Recognizing Activities of Daily Living with a Wrist-mounted Camera
Supplemental Material

Katsunori Ohnishi, Atsushi Kanehira, Asako Kanezaki, Tatsuya Harada
Graduate School of Information Science and Technology, The University of Tokyo
{ohnishi, kanehira, kanezaki, harada}@mi.t.u-tokyo.ac.jp

A. Calculating PLS to compute eigenvectors that construct a weight matrix $W_{\rm sp}$

We compute eigenvectors that construct $W_{\rm sp}$ by calculating partial least squares (PLS). The idea to use PLS for obtaining weights for discriminative features was derived from [1]. Here, we reproduce Eq. (3), (4), and (5) in our paper below:

$$V = \sum_{i=1}^{a}\sum_{j=1}^{a} \boldsymbol{v}(i,j)\boldsymbol{w}_{(i,j)}^{\top} = V_{\rm sp}W_{\rm sp}, \tag{1}$$

$$V_{\rm sp} = (\boldsymbol{v}(1,1), \boldsymbol{v}(1,2), \ldots, \boldsymbol{v}(a,a)), \tag{2}$$

$$W_{\rm sp} = (\boldsymbol{w}_{(1,1)}, \boldsymbol{w}_{(1,2)}, \ldots, \boldsymbol{w}_{(a,a)})^{\top}. \tag{3}$$

We let $\boldsymbol{w} \in \mathbb{R}^{a^2}$ be a column vector in $W_{\rm sp}$ and $\boldsymbol{x} \in \mathbb{R}^{MK}$ denote the corresponding column vector in V (*i.e.* $\boldsymbol{x} = V_{\rm sp}\boldsymbol{w}$). Suppose that we have N labeled training samples $\{\boldsymbol{x}_i, y_i\}_{i=1}^{N}$ with C classes, where $\boldsymbol{x}_i = V_{\rm sp}^{i}\boldsymbol{w}$ and y_i represents the class label of the i-th training sample ranging from 1 to C. The between-class covariance matrix S_b can be written as follows:

$$S_b = \frac{1}{N}\sum_{c=1}^{C} n_c(\bar{\boldsymbol{x}}_c - \bar{\boldsymbol{x}})(\bar{\boldsymbol{x}}_c - \bar{\boldsymbol{x}})^{\top}, \tag{4}$$

where $\bar{\boldsymbol{x}}_c = \frac{1}{n_c}\sum_{i\in\{i|y_i=c\}} \boldsymbol{x}_i$, $\bar{\boldsymbol{x}} = \frac{1}{N}\sum_i \boldsymbol{x}_i$, and n_c is the number of samples in the c-th class. The trace of S_b is given by:

$$\mathrm{tr}S_b = \boldsymbol{w}^{\top}\Sigma_b\boldsymbol{w}, \tag{5}$$

where

$$\Sigma_b = \frac{1}{N}\sum_{c=1}^{C} n_c(M_c - M)^{\top}(M_c - M). \tag{6}$$

Here, $M_c = \frac{1}{n_c}\sum_{i\in\{i|y_i=c\}} \boldsymbol{x}_i$ is the mean of $\boldsymbol{x}_i$ belonging to the c-th class, and $M = \frac{1}{N}\sum_i \boldsymbol{x}_i$ is the mean of all samples in the training dataset. By maximizing Eq. (5) under the condition $\boldsymbol{w}^{\top}\boldsymbol{w} = 1$, we obtain the eigenvector of the following eigenvalue problem:

$$\Sigma_b\boldsymbol{w} = \lambda\boldsymbol{w}, \tag{7}$$

where λ is the eigenvalue corresponding to the eigenvector $\boldsymbol{w}$. We select the $N_{\rm sp}$ largest eigenvalues $\lambda_1, \ldots, \lambda_{N_{\rm sp}}$, and the corresponding eigenvectors $\boldsymbol{w}_1, \ldots, \boldsymbol{w}_{N_{\rm sp}}$. Finally, we create $W_{\rm sp}$ by arranging $\boldsymbol{w}_1, \ldots, \boldsymbol{w}_{N_{\rm sp}}$ in a row. As described in our paper, $W_{\rm tmp}$ can be obtained in the same manner as $W_{\rm sp}$ when $W_{\rm sp}$ is fixed.

B. Action definition of our dataset

Table 1 shows the definition of each class in our dataset.

Activity Class	Definition
1. Vacuuming	The initial action is taking a vacuum cleaner and the final action is putting it back after vacuuming.
2. Empty trash	The initial action is taking a garbage bag and the final action is tying it.
3. Wipe desk	The initial action is taking a kitchen cloth, the next action is wiping, and the final action is putting it after washing.
4. Turn on air-conditioner	The initial action is taking a remote controller, the next action is turing on, and the final action is making sure an air-conditioner turns on.
5. Open and close door	The initial action is taking door knob and the final action is releasing user's hand.
6. Make coffee	The initial action is taking a drip filter and the final action is throwing it away after pouring.
7. Make tea	The initial action is taking tea bag and the final action is throwing it away after pouring.
8. Wash dishes	The initial action is turning a tap on and the final action is turning it off after washing.
9. Dry dishes	The initial action is taking a cloth and the final action is putting it after drying.
10. Use microwave	The initial action is putting a thing into microwave and the final action is opening microwave and taking the thing out.
11. Use refrigerator	The initial action is opening a refrigerator door, the next action is taking a thing, and the final action is closing the door.
12. Wash hands	The initial action is turning a tap on and the final action is turning it off.
13. Dry hands	The initial action is taking a cloth and the final action is putting the cloth after drying.
14. Drink water from a bottle	The initial action is taking a plastic bottle and the final action is putting it after drinking.
15. Drink water from a cup	The initial action is taking a cup and the final action is putting it after drinking.
16. Read book	The initial action is taking a book and the final action is putting it after reading.
17. Write on paper	The initial action is taking a pen and the final action is putting it after writing.
18. Open and close drawer	The initial action is opening a drawer, the next action is taking a thing, and the final action is closing the drawer.
19. Cut paper	The initial action is taking scissors and the final action is putting them after cutting.
20. Staple paper	The initial action is taking stapler and the final action is putting it after stapling.
21. Fold origami	The initial action is taking origami and the final action is folding it.
22. Use smartphone	The initial action is taking a smartphone and the final action is putting it after using.
23. Watch TV	The initial action is taking a remote controller and the final action is turning TV off after watching.

Table 1. Activity definitions of our datasets

C. Confusion matrix

Figures 1, 2, 3, and 4 are the confusion matrices. We can see how action classes are confused in each figure. For example, "make coffee" and "make tea" are confused in every case. These actions have common handled objects such as a mug and pot. The biggest difference is whether the user uses tea bag or coffee beans and filter. It is difficult for head-mounted camera to recognize these objects. However, wrist-mounted camera can recognize small handed objects easily. Thus, LCD [5] on wrist-mounted camera dataset (WCD) recognizes "make coffee" and "make tea" better than LCD on head-mounted camera dataset (HCD).

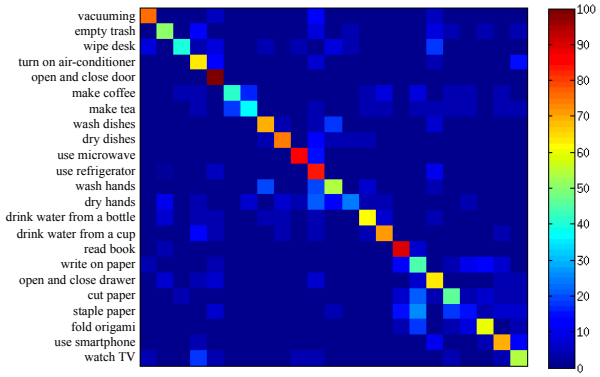

Figure 1. The confusion matrix for LCD on HCD

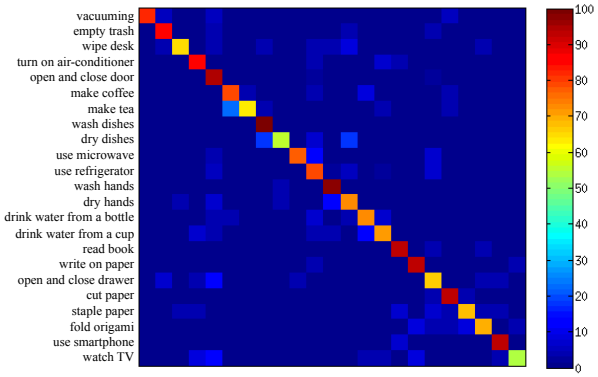

Figure 2. The confusion matrix for LCD on WCD

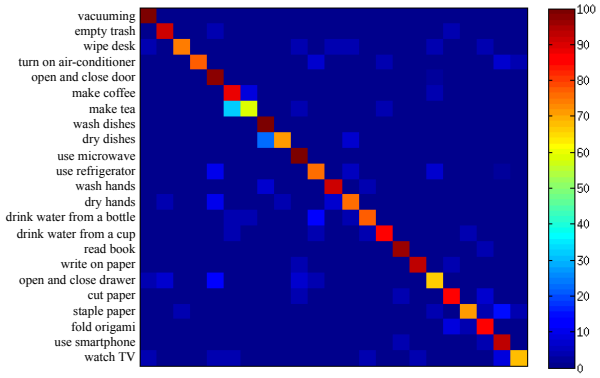

Figure 3. The confusion matrix for DSTAR on WCD

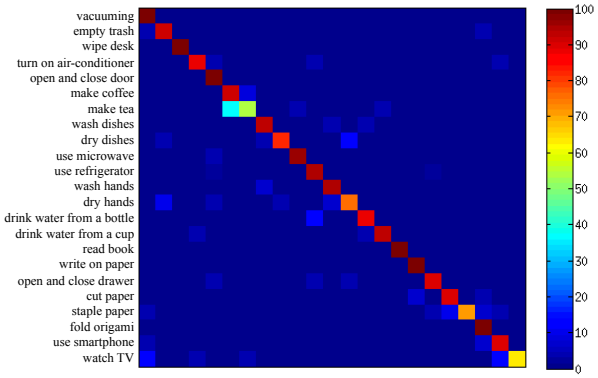

Figure 4. The confusion matrix for DSTAR on WCD & iDT on HCD

D. Analysis of the impact of parameters

In this section, we find the best parameters for each method.

D.1. Parameters for LCD

We first find the best parameters for LCD; the number of centers K in VLAD and descriptor dimension. Table 2 shows that features get more discriminative with the increase of K on WCD. However when $K = 1024$, the features get too sparse and less discriminative. Though we find the best parameter $(K, D) = (128, 256)$, the compressed dimension by PCA dose not seem to have much effect.

Table 3 also shows that features get more discriminative with the increase of K on HCD. However when $K = 512$, the features get too sparse and less discriminative. Unlike WCD, when dimensions of each descriptor are compressed from 512-D to 64-D, they lose the discriminative ability. This is understood as follows: the images captured by wrist-mounted camera have less variety than the images by head-mounted camera. Thus, the descriptors extracted from WCD can be more compact than those from HCD.

Dimensions \ Clusters	$K = 64$	$K = 128$	$K = 256$	$K = 512$	$K = 1024$
64-D	74.0	74.1	76.0	78.2	77.1
128-D	73.4	74.3	75.8	**78.6**	77.0
256-D	74.5	76.8	77.1	78.5	77.2

Table 2. Impact on dimensions and numbers of centers K for LCD on WCD

Dimensions \ Clusters	$K = 64$	$K = 128$	$K = 256$	$K = 512$	$K = 1024$
64-D	48.5	52.5	57.6	56.4	57.0
128-D	54.9	56.0	**62.4**	62.1	61.2
256-D	56.1	60.8	60.1	61.0	61.5

Table 3. Impact on dimensions and numbers of centers K for LCD on HCD

We also find the best parameter for LCD_{SPP}. Tables 4 and 5 show the obtained results of LCD_{SPP}. We can see similar trend as LCD without Spatial Pooling Pyramid (SPP) layer shown in Tables 2 and 3. The best parameter (K, D) are $(128, 256)$ for LCD_{SPP} on WCD and $(256, 256)$ for LCD_{SPP} on HCD. We employ the score obtained with these parameters in submitted paper.

Dimensions \ Clusters	$K = 64$	$K = 128$	$K = 256$	$K = 512$
64-D	65.9	68.6	70.9	72.0
128-D	70.0	70.2	71.9	73.2
256-D	73.3	73.1	**73.4**	72.9

Table 4. Impact on dimensions and numbers of centers K for LCD_{SPP} on WCD

Dimensions \ Clusters	$K = 64$	$K = 128$	$K = 256$	$K = 512$
64-D	45.9	46.3	45.9	45.9
128-D	46.6	47.0	48.7	46.1
256-D	48.5	46.9	**51.3**	48.1

Table 5. Impact on dimensions and numbers of centers K for LCD_{SPP} on HCD

D.2. Number of spatial elements

Next, we find the best parameters for DSAR; the number of centers K in VLAD, descriptor dimension, and N_{sp}. Tables 6 and 7 show the best parameter for DSAR on WCD and DSAR on HCD. We can see that N_{sp} dose not need to be a large number though it can be set up-to 49 in VGG-net [3] case. The similar trend can be seen in the D-SPR [1]. If features are cast into well-isolated space by PLS, using too large N_{sp} means adding inefficient features.

For numbers of clusters, $K = 512$ seems too sparse unlike Table 2. We calculate weights W_{sp}, shown in Eq. (3), from separately aggregated features in each cell. These separately aggregated features can be more sparse than LCD features. Thus, the best number of clusters for DSAR is smaller than that of LCD.

We can find the best parameter $(K, D, N_{\mathrm{sp}}) = (64, 256, 5)$ for DSAR on WCD and $(K, D, N_{\mathrm{sp}}) = (256, 128, 5)$ for DSAR on HCD from Tables 6 and 7.

Clusters	$K = 64$			$K = 128$			$K = 256$			$K = 512$		
N_{sp} / Dimensions	5	10	20	5	10	20	5	10	20	5	10	20
64-D	77.9	78.8	76.9	78.8	78.2	77.4	80.2	78.7	75.8	77.4	78.6	74.2
128-D	80.2	79.3	77.9	80.4	81.4	78.9	81.1	81.1	77.1	80.9	79.9	76.1
256-D	**82.0**	81.0	80.0	81.6	81.0	79.3	79.7	80.5	78.1	80.6	78.6	76.7

Table 6. Impact on dimensions, numbers of centers K, and N_{sp} for DSAR on WCD.

Clusters	$K = 64$			$K = 128$			$K = 256$			$K = 512$		
N_{sp} / Dimensions	5	10	20	5	10	20	5	10	20	5	10	20
64-D	58.6	59.3	57.2	57.3	55.6	52.4	57.1	56.9	53.4	58.8	56.9	54.0
128-D	60.1	57.3	56.0	60.2	59.3	56.4	**61.6**	58.4	57.5	60.9	58.4	56.7
256-D	59.7	59.1	57.8	59.9	59.3	56.9	61.5	59.2	56.3	60.8	59.2	57.4

Table 7. Impact on dimensions, numbers of centers K, and N_{sp} for DSAR on HCD.

D.3. Number of spatial and temporal elements

We finally find the best parameter for DSTAR; the number of centers K in VLAD, descriptor dimension, and N_{tmp}. Following the result described in Section D.2, we fix $N_{\mathrm{sp}} = 5$. Tables 8 and 9 show the best parameter for DSTAR on WCD and DSTAR on HCD. We can find the best parameter $(K, D, N_{\mathrm{tmp}}) = (128, 64, 5)$ for DSAR on WCD and $(K, D, N_{\mathrm{sp}}) = (128, 128, 5)$ for DSAR on HCD from Tables 6 and 7.

Clusters	$K = 64$		$K = 128$	
N_{tmp} / Dimensions	3	5	3	5
64-D	81.3	81.3	82.8	**83.7**
128-D	82.8	83.2	82.6	83.5

Table 8. Impact on dimensions, numbers of centers K, and N_{tmp} for DSTAR on WCD, with fixed $N_{\mathrm{sp}} = 5$.

Clusters	$K = 64$		$K = 128$	
N_{tmp} / Dimensions	3	5	3	5
64-D	60.4	60.0	58.7	59.4
128-D	60.2	59.6	60.7	**62.0**

Table 9. Impact on dimensions, numbers of centers K, and of N_{tmp} for DSTAR on HCD, with fixed $N_{\mathrm{sp}} = 5$.

E. Parameters on existing datasets

E.1. UCIADL

We find the best parameters for LCD, DSAR, and DSTAR on UCIADL [2]. Following Section D, we fixed $N_{\text{sp}} = 5$ and $N_{\text{tmp}} = 5$.

Dimensions \ Clusters	$K = 64$	$K = 128$	$K = 256$
64-D	65.7	67.8	68.9
128-D	70.8	71.0	68.7
256-D	71.1	**73.7**	70.8

Table 10. Impact on dimensions and numbers of centers K for LCD on UCIADL.

Dimensions \ Clusters	$K = 64$	$K = 128$	$K = 256$
64-D	68.8	69.6	70.6
128-D	70.8	70.6	70.4
256-D	70.9	**71.8**	69.4

Table 11. Impact on dimensions and numbers of centers K for DSAR on UCIADL, with fixed $N_{\text{sp}} = 5$.

Dimensions \ Clusters	$K = 64$	$K = 128$	$K = 256$
64-D	71.6	71.8	**72.6**
128-D	71.6	72.0	70.5
256-D	70.0	72.0	71.1

Table 12. Impact on dimensions and numbers of centers K for DSTAR on UCIADL, with fixed $N_{\text{sp}} = 5$ and $N_{\text{tmp}} = 5$.

E.2. UCF101

We also find the best parameters for LCD, DSAR, and DSTAR on one of the representative action recognition datasets, UCF101 [4]. Following Section D, we fixed $N_{\text{sp}} = 5$ and $N_{\text{tmp}} = 5$. Note that we did not calculate when the number of clusters is 256 and the number of dimensions is 256 due to memory shortage.

Dimensions \ Clusters	$K = 64$	$K = 128$	$K = 256$
64-D	70.2	72.5	73.8
128-D	72.9	74.5	75.8
256-D	75.2	76.5	**76.8**

Table 13. Impact on dimensions and numbers of centers K for LCD on UCF101.

Dimensions \ Clusters	$K = 64$	$K = 128$	$K = 256$
64-D	75.9	76.5	76.4
128-D	77.8	77.8	77.2
256-D	**78.7**	78.5	77.4

Table 14. Impact on dimensions and numbers of centers K for DSAR on UCF101, with fixed $N_{\text{sp}} = 5$.

Dimensions \ Clusters	$K = 64$	$K = 128$	$K = 256$
64-D	77.0	77.4	76.9
128-D	78.5	77.9	77.1
256-D	**79.3**	78.3	-

Table 15. Impact on dimensions and numbers of centers K for DSTAR on UCF101, with fixed $N_{\text{sp}} = 5$ and $N_{\text{tmp}} = 5$.

F. dataset example

In this section, we show example images of all action classes in our dataset.

Figure 5. example of head-mounted camera dataset with a label, “vacuuming”

Figure 6. example of wrist-mounted camera dataset with a label, “vacuuming”

Figure 7. example of head-mounted camera dataset with a label, “empty trash”

Figure 8. example of wrist-mounted camera dataset with a label, “empty trash”

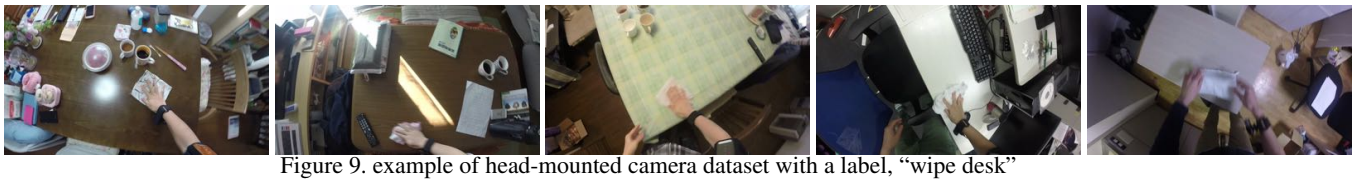

Figure 9. example of head-mounted camera dataset with a label, “wipe desk”

Figure 10. example of wrist-mounted camera dataset with a label, “wipe desk”

Figure 11. example of head-mounted camera dataset with a label, “turn on air-conditioner”

Figure 12. example of wrist-mounted camera dataset with a label, “turn on air-conditioner”

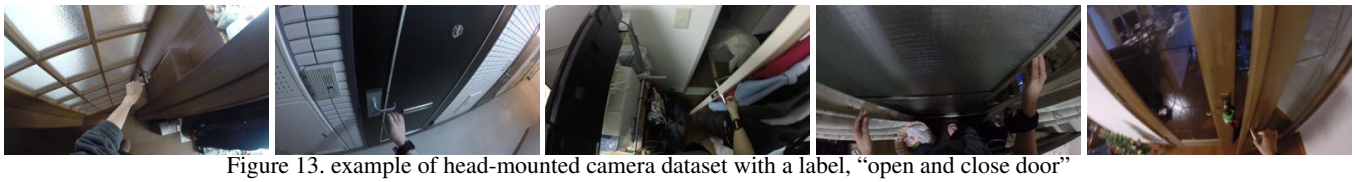

Figure 13. example of head-mounted camera dataset with a label, “open and close door”

Figure 14. example of wrist-mounted camera dataset with a label, “open and close door”

Figure 15. example of head-mounted camera dataset with a label, “make coffee”

Figure 16. example of wrist-mounted camera dataset with a label, “make coffee”

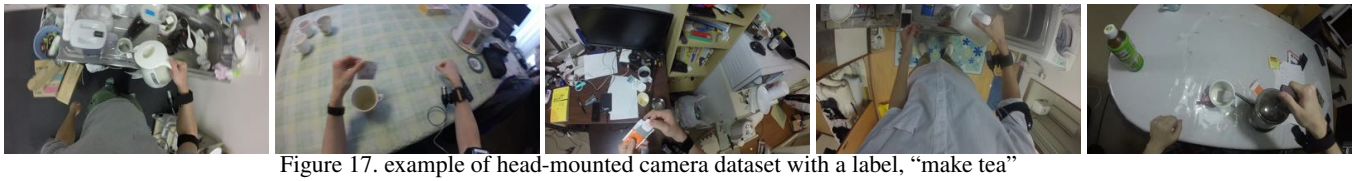

Figure 17. example of head-mounted camera dataset with a label, “make tea”

Figure 18. example of wrist-mounted camera dataset with a label, “make tea”

Figure 19. example of head-mounted camera dataset with a label, “wash dishes”

Figure 20. example of wrist-mounted camera dataset with a label, “wash dishes”

Figure 21. example of head-mounted camera dataset with a label, “dry dishes”

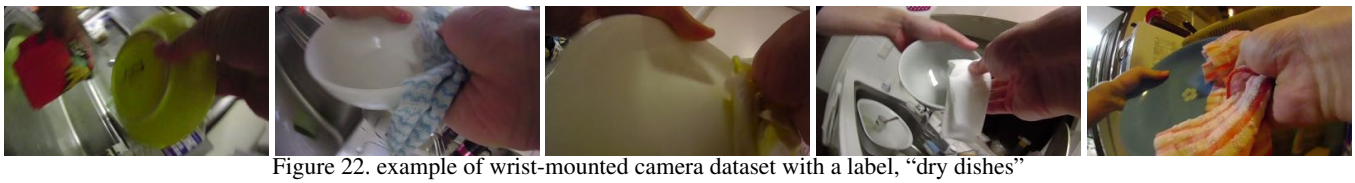

Figure 22. example of wrist-mounted camera dataset with a label, “dry dishes”

Figure 23. example of head-mounted camera dataset with a label, “use microwave”

Figure 24. example of wrist-mounted camera dataset with a label, “use microwave”

Figure 25. example of head-mounted camera dataset with a label, “use refrigerator”

Figure 26. example of wrist-mounted camera dataset with a label, “use refrigerator”

Figure 27. example of head-mounted camera dataset with a label, “wash hands”

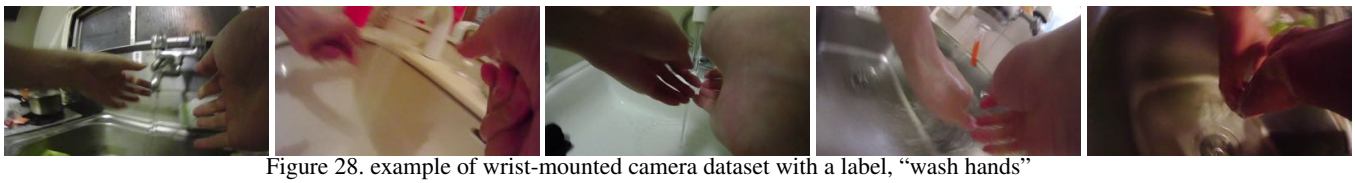

Figure 28. example of wrist-mounted camera dataset with a label, “wash hands”

Figure 29. example of head-mounted camera dataset with a label, “dry hands”

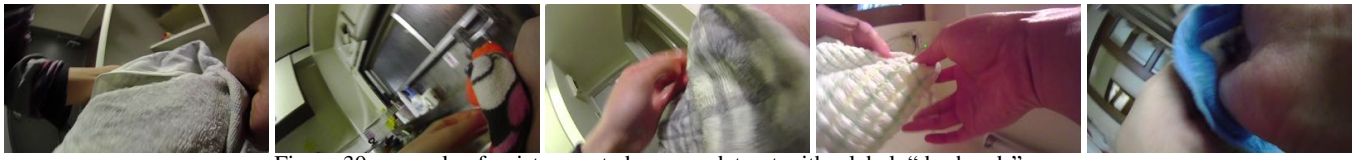

Figure 30. example of wrist-mounted camera dataset with a label, “dry hands”

Figure 31. example of head-mounted camera dataset with a label, “drink water from a bottle”

Figure 32. example of wrist-mounted camera dataset with a label, “drink water from a bottle”

Figure 33. example of head-mounted camera dataset with a label, “drink water from a cup”

Figure 34. example of wrist-mounted camera dataset with a label, “drink water from a cup”

Figure 35. example of head-mounted camera dataset with a label, “read book”

Figure 36. example of wrist-mounted camera dataset with a label, “read book”

Figure 37. example of head-mounted camera dataset with a label, “write on paper”

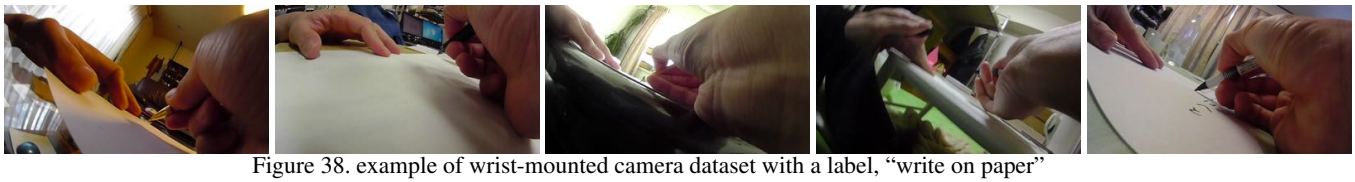

Figure 38. example of wrist-mounted camera dataset with a label, “write on paper”

Figure 39. example of head-mounted camera dataset with a label, “open and close drawer’

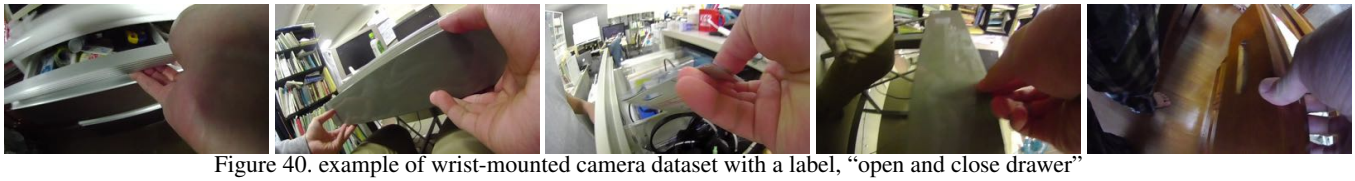

Figure 40. example of wrist-mounted camera dataset with a label, “open and close drawer”

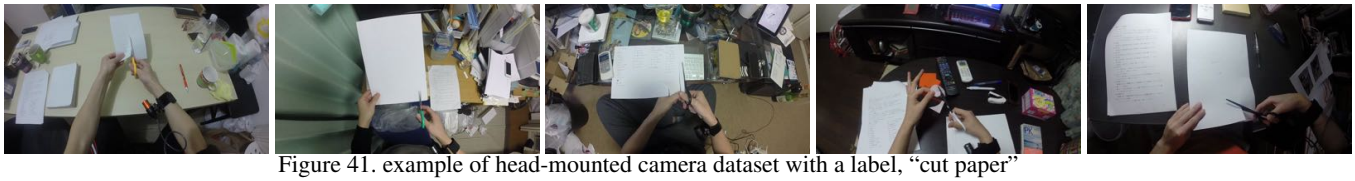

Figure 41. example of head-mounted camera dataset with a label, “cut paper”

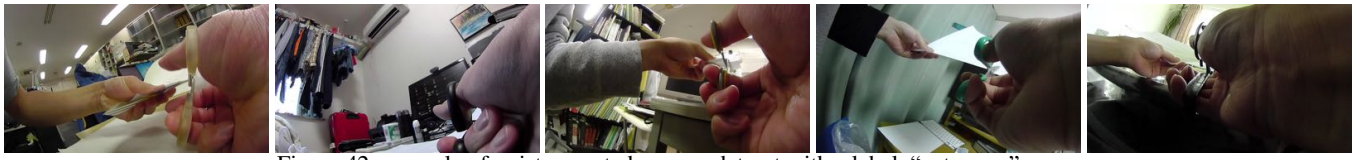

Figure 42. example of wrist-mounted camera dataset with a label, “cut paper”

Figure 43. example of head-mounted camera dataset with a label, “staple paper”

Figure 44. example of wrist-mounted camera dataset with a label, “staple paper”

Figure 45. example of head-mounted camera dataset with a label, “fold origami”

Figure 46. example of wrist-mounted camera dataset with a label, “fold origami”

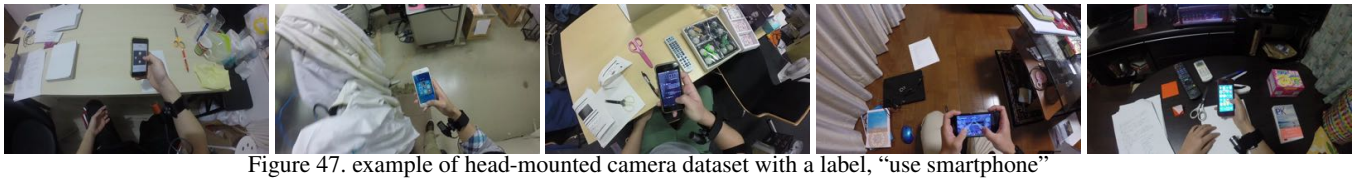

Figure 47. example of head-mounted camera dataset with a label, “use smartphone”

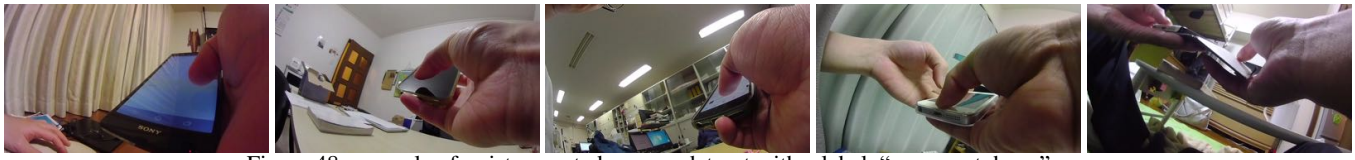

Figure 48. example of wrist-mounted camera dataset with a label, “use smartphone”

Figure 49. example of head-mounted camera dataset with a label, “watch TV”

Figure 50. example of wrist-mounted camera dataset with a label, “watch TV”

References

[1] T. Harada, Y. Ushiku, Y. Yamashita, and Y. Kuniyoshi. Discriminative spatial pyramid. In *CVPR*, 2011. 1, 5

[2] H. Pirsiavash and D. Ramanan. Detecting activities of daily living in first-person camera views. In *CVPR*, 2012. 6

[3] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv:1409.1556*, 2014. 5

[4] K. Soomro, A. R. Zamir, and M. Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv:1212.0402*, 2012. 7

[5] Z. Xu, Y. Yang, and A. G. Hauptmann. A discriminative CNN video representation for event detection. In *CVPR*, 2015. 3