems [9]. A few methods have been proposed to recognize activities using wrist-mounted camera [8,10]. They show that wrist-mounted sensor system can be small and user-friendly. They also primarily focus on fusing different sensing modalities. However, we focus on designing a deep-learning-based vision algorithm to improve recognition accuracy (see Sect. 2.3 for more comparison). Most importantly, we are one of the first to propose a novel two-streams deep network taking advantages of both HandCam and HeadCam. All our methods are design to classify hand states including free vs. active (i.e., hands holding objects or

not), object categories, and hand gestures (Sect. 3.3). A similar method is also proposed to discover object categories in an unseen scene (Sect. 3.7).

To evaluate our system, we collected a new synchronized HandCam and HeadCam dataset for hand state recognition. The dataset consists of 20 sets of video sequences (i.e., each set includes two HandCams and one HeadCam synchronized videos) captured in three scenes: a small office, a mid-size lab, and a large home. In order to thoroughly analyze recognition tasks, we ask users to interact with multiple object categories and multiple object instances. We also ask multiple users to wear HandCam in a casual way to consider the variation introduced by multiple users. To overcome this variation, a fully automatic hand alignment method is proposed (Sect. 3.2).

Experiments show that our HandCam system consistently outperforms a deep-learning-based HeadCam method (with estimated manipulation regions [11]) and a dense-trajectory-based [12] HeadCam method in all tasks. Moreover, we show that HandCam videos captured by different users can be easily aligned to improve free vs. active recognition accuracy (3.3 % acc. improvement) in across-scenes use case. In all experiments, we use state-of-the-art Convolutional Neural Network (CNN) [13] features. We observe that finetuning CNN consistently improves accuracy (at most 4.9 % improvement). Finally, our method combining HandCam and HeadCam features achieves the best performance.

2 Related Work

A few non-vision-based methods have been proposed to recognize human daily activities based on recognizing hand states [14,15]. Wu et al. [16] combine sparse RFID data with a third-person video to recognize human daily activities based on objects used. In the following, we focus on the vision-based methods and review related work in egocentric recognition, hand detection and pose estimation, and a few works inspired our HandCam system.

2.1 Egocentric Recognition

[9,17,18] are the early egocentric works learning to recognize objects, actions, and activities. These methods assume foreground objects and hands can be easily separated from background using appearance, geometric, and motion cues. Their methods are evaluated on an egocentric activity dataset where the users move mainly their hands in front of a static table. In contrast, we allow users to naturally move to different places in a scene, which creates addition challenge for egocentric system to localize hands. For instance, Pirsiavash and Ramanan [1] also propose to recognize activities through recognizing objects while users is moving in their homes. Since their approach is based on detecting hand-object manipulation in the egocentric field-of-view, it is confused mainly with activities observing similar objects in the field-of-view without manipulation. However,

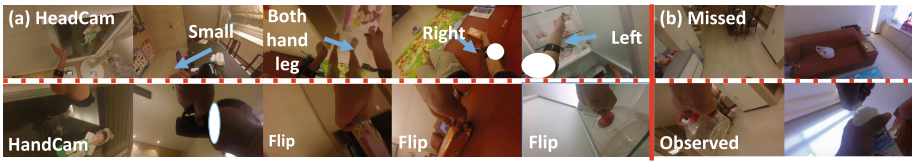

Fig. 2. HandCam (bottom-row) vs. HeadCam (top-row). Panel (a) compares the hand location variation. The variation in HandCam (bottom) is significantly less than variation in HeadCam (top). We also know exactly which video captures left or right hand. We flip left hand images to mimic right hand images and train a single deep network. Panel (b) shows typical examples of missed hands in HeadCam but observed hands in HandCam. For example, we do not require consistent hand-eye coordination for opening a water bottle while walking.

since our HandCam significantly reduces hand location variation (Fig. 2(a)), this scenario won't be a big issue for our HandCam system.

[11,19,20] further show the importance of gaze to help recognizing actions requiring "hand-eye coordination". We argue that not all daily activities consistently requires hand-eye coordinate. For instance, we do not require consistent hand-eye coordination for opening a water bottle while walking (Fig. 2-b-Right). In such case, head movement and gaze information might be misleading, and the user's hand and object of interest might move outside the field-of-view of the egocentric camera (Fig. 2(b)). On the other hand, our HandCam system more consistently captures hand-object interaction for a variety of daily activities. Finally, a few works [21–23] focus on summarizing egocentric videos by recognizing objects, people, and scenes.

Extra Sensors. Fernando et al. [24] utilize motion capture techniques, static cameras, wearable IMUs, and a head-mounted camera to study food preparation process in an instrumented kitchen. [4] proposes to combine egocentric camera with gaze tracker to robustly "discover" objects of interest given multiple sequences recorded in the same scene conducting the same tasks by different subjects. Moghimi et al. [25] propose to use a head-mounted RGBD camera to recognize both manipulation and non-manipulation activities. Damen et al. [26] also propose to use RGBD camera to model background and "discover" foreground objects. In this work, we show that with our HandCam system, objects can also be discovered without the need of gaze tracker or RGBD sensor (Sect. 3.7).

