Object-based Activity Recognition Using Egocentric Video Based on Web Knowledge

Tomoya Nakatani
Graduate School of
Information Science and Technology,
Osaka University
nakatani.tomoya@ist.osaka-u.ac.jp

Ryohei Kuga
Graduate School of
Information Science and Technology,
Osaka University

Takuya Maekawa
Graduate School of
Information Science and Technology,
Osaka University
maekawa@ist.osaka-u.ac.jp

***Abstract*—This study attempts to recognize daily activities based on a wearable camera without using training data prepared by a user in her environment. Recently, deep learning frameworks have been publicly available, and we can now easily use deep convolutional neural networks (CNNs) pre-trained on a large image data set. In our method, we first detect objects used in the user's activity from her first-person images using a pre-trained CNN for object recognition. We then estimate an activity of the user using the object detection result because objects used in an activity strongly relate to the activity. To estimate the activity without using training data, we utilize knowledge on the Web because the Web is a repository of knowledge that reflects real-world events and common sense. Specifically, we compute semantic similarity between a list of the detected object names and a name of each activity class based on the Web knowledge. The activity class with the largest similarity value is the estimated activity of the user.**

***Index Terms*—Activity recognition, egocentric video, object**

I. Introduction

Due to the recent proliferation of cheap and small sensors, many researchers have employed wearable and ubiquitous sensors to recognize human daily activities. Two main approaches are used for activity recognition studies: environment augmentation and wearable sensing. The environment augmentation approach employs ubiquitous sensors embedded in our daily life environments such as accelerometers, RFID tags, switch sensors, and vibration sensors attached to daily life objects [1]–[5] Although the environment augmentation approach can achieve fine-grained measurements of daily lives, its deployment and maintenance costs.

The wearable sensing approach attempts to recognize a user's activities by employing body-worn sensors such as accelerometers to capture characteristic body movements and postures adopted for certain activities [6]–[9]. An advantage of this approach is that it does not require environment embedded sensors, which require huge install and maintenance costs and detract from the aesthetics of artifacts in the home. However, the studies based on the accelerometers can recognize only simpler activities than the environment augmentation approach, which senses object use. In addition, the environment augmentation and wearable sensing studies that employ machine learning techniques require the user to prepare labeled training data herself in her daily life environment in many cases.

Meanwhile, due to the recent proliferation of wearable cameras, activity recognition using egocentric videos captured by wearable cameras has been attracting attention. This study also focuses on activity recognition using egocentric videos. Recent state-of-the-art studies rely on deep convolutional neural networks (CNNs) to recognize daily activities using egocentric videos. However, these approaches also require the user to prepare labeled training data herself in her daily life environment.

Recently, deep learning frameworks have been publicly available, and we can now easily use CNNs pre-trained on a large image data set such as ImageNet [10]. In this study, we attempt to recognize object-based activities based on a pre-trained CNN without using training data prepared by a user in her environment. Our idea is very simple but effective. We first detect objects used in the user's activity from her first-person images using a pre-trained CNN for object recognition. We then estimate an activity class of the user using the object detection result. To estimate the activity class without using training data, we utilize knowledge on the World Wide Web because it is assumed that the Web is an easily accessible repository of knowledge that reflects real-world events and common sense. Specifically, we compute semantic similarity between a list of the detected object names and a name of each activity class based on the Web knowledge. The activity class with the largest similarity value is the estimated activity of the user.

In this study, we employ 1) a Web search engine and 2) a lexical database on the Web as the Web knowledge, and compare them in the evaluation section. To measure semantic similarity between two entities, we employ page count information provided by a Web search engine, which shows co-occurrence of the two entities in the Web. Also, since all entities in a lexical database such as WordNet are connected to other entities by means of semantic relations, we believe that using the lexical database can capture semantic similarity between two entities based on the distance between the two entities, *e.g.,* number of hops in WordNet's graph structure.

The contributions of this study are described as follows.
(1) To the best of our knowledge, this is the first study that recognizes object-based activities using egocentric videos without using any training data collected in an environment

978-1-5386-9151-9/19/$31.00 ©2019 IEEE

of interest. (2) We introduce query expansion techniques to facilitate computing semantic similarity between objects used by a user and a name of an activity class. (3) We fuse existing deep learning frameworks for object recognition and knowledge on the Web to recognize activities without using training data collected in an environment of interest.

II. Related Work

In the ubicomp research field, wearable technologies for object-based activity recognition have been actively studied. Maekawa *et al.* [11], [12] use a wrist-worn camera along with an accelerometer and microphone to detect objects used in object-based activities. In addition, Maekawa *et al.* [13], [14] recognize the use of electrical devices using hand-worn magnetic sensors.