2.2 Hand Detection and Pose Estimation

[2,3] focus on estimating 3D hand poses using wearable RGBD camera. Despite many success in 3D hand pose recognition, Rogez et al. [2] show that egocentric 3D hand poses estimation is very challenging due to common interaction between hands and other objects or scene elements. [27–29] focus on detecting hand pixels in RGB images while users are moving around various environments. Betancourt et al. [30] study the weakness of [27] and proposes method for reducing false

positive detection of hands. Although these RGB methods are not as robust as RGBD methods, these methods have been applied to discover hand gestures [31].

2.3 Camera for Hands

A few work have proposed to wear cameras on wrists or other body parts to recognize gestures, poses, and activities of hands. In [5,32], cameras are mounted on a user's wrists to always observe user's hand pose. This allows a user to issue gesture commands at any time. However, the project assumes that a user is hand free of objects. In contrast, our HandCam system focuses on recognizing hand-object interactions. Similarly, a camera has been mounted on a robot arm for it to accurately pick up an object [6]. Although the robot has other sensors like a stereo camera and a lazer range finder which are not mounted on the robot arm, it has been shown that the HandCam is essential for picking up an object. Chan et al. [33] recently propose to wear a camera on hand webbings to recognize hand gestures and context-aware interactions such as events triggered by object recognition. However, they assume that objects are instrumented with QR codes. These works suggest that egocentric camera systems might not be the only wearable options for understanding human behaviors.

Maekawa et al. [8] is the most relevant prior work aiming for object-based activity recognition using sensors on wrist including a camera. We share the same idea to take advantage of wrist-mounted camera to recognize human-object interaction. However, it focuses on fusing the observation of heterogeneous sensors including a camera, a microphone, and an accelerometer. Compared to our deep learning approach, they utilize simple and efficient color histogram as the feature. Moreover, they train/test in the same environment and use same object instances, whereas we train/test different object instances across different environments (e.g., train: lab+office; test: home). Ohnishi et al. [34] present a recent paper that achieves an outstanding recognition accuracy using a wrist-mounted camera (only on right hand). They also focus on vision approach using deep features and dense-trajectory-based features [12]. Our HandCam method does not use motion feature [12], since it is time consuming to compute (on average a few seconds for each prediction). On the other hand, our deep feature can be computed in real-time on a GPU. Moreover, they use pre-trained CNN feature only, whereas we train a novel two-streams CNN to learn representation for wearable cameras. Finally, in their experiments, they assume that the temporal segment of each action is given. Hence, they evaluate per-segment classification accuracy, whereas we evaluate per-frame classification accuracy.

3 Our System

Our wearable camera system (Fig. 1) consists of three wide-angle cameras: two HandCams and one HeadCam. We first propose a method utilizing deep-learning techniques to classify hand states (free vs. active, object categories, and hand gestures) observed by either HandCam or HeadCam separately. Finally, we propose a novel two-streams CNN model to take advantage of all cameras in Sect. 3.8.

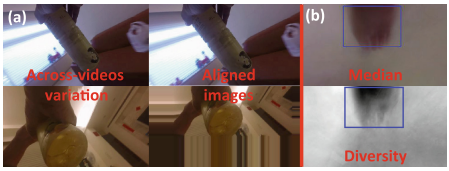

Fig. 3. Across-videos Hand Alignment. Panel (a)-left shows the across-videos hand variation. Panel (a)-right shows aligned images. Panel (b) shows example of median and diversity images on the top and bottom, respectively.

3.1 Wearable Cues

The strength of a wearable system essentially lies in the unique cues it can extract. For an egocentric system, these cues include gaze, hand, and foreground information. Some systems utilize active sensors to reliably localize hands (e.g., RGBD sensor in [3]) or predict user's attention (e.g., gaze tracker in [4]). However, they require extra hardware expenses and more power consumption. Many other egocentric systems require only a camera. However, sophisticated pre-processing steps [9,11,17,20] are required for removing background information.

Our HandCam system is designed with two focuses:

- Stable Hand Cue. Significantly reduced hand location variation (Fig. 2(a)-bottom) as compared to egocentric systems which have larger hand location variation (Fig. 2(a)-top). Our system also typically won't be confused between left and right hands, since they are recorded by different cameras. Moreover, we augment our dataset by flipping left hand images to mimic right hand images. We found that the data augmentation procedure [35] improves the accuracy of our deep network.
- Consistent Observation. Almost all hand related activities are observed as compared to egocentric systems which have limited field-of-view that missed some hand related activities (Fig. 2(b)).

Human Factors. As we design our system to be used by general users, we let each user to wear the HandCam under a general guideline. As a consequence, different users mount the HandCam with slightly different distances and orientation. Therefore, the hand location variation "across" video sequences (Fig. 3(a)-left) is noticeable. However, once the camera is mounted on a user's wrists, the spatial variation of hand regions are small within the video. By utilizing this fact, we propose a fully automatic across-videos hand alignment method.

3.2 Hand Alignment

In each video sequence, we model the pixel value (i.e., a value between $0 \sim 255$) distribution across time for each pixel and each color channel as a Laplace distribution parameterized by its center (μ) and its diversity (β). We estimate the

Fig. 4. Typical hand states in HandCam view: object category (top-white-font) and hand gesture (bottom-red-font). The statisitcs of states in our dataset in shown in Fig. 8. (Color figure online)

parameters of the distribution using maximum likelihood estimators, where the median image represents the common pattern (Fig. 3(b)-top) and the diversity image represents the variation of pixel values (black indicates small variation in Fig. 3(b)-bottom). We simply treat the pixels with diversity β smaller than β_{th} for all color channel as "stable" hand mask (within blue box in Fig. 3(b)-bottom). We find the video with the smallest "stable" hand mask across all videos as the reference video, and use the median image within the mask as alignment template (blue box in Fig. 3(b)-top). We apply multiscale normalized cross-correlation to align the template to the median images of other videos. Then, we apply cropping and replicate padding to generate hand aligned images (Fig. 3(a)-right).

3.3 Hand States Recognition

Given the aligned (Fig. 3(a)-right) and stable (Fig. 2(b)-bottom) observation of hand, we propose to recognize the hand states for multiple tasks (Fig. 4).

Free vs. Active. The most fundamental states of interests is to recognize whether the hand is manipulating an object (referred to as "active state"), or not (referred to as "free state"). In this work, we explicitly evaluate the performance of active hands recognition undergoing some unseen activities.

Hand Gesture Recognition. At a slightly finer level of granularity (12 gesture classes shown in technical report [36]), we propose to recognize hand gestures of active hands. Note that gesture is an important affordance cue for recognizing activities without the need to recognize a large number of object categories involving in the same activity.

Object Category Recognition. At the finest level of granularity (23 object categories), we propose to recognize object categories which have been manipulated by hands. Categorical recognition allows our system to recognize an unseen object instance within a know category.

We take a fully supervised approach and train a frame-based multiclass state classifier. The classifier generates a confidence $u(s_i)$ for state s_i in the i^{th} frame. For example, $u(s_i = Active)$ specifies the confidence that the i^{th} frame contains an active hand. $u(s_i = Notebook)$ specifies the confidence that the i^{th} frame contains a hand manipulating a notebook. We take advantage of the recent breakthrough in deep learning and extract frame-based feature f_i from Convolutional

Neural Network (CNN) [13], where i denotes the frame index. The deep feature f_i is a high-level representation used as the input of the state classifier and the state change detector described next. We describe the setting of the CNN model that we use in our application in Sects. 3.6 and 3.8.

3.4 State Change Detection

Since frame-based state recognition inevitably will contain spurious error predictions, we propose to detect possible state changes.

Frame-Based Change. We train a frame-base binary state change classifier (i.e., change or no change) by treating frames within d frames distance away from a ground truth change frame as positive examples and remaining frames as negative examples. The large value of d will increase the number of positive examples, but decreasing the localization ability of the true change locations. In order to reduce the visual variation of changes, we propose the following feature,

$$cf_i = |f_{i-d} - f_{i+d}| = |f_{i+d} - f_{i-d}|, \tag{1}$$

where f is the same deep feature used for state classifier. Note that f is a high-level semantic feature. Hence, cf measures semantic changes, but not low-level motion or lighting condition changes. Moreover, cf implies that transition from active to free should have a similar feature representation as transition from free to active. Given cf, we apply a change classifier to obtain frame-based state change confidences for all frames (Fig. 5).

Change Candidates. Similar to classical edge detection [37], we need to remove redundant change candidates with high confidences. Hence, we apply non-maximum suppression to find local maximum with respect to state change confidences (Fig. 5). We define the local maximum locations as change candidates $\{i\}_{i \in C}$, where C contains a set of local maximum change locations. Note that we prefer high recall (i.e., >95 %) at this step.

3.5 Full Model

We now combine frame-based state classification with detected change candidates to improve the classification of states. Both information are captured into a pairwise scoring function below,

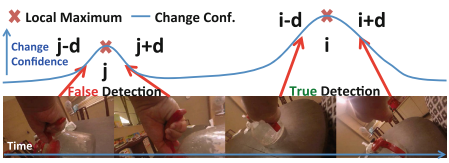

Fig. 5. Illustration of state change detection. Two change candidates are shown, where the first one corresponds to a false detection and the second one corresponds to a true detection. Our system ensures high recall of state changes at this step.

$$R(S) = \sum_{i=1}^{N} u(s_i) + \lambda \sum_{i=1}^{N-1} b(s_i, s_{i+1}), \tag{2}$$

where $R(S)$ is the score as a function of a set of states $S = [s_1, s_2, ..., s_i, ...]$, i is the index of frame, s_i is the state of the i^{th} frame, the space of s_i is $\{state1, state2, ...\}$, N is the total number of frames, λ balances the potentials, and $u(.), b(.)$ are the unary and binary scoring functions, respectively.

Scoring Functions. The unary scoring function is exactly the same as the scores in Sect. 3.3. The binary scoring function is defined below,

$$\textit{for } i \notin C\textit{; if } s_i \neq s_{i+1},\ b(s_i, s_{i+1}) = -\inf;\ \textit{otherwise, } b(s_i, s_{i+1}) = 0, \tag{3}$$

which means no change is allowed when the i^{th} frame is not a change candidate;

$$\textit{for } i \in C\textit{; if } s_i \neq s_{i+1},\ b(s_i, s_{i+1}) = -S(\bar{f}_i, \bar{f}_{i+1});\ \textit{otherwise, } b(s_i, s_{i+1}) = S(\bar{f}_i, \bar{f}_{i+1}),$$

where $S(\bar{f}_i, \bar{f}_{i+1})$ is the cosine similarity between $\bar{f}_i, \bar{f}_{i+1}$, $\bar{f}_i$ is the average frame-based deep features between the change candidate immediately before the i^{th} frame and the change candidate at the i^{th} frame, and $\bar{f}_{i+1}$ is the average frame-based deep features between the change candidate at the i^{th} frame and the change candidate immediately after the i^{th} frame. We apply a dynamic programming inference procedure to predict the states maximizing $R(S)$.

3.6 Deep Feature

We extract our deep feature from the well-known AlexNet [13][1] CNN model. Instead of using the pre-trained $1K$ dimension final output as feature, we try

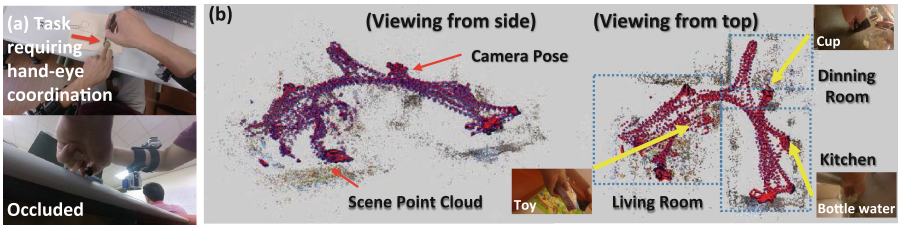

Fig. 6. Panel (a) shows an example where HandCam (bottom) is occluded but HeadCam (top) observed the activity requiring hand-eye coordination. Panel (b) shows 3D Structure [38] of the Scene reconstructed from HeadCam images. A pair of blue and red dots indicates the recovered camera 3D pose and other color-coded dots show the color of the scene. This shows that HeadCam contains place information which potentially is useful for hand states recognition. (Color figure online)

[1] VGG [39] can be used to achieve $1-2\%$ improvement in general.

different design choices to address the following questions: (1) which layer should we extract feature? and (2) will finetuning improve recognition from the new HandCam observation? In our pilot experiment, we found that a compact six layers model achieves the best accuracy, while being more computationally efficient than the original AlexNet (see technical report [36]). Hence, we use the fc6 output of AlexNet by default in all experiments of this paper. In Sect. 6, we also show that finetuning consistently improves state prediction accuracy.

3.7 Object Discovery

Given many observation of how users interact with objects in a new scene, we propose a simple method to discover common object categories. Firstly, we predict the active hand segments which is typically over-segmented. Then, we calculate segment-base feature $\bar{f}$ as the average of the frame-based features and apply a hierarchical clustering method using cosine similarity to group similar segments into clusters. By assuming that the same object category is manipulated by hands multiple-times in a similar way, two similar segments likely corresponds to the same object categories. In Sect. 6.3, we show that our HandCam system can discover categories more accurately than a HeadCam system.

3.8 Combining HandCam with HeadCam

Since our goal is to achieve the best accuracy, we would like to combine HandCam with HeadCam to do even better. Intuitively, HeadCam should be complementary to HandCam in some ways. Firstly, sometimes HandCam is occluded by other objects, whereas HeadCam keeps a clear view of the hands (Fig. 6(a)) due to required hand-eye coordination. Second, HeadCam observed more scene/place information which might be useful. For instance, we have used the observation from HeadCam to reconstruct[2] the scene as well as localize the HeadCam in the scene as shown in Fig. 6(b). It is possible that certain place information observed by HeadCam can be beneficial for recognizing some hand states. We propose two approaches to combine HeadCam and HandCam.

Feature Concatenation. We simply concatenate the separately finetuned HeadCam and HandCam features. Then, we use the concatenate feature to train the state classifier and state change detector as described before. Although this is a straight forward approach, we show in our experiment that it already outperforms other settings.

Two-Streams CNN. Inspired by [40], we treat each camera as one unique stream and design a novel two-streams CNN. Each stream first feeds-forward through a CNN of six layers with the same first six layers in the AlexNet [13] architecture. Then, the fc6 (each with 4096 dimension) outputs of both streams are concatenated (total 8192 dimension) before forwarding to the next two fully connected layers. We use this two-streams CNN to predict Free/Active states,

[2] We use visualsfm [38] for reconstruction.

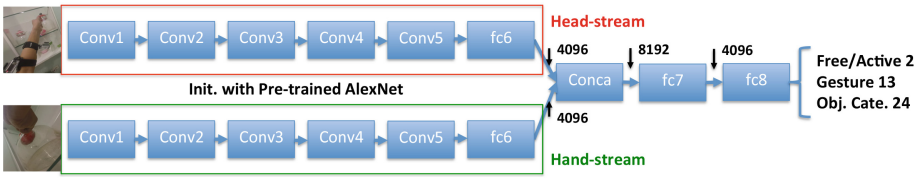

Fig. 7. Architecture of our two-streams CNN. The top and bottom streams take the HeadCam and the HandCam, respectively, as inputs. The two-streams CNN is used to predict Free/Active states, 13 hand gesture states, or 24 object category states. Conv, fc, and Conca denote convolution, fully-connected, and concatenate, respectively.

13 hand gesture states, or 24 object category states. Please see Fig. 7 for the detail architecture. The model weights of both streams are initialized with the ImageNet pre-trained AlexNet model. Then, we finetune the full two-streams CNN with parameters detailed in Sect. 5. After finetuning, we take the last hidden representation of our two-streams CNN as feature to train the state classifier and state change detector as described before. Our experiment shows that jointly finetuning two-streams achieves the best performance in four out of five tasks.

4 Dataset

We have collected a new synchronized "HandCam" and "HeadCam" video data for hand states recognition. Our dataset contains 20 round of data collection (60 video sequences), where each round consists of three synchronized video sequences (two from HandCam and one from HeadCam). In total, our dataset contains ∼115.5 min of videos, which is at a similar scale of the egocentric video dataset [4]. For HandCam, we ask each user to mount the camera so that the palm is in the center of the camera. For HeadCam, we found it is much harder to be mounted correctly by each user. Hence, we help each user to mount the HeadCam so that the user's gaze is in the center of the camera while manipulating objects. Our dataset can be accessed at http://aliensunmin.github.io/project/handcam/.

	# Vid.	# Fra.	# Users	#TO	#Cat.	#Inst.	#Gest.	Categories Instances
Office	6	7213	6	1	6	30	10	lamp switch 1; whiteboard pen 6; thermos bottle 5; book 6; computer 6; magnet 6
Lab	8	9299	8	3	9	58	11	whiteboard eraser 1; computer 7; cellphone 8; coin 4; ruler 8; thermos bottle 7; whiteboard pen 7; pen 8; cup 8
Home	6	11390	4	3	12	35	11	TV remote 1; AC remote 1; switch 1; window 1; fridge 1; cupboard 1; water tap 1; toy 4; kettle 6; cup 6; bottle 6; snack 6
Total	20	27902	11	7	23	111	12	

Fig. 8. Statistics of our HandCam dataset. Vid., Fra., RO, Cars., Inst., and Gest. stand for videos, frames, task-order, categories, instances,and gestures, respectively.

In order to thoroughly analyze tasks involving recognizing object category, hand gesture, etc., we explicitly collect videos in multiple indoor scenes, interacting with multiple object categories, and multiple object instances within each category. A thorough statistics is shown in Fig. 8. We summarize the properties of our dataset below.

- Scene: We have collected videos in three scenes: a small office, a mid-size lab, and a large home (Fig. 6(b)), where office and lab involve many similar object interactions, but involve very different object interactions as in home.
- Task: We pre-define a number of tasks for users to follow. To increase variation, each user randomly selects a task-order to perform.
- Object category: We prepare many instances for most movable objects in Fig. 8. We ensure that these instances are separable in our train/test splits.
- User: We have 11 unique users involved in collecting the videos.

Annotating Interface. For annotating hand states, we ask annotators to watch the synchronized three videos (i.e., two HandCams and one HeadCam) to make label decision. A snapshot of our viewer is shown in technical report [36].

Training vs. Testing Set. We have two settings. Firstly, we train on office and lab. Then, we test on home. We refer this as "Home" setting. This is a challenging setting, since home is an unseen scene and there are many unseen object categories and activities. In the second setting, we evenly divide the video sequences into half for training and the remaining half for testing in all scenes. We refer this as "AllScenes" setting.

5 Implementation Details

Camera System. Our system consists of three synchronized GoPro 3+ cameras to record videos with 1920×1080 resolution at 60 fps and we process them at 6 fps. In the future, we will use small fisheye cameras to to mitigate the issues of unnatural behavior and self-occlusion due to the relatively big GoPro cameras.

Alignment. We achieve stable result by setting $\beta^{th} = 40$ and trying seven scales (i.e., $[0.9, 1, 1.1, 1.2, 1.3, 1.4, 1.5]$) in our multi-scales alignment method.

Training. We set SVM regularization parameters, parameter d of state change features, and λ automatically using 5-fold cross-validation for each setting. We finetune an imagenet pre-trained AlexNet on our dataset using the following parameters consistently for all tasks: maximum iterations = 40000, step-size = 10000, momentum = 0.9, every 10000 iteration weight decay =0.1, and learning rate = 0.001. To augment our dataset, we flip the left HandCam frames horizontally and jointly trained with the right HandCam frames.

Training Two-Streams CNN. For finetuning the two-streams CNN on our dataset, we set maximum iterations = 10000, step-size = 2500, momentum = 0.9, every 2500 iteration weight decay =0.1, and learning rate = 0.001. We also augment our dataset by horizontal flipping frames.

Table 1. Frame-based classification accuracy of hand states: free vs. active, gesture, and object category. Unary denotes the method relying on unary scoring function. Full denotes our full model. See method abbreviation for the naming of each column.

	NoAlign	Align	AlignFT	BL	BLCrop	BCropFT	BCropFTv2	IDT
Free vs. Active								
Home-Unary	70.1 %	71.4 %	74.1 %	60.7 %	61.1 %	61.7 %	57.0 %	59.9 %
Home-Full	71.4 %	74.7 %	75.5 %	62.1 %	63.0 %	64.7 %	57.8 %	60.8 %
AllScene-Unary	76.1 %	75.5 %	79.3 %	65.7 %	68.9 %	69.5 %	64.5 %	70.5 %
AllScene-Full	77.2 %	76.9 %	80.6 %	67.1 %	70.6 %	73.1 %	62.3 %	70.9 %
Gesture								
Home-Unary	53.9 %	53.8 %	54.1 %	48.9 %	47.4 %	44.7 %	52.1 %	54.5 %
Home-Full	55.2 %	55.6 %	56.6 %	51.3 %	48.9 %	50.1 %	55.4 %	55.3 %
AllScene-Unary	60.5 %	60.2 %	63.1 %	55.7 %	57.1 %	53.7 %	53.0 %	59.9 %
AllScene-Full	61.8 %	62.4 %	65.1 %	56.8 %	58.3 %	59.1 %	56.1 %	60.1 %
Object								
AllScene-Unary	60.0 %	59.5 %	62.8 %	53.6 %	55.2 %	51.4 %	51.6 %	56.1 %
AllScene-Full	61.8 %	61.5 %	66.5 %	54.9 %	56.6 %	57.4 %	54.9 %	58.6 %

6 Experiment Results

We evaluate recognition tasks of three hand state: free vs. active, gesture, and object category. Most tasks are conducted in two train/test settings: Home? and AllScenes? as described in Sect. 4. All the following experiments are conducted using fc6 features in Alexnet.

HeadCam Baseline. We apply two state-of-the-art hand segmentation methods [28,29] to predict manipulation region. Similar to [11], we predict at most two boxes, one for left and one for right hands. Typical ground truth and predicted boxes are shown in technical report [36]. Next, we crop the HeadCam images with respect to the predict manipulation region and apply the same methods introduced in this paper to recognize hand states (see technical report [36] for more details). We also use improved dense trajectory [12] of the whole frame to capture the motion cues as a strong but time-consuming baseline.

Method Abbreviation. To facilitate discussion, we introduce the following abbreviations for different methods.

- HeadCam: IDT, BL, BLCrop, BCropFT, and BCropFTv2. IDT is a HeadCam baseline using [12]. BL is a HeadCam baseline using pre-trained feature of the whole frame. BLCrop is a HeadCam baseline using pre-trained feature with regions cropped by [28]. BCropFT is using finetuned feature with regions cropped by [28]. BCropFTv2 is using finetuned feature with regions cropped by [29].
- HandCam: NoAlign, Align, and AlignFT. NoAlign is HandCam without hand alignment using pre-trained feature. Align is HandCam with alignment using pre-trained feature. AlignFT is Align using finetuned feature.

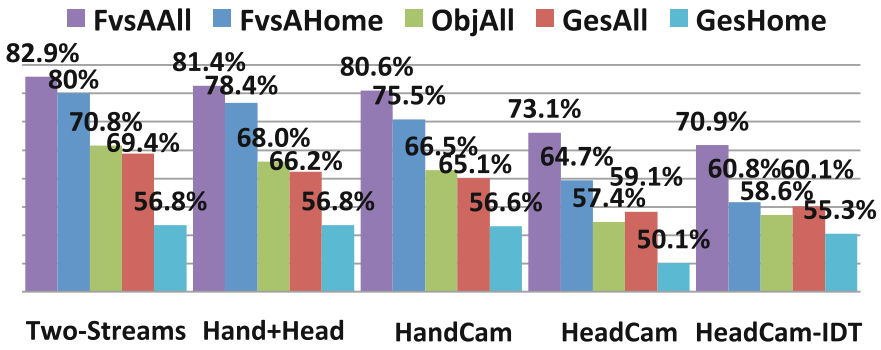

Fig. 9. Comparing two-streams CNN with HandCam+HeadCam, HandCam (AlignFT), HeadCam (BLCropFT), and HeadCam-IDT in five tasks: FvsA stands for free vs. active. Obj stands for object category. Ges stands for gesture.

6.1 Free Vs. Active Recognition

Free vs. active recognition accuracy is shown in the top part of Table 1.

Pre-trained CNN. Using pre-trained CNN feature, our full method already outperforms the non-cropped, cropped, and IDT HeadCam baselines. These results confirm that HandCam is a great alternative to HeadCam systems.

Unary vs. Full. Our full model also consistently outperforms the unary model in all settings and for both HandCam and HeadCam.

Hand Alignment. Although hand alignment shows no critical improvement in AllScenes, we confirm that hand alignment improves from 71.4 % to 74.7 % acc. in the challenging Home setting.

Finetune CNN. Finetuning CNN shows consistent improvement in all settings and for both HandCam and HeadCam. Our finetuned full method achieves 75.5 % acc. in Home and 80.6 % acc. in AllScenes.

6.2 Gesture Recognition

Gesture recognition (see the middle part of Table 1) shares the same trend in free vs. active recognition. Except that, in Home setting, hand alignment only shows 0.4 % acc. improvement using pre-trained feature. Nevertheless, our finetuned full method achieves 56.6 % acc. in Home and 65.1 % acc. in AllScenes. They are 6.6 % (Home) and 6 % (AllScenes) better than the finetuned cropped baseline, and 1.3 % (Home) and 5 % (AllScenes) better than IDT.

6.3 Object Category Recognition

Object category recognition accuracy in AllScenes setting is shown in the last part of Table 1. We found that it shares the same trend in Sect. 6.1. Most importantly, our finetuned full method achieves the best accuracy (66.5 %). In Home

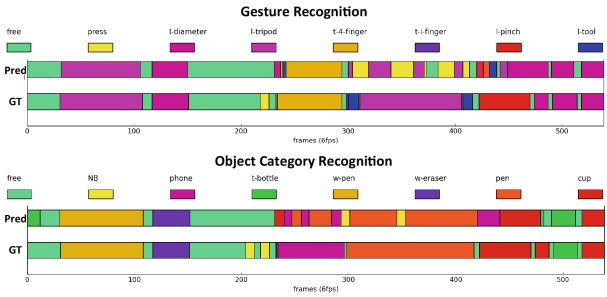

Fig. 10. Temporal visualization of predicted hand gesture (top-row) and object category (bottom-row) using two-streams CNN in AllScenes. Pred and GT stands for prediction and ground truth. The color-code of states are on top of each visualization.

setting, since many object categories are not observed in training, we evaluate object discovery task. We treat object category discovery as a clustering task, and compare the clustering results between our best HandCam configuration (i.e., AlignFT) and the best HeadCam configuration (i.e., BLCropFT). We report a modified purity (defined in technical report [36]) to focus on discovering object categories (not free-hand?). We calculate the purity with different number of clusters (see technical report [36]), and find that HandCam outperforms HeadCam by about 10 % from 30 to 100 clusters.

6.4 Combining HandCam with HeadCam

We show comparison among HeadCam (BLCropFT), HeadCam motion (HeadCam-IDT), HandCam (AlignFT), HandCam+HeadCam (feature concatenation), and our novel two-streams CNN in Fig. 9. HandCam+HeadCam with simple feature concatenation already outperforms the single camera settings in all five tasks. Most importantly, our novel two-streams CNN achieves the best performance in four our of five tasks (except gesture in Home setting). We show temporal visualization of predicted vs. ground truth hand gesture and object category of our two-streams CNN in Fig. 10 (see more in technical report [36]).

7 Conclusion

We revisit a wrist-mounted camera system (HandCam) for recognizing various hand states. To evaluate our system, we collect a new dataset with synchronized HandCam and HeadCam observing multiple object categories, instances, gestures in multiple scenes. HandCam with deep-learning-based method consistently outperforms HeadCam systems in all tasks by at most 10.8 % improvement in accuracy. Most importantly, we show that combining HandCam with HeadCam using a novel two-streams CNN gives the best performance in four out of five tasks. With more data and a more sophisticated network, we believe the recognition performance of our system can be greatly improved in the future.

Acknowledgements. We thank MOST 104-2221-E-007-089-MY2 in Taiwan for their support. We also thank Hou Ning Hu for collaboration.

References

1. Pirsiavash, H., Ramanan, D.: Detecting activities of daily living in first-person camera views. In: CVPR (2012)
2. Rogez, G., Supani, J.S., Khademi, M., Montiel, J.M.M., Ramanan, D.: 3d hand pose detection in egocentric RGB-D images. CoRR abs/1412.0065 (2014)
3. Rogez, G., Supani, J.S., Ramanan, D.: First-person pose recognition using egocentric workspaces. In: CVPR (2015)
4. Damen, D., Leelasawassuk, T., Haines, O., Calway, A., Mayol-Cuevas, W.: You-do, i-learn: discovering task relevant objects and their modes of interaction from multi-user egocentric video. In: BMVC (2014)
5. Kim, D., Hilliges, O., Izadi, S., Butler, A.D., Chen, J., Oikonomidis, I., Olivier, P.: Digits: freehand 3d interactions anywhere using a wrist-worn gloveless sensor. In: UIST (2012)
6. Saxena, A., Driemeyer, J., Ng, A.: Robotic grasping of novel objects using vision. Int. J. Rob. Res. **27**(2), 157–173 (2008)
7. Mayol-Cuevas, W., Tordoff, B., Murray, D.: On the choice and placement of wearable vision sensors. IEEE Trans. Syst. Man Cybern. Part A: Syst. Hum. **39**(2), 414–425 (2009)
8. Maekawa, T., Yanagisawa, Y., Kishino, Y., Ishiguro, K., Kamei, K., Sakurai, Y., Okadome, T.: Object-based activity recognition with heterogeneous sensors on wrist. In: Floréen, P., Krüger, A., Spasojevic, M. (eds.) Pervasive 2010. LNCS, vol. 6030, pp. 246–264. Springer, Heidelberg (2010). doi:10.1007/978-3-642-12654-3_15
9. Fathi, A., Ren, X., Rehg, J.M.: Learning to recognize objects in egocentric activities. In: CVPR (2011)
10. Maekawa, T., Kishino, Y., Yanagisawa, Y., Sakurai, Y.: Wristsense: wrist-worn sensor device with camera for daily activity recognition. In: PERCOM Workshops. IEEE (2012)
11. Li, Y., Fathi, A., Rehg, J.M.: Learning to predict gaze in egocentric video. In: ICCV(2013)
12. Wang, H., Schmid, C.: Action recognition with improved trajectories. In: ICCV (2013)
13. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: NIPS (2012)
14. Patterson, D.J., Fox, D., Kautz, H., Philipose, M.: Fine-grained activity recognition by aggregating abstract object usage. In: ISWC (2005)
15. Stikic, M., Huynh, T., Laerhoven, K.V., Schiele, B.: ADL recognition based on the combination of RFID, and accelerometer sensing. In: Pervasive Computing Technologies for Healthcare (2008)
16. Wu, J., Osuntogun, A., Choudhury, T., Philipose, M., Rehg, J.M.: A scalable approach to activity recognition based on object use. In: ICCV (2007)
17. Fathi, A., Farhadi, A., Rehg, J.M.: Understanding egocentric activities. In: ICCV (2011)
18. Fathi, A., Rehg, J.M.: Modeling actions through state changes. In: CVPR (2013)

19. Fathi, A., Li, Y., Rehg, J.M.: Learning to recognize daily actions using gaze. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012. LNCS, vol. 7572, pp. 314–327. Springer, Heidelberg (2012). doi:10.1007/978-3-642-33718-5_23
20. Li, Y., Ye, Z., Rehg, J.M.: Delving into egocentric actions. In: CVPR (2015)
21. Ghosh, J., Lee, Y.J., Grauman, K.: Discovering important people and objects for egocentric video summarization. In: CVPR (2012)
22. Lu, Z., Grauman, K.: Story-driven summarization for egocentric video. In: CVPR (2013)
23. Sun, M., Farhadi, A., Taskar, B., Seitz, S.: Salient montages from unconstrained videos. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8695, pp. 472–488. Springer, Heidelberg (2014). doi:10.1007/978-3-319-10584-0_31
24. De la Torre, F., Hodgins, J.K., Montano, J., Valcarcel, S.: Detailed human data acquisition of kitchen activities: the CMU-multimodal activity database (CMU-MMAC). In: Workshop on Developing Shared Home Behavior Datasets to Advance HCI and Ubiquitous Computing Research, in conjuction with CHI 2009 (2009)
25. Moghimi, M., Azagra, P., Montesano, L., Murillo, A.C., Belongie, S.: Experiments on an rgb-d wearable vision system for egocentric activity recognition. In: CVPR Workshop on Egocentric (First-person) Vision (2014)
26. Damen, D., Gee, A., Mayol-Cuevas, W., Calway, A.: Egocentric real-time workspace monitoring using an rgb-d camera. In: IROS (2012)
27. Li, C., Kitani, K.M.: Pixel-level hand detection in egocentric videos. In: CVPR (2013)
28. Li, C., Kitani, K.M.: Model recommendation with virtual probes for egocentric hand detection. In: ICCV (2013)
29. Bambach, S., Lee, S., Crandall, D.J., Yu, C.: Lending a hand: Detecting hands and recognizing activities in complex egocentric interactions. In: ICCV (2015)
30. Betancourt, A., Lopez, M., Regazzoni, C., Rauterberg, M.: A sequential classifier for hand detection in the framework of egocentric vision. In: CVPRW (2014)
31. Huang, D.A., Ma, M., Ma, W.C., Kitani., K.M.: How do we use our hands? discovering a diverse set of common grasps. In: CVPR (2015)
32. Vardy, A., Robinson, J., Cheng, L.T.: The wristcam as input device. In: ISWC (1999)
33. Chan, L., Chen, Y.L., Hsieh, C.H., Liang, R.H., Chen, B.Y.: Cyclopsring: enabling whole-hand and context-aware interactions through a fisheye ring. In: UIST (2015)
34. Ohnishi, K., Kanehira, A., Kanezaki, A., Harada, T.: Recognizing activities of daily living with a wrist-mounted camera. In: CVPR (2016)
35. Wu, R., Yan, S., Shan, Y., Dang, Q., Sun, G.: Deep image: scaling up image recognition. CoRR abs/1501.02876 (2015)
36. Chan, C.S., Chen, S.Z., Xie, P.X., Chang, C.C., Sun, M.: Technical report of recognition from hand cameras. http://aliensunmin.github.io/project/handcam/
37. Canny, J.: A computational approach to edge detection. PAMI **PAMI–8**(6), 679–698 (1986)
38. Wu, C.: Towards linear-time incremental structure from motion. In: 3DV (2013)
39. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: ICLR (2015)
40. Simonyan, K., Zisserman, A.: Two-stream convolutional networks for action recognition in videos. In: NIPS (2014)