In the computer vision research field, object-based activity recognition using egocentric videos have been actively studied. Pirsivash *et al.* [15] train classifiers for activities based on the output of a part-based model [16], which is a collection of object parts arranged in a deformable configuration. Ma *et al.* [17] develop a deep convolutional neural network (CNN) for activity recognition that deals with appearance information as well as hand motion information. All of the above approaches require labeled training data collected in a user's environment.

III. Method

A. Overview

An overview of our method is shown in Fig. 1. We estimate a user's activity for each time window. Because several egocentric images are included in a window, we recognize objects in each image and then construct a list of objects included in the images in the window. We then compute semantic similarity between the list and a name of an activity class. The activity class with the highest similarity becomes the estimated activity. Our proposed method has the following features.

1) To facilitate computing semantic similarity between recognized objects and a name of an activity class, we expand a name of an activity class by using names of objects expected to be used in the activity based on query expansion techniques.

2) Since we compute semantic similarity between a list of recognized objects and a list of objects expected to be used in an activity based on knowledge on the Web, this approach enables us to estimate the user activity without using training data collected in an environment of interest.

B. Expanding activity name

We expand a name of an activity class by using names of objects expected to be used in the activity. Because a name of an activity and names of objects used in the activity can be usually contained in the same Web document, we obtain the object names using a Web search engine.

We first retrieve documents simply using a name of each activity class as a query. To focus on Web documents related to daily activities, we retrieve documents only from how-to website, *i.e.,* wikiHow. From the retrieved how-to documents, we extract object names and rank the object names based on their importance. We construct a list of objects expected to be used in the activity concatenating the top-k_a objects.

Here we explain how we find object names in the retrieved documents. We obtain a list of object names from ImageNet ILSVRC-2012 dataset, which lists names of 1000 daily object classes, in advance. For each object name obtained from ImageNet, we compute its importance for the activity using the retrieved documents based on tf-idf, which is usually used to compute the importance of term t in document d. tf-idf is the product of term frequency and inverse document frequency. The term frequency $\mathrm{tf}(t,d)$ is the number of times that term t occurs in document d, meaning the relevance of document d for term t. The inverse document frequency $\mathrm{idf}(t,D)$ is a measure of the general importance of term t in collection of documents D, which is calculated as the logarithm of the number of documents in D, divided by the number of documents that contain term t.

In our method, we retrieve a set of documents D_n using a name of the nth activity class as a query. We compute the importance of object t using the documents based on tf-idf by

$$\sum_{d \in D_n} \text{tf-idf}(t,d,D) = \sum_{d \in D_n} \mathrm{tf}(t,d) \cdot \mathrm{idf}(t,D).$$

Note that D is a collection of wikiHow documents. Objects with the top-k_a importance values become a set of objects expected to be used in the activity.

As above, for the nth activity class, we obtain a set of objects A_n. Note that an object listed in ImageNet corresponds to a synset in WordNet, which is a set of synonyms. (We explain WordNet in detail later.) Therefore, an element of A_n corresponds to WordNet synset w_i.

C. Recognizing objects

We recognize objects in an egocentric image in a time window. Here, objects that do not relate to the user's activity can be included in the image. To detect objects related to the user's activity, we find an image region that the user is focusing on based on saliency maps because the user may focus on objects related to the activity. We then detect objects in the image region using CNN.

1) Saliency estimation: Since the head-mounted camera captures the face direction, objects used in an activity of the user may be located near the center of an egocentric image. Meanwhile, in the computer vision research field, salient locations in an image are extracted by mimicking humans' attentional mechanisms in order to reduce the search space and computational cost. Therefore, saliency estimation technique is usually used to find an image region that a user is paying attention to in an egocentric image in vision-based activity and object recognition studies [18]–[20]. Based on them, we detect a saliency region with a bias to the image center.

Fig. 2 shows an example input and output of the trained prediction model, and bright pixels in the output image flag salient locations in the input image. We simply crop the input image with a rectangle so that bright pixels in the saliency map are included in the rectangle, and the cropped rectangle is the detected salient image region.

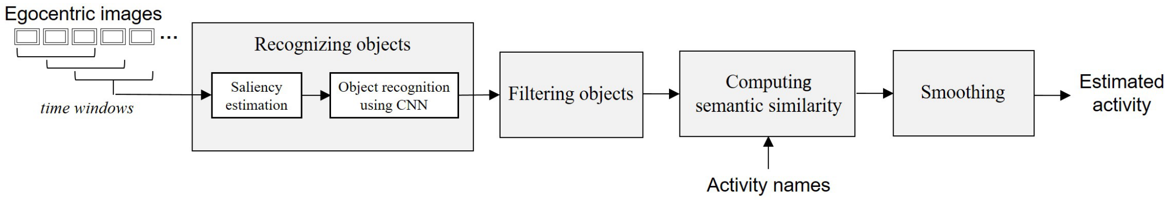


Fig. 1. Overview of estimating daily activity with egocentric video

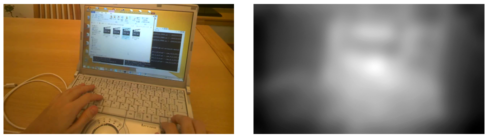
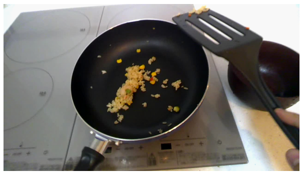


Fig. 2. Example of saliency prediction (right) of input image (left). Bright pixels flag salient locations. We simply assume that pixels whose brightness values are larger than a threshold as salient pixels.



object	score
wok	0.354
frying pan	0.240
spatula	0.055
steel drum	0.030
odometer	0.022

Fig. 3. Example of egocentric image and list of objects recognized by CNN

2) Object recognition using CNN: In a detected region in an egocentric image, we recognize objects using a pre-trained CNN. We then construct a set of objects combining the detected objects. CNNs can automatically learn feature representations and are now attracting considerable attention. In this study, we focus on caffe deep learning framework [21] and employ the CNN architecture pre-trained on the ImageNet ILSVRC-2012 dataset [10], [21], which achieves good recognition performance in general object recognition tasks. Using the pre-trained CNN permits us to recognize objects in an egocentric image without preparing training data in the user's environment.

The CNN used in this study outputs a set of objects included in an input image associated with their scores, which are computed by their corresponding activation functions. An object class used in the CNN corresponds to a synset of WordNet. Therefore, from an input image, we can obtain a set of synsets in WordNet and their scores. For example, Fig. 3 shows an output of the CNN for an input egocentric image.

Because multiple images are included in each time window and thus multiple object sets are obtained from the window, we construct an object set for the window combining the obtained sets. That is, for a window at time t, we obtain a set of detected objects O_t consisting of pairs of synset w_i and its score s_i. Note that the score is the sum of the scores in the object sets for the images in the time window.

D. Filtering objects

Because the user can sometimes look away when the user is performing an activity, O_t can include objects unrelated to the activity. We simply retain synsets with the top-k_o scores in O_t and remove the remaining synsets from O_t because low-score synsets are considered as estimation errors of the CNN or objects included in images captured when the user looks away.

E. Computing semantic similarity

We compute semantic similarity between O_t for a time window at time t and A_n for the nth activity. In this paper, we prepare two methods for measuring the semantic similarity and compare them in the evaluation section. These methods compute the similarity based on semantic similarities between an object included in O_t and an object in A_n.

1) Semantic similarity based on lexical database: We employ WordNet to compute semantic similarity between O_t and A_n. WordNet is an online lexical database, which groups words into sets of synonyms called synsets that are linked together by their semantic relationships. In WordNet's graph structure, synsets are the nodes of the graph, and relations among the synsets are the edges of the graph. Therefore, we can obtain semantic similarity between two synsets as the number of hops between the two synsets. Based on similarity between two synsets, we compute semantic similarity between O_t and A_n by

$$S_{wn}(O_t, A_n) = \sum_{w_i \in O_t} \sum_{w_j \in A_n} \frac{s_i}{d(w_i, w_j) + 1},$$

where s_i is a score of w_i in O_t and $d(w_i, w_j)$ is the number of hops between w_i and w_j.

2) Semantic similarity based on Web search engine: We employ a Web search engine to compute semantic similarity between O_t and A_n. This study employs Google search API to access a search engine. We compute the semantic similarity $S_{se}(O_t, A_n)$ using the mutual information, Jaccard coefficient, and Dice coefficient, which are usually used in the web mining research field to estimate semantic similarity between two terms. We detail and compare them in the evaluation section.

F. Smoothing

As above, we compute semantic similarity between O_t and A_n for each time window. To incorporate temporal regularity of activities, we smooth the computed similarity values for

TABLE I
ACTIVITIES PERFORMED IN OUR EXPERIMENT

A	watching television	H	playing with pet
B	using computer	I	making tea
C	using cellphone	J	watering plants
D	cooking	K	using curtain
E	eating	L	toilet
F	making coffee	M	washing dishes
G	using microwave		

each activity class. We simply smooth a time series of similarity values for an activity class using the moving average filter. The use of the moving average filter can smooth out sporadic errors. Finally, the class with the largest smoothed similarity value becomes the classified class at time t.

IV. EVALUATION

A. Data set

We collected a data set in three houses. In each environment, two participants collected egocentric videos with a Google Glass. (Environment 1 has only one participant.) The Google Glass captured 1280 by 720 pixel 24-bit color JPEG images at about 30 fps. The sampling rate of a three-axis accelerometer on the Glass was 30 Hz.

Here, the most natural data would be acquired from the normal daily lives in the environments. Since obtaining sufficient samples of such data is very costly, we collect sensor data by using a semi-naturalistic collection protocol [6] that permits greater variability in participant behavior than laboratory data. In the protocol, participants perform a random sequence of activities (obstacles) following instructions on a worksheet. They were granted much freedom regarding how they performed each activity because the instructions are relatively vague: "go to the toilet" or "watch TV." During the experimental period, the participants completed data collection sessions that included the random sequence of activities listed in Table I. The object-based activities were mainly selected from existing studies on object-based activity recognition [11], [15], [22]. The names of activities were also selected from the existing studies. In each environment, three-session data were collected.

B. Evaluation methodology

To investigate the effectiveness of our proposed method, we test the following methods.

- *WN+*: This method uses WordNet to compute semantic similarity between an activity name and an object list.

- *WN*: This method uses WordNet to compute semantic similarity. Note that this method does not expand an activity name. In this method, we first find a noun synset in the activity name or convert a verb in the activity name to a noun synset via a "morphosemantic" link in WordNet. Using the noun synset of the nth activity class a_n, we compute semantic similarity between O_t and a_n by

$$S_{wn}(O_t, w_{a_n}) = \sum_{w_i \in O_t} \frac{s_i}{d(w_i, w_{a_n}) + 1},$$

where w_{a_n} is a noun synset for the nth activity class.

- *WMI+*: This method uses a Web search engine to compute semantic similarity. In [23], semantic similarity between two terms is computed based on the mutual information, which measures the mutual dependence between two variables and is defined by

$$I(x, y) = \log \frac{p(x, y)}{p(x)p(y)}$$

if two words, x and y, have probabilities $p(x)$ and $p(y)$. When $h(q)$ is page count for query "q" provided by the search engine and $h(q_1, q_2)$ is page count for query "q_1 AND q_2," $p(x) = h(x)/W$ and $p(x, y) = h(x, y)/W$, where W is the number of indexed documents. Based on them, this method computes semantic similarity between O_t and A_n by

$$\begin{aligned} S_{se}(O_t, A_n) &= \sum_{w_i \in O_t} \sum_{w_j \in A_n} s_i I(w_i, w_j) \\ &= \sum_{w_i \in O_t} \sum_{w_j \in A_n} s_i \log W \frac{h(w_i, w_j)}{h(w_i)h(w_j)}. \end{aligned}$$

Note that, when we obtain page count for synset w, we use the first synonym included in synset w to form a query because it is the most common term for w.

- *WMI*: This method uses the mutual information to compute semantic similarity. Note that this method does not expand an activity name. Therefore, an activity name is simply used to construct a query. We compute semantic similarity between O_t and a_n by

$$S_{se}(O_t, a_n) = \sum_{w_i \in O_t} s_i I(w_i, a_n).$$

- *WJ+*: This method uses the Jaccard coefficient to compute semantic similarity based on page count information provided by a Web search engine. The Jaccard coefficient is computed based on page count information by

$$J(x, y) = \frac{h(x, y)}{h(x) + h(y) - h(x, y)}.$$

Therefore, we compute semantic similarity between O_t and A_n by

$$S_{se}(O_t, A_n) = \sum_{w_i \in O_t} \sum_{w_j \in A_n} s_i J(w_i, w_j).$$

- *WJ*: This method uses the Jaccard coefficient to compute semantic similarity. Note that this method does not expand an activity name.

- *WD+*: This method uses the Dice coefficient to compute semantic similarity based on page count information provided by a Web search engine. The Dice coefficient is computed based on page count information by

$$D(x, y) = \frac{2h(x, y)}{h(x) + h(y)}.$$

Therefore, we compute semantic similarity between O_t and A_n by

$$S_{se}(O_t, A_n) = \sum_{w_i \in O_t} \sum_{w_j \in A_n} s_i D(w_i, w_j).$$

TABLE II
ACTIVITY RECOGNITION ACCURACIES FOR METHODS

	Avg. precision	Avg. recall	Avg. F-measure
WN+	0.638	0.643	0.592
WN	0.339	0.452	0.359
WMI+	0.616	0.447	0.381
WMI	0.264	0.178	0.091
WJ+	0.644	0.603	0.563
WJ	0.329	0.307	0.223
WD+	0.643	0.589	0.558
WD	0.334	0.278	0.207
SL (LOSO)	0.847	0.858	0.852
SL (LOEO)	0.523	0.522	0.520

- *WD*: This method uses the Dice coefficient to compute semantic similarity. Note that this method does not expand an activity name.
- *SL*: This method relies on supervised machine learning techniques. Therefore, an activity classification model is trained on labeled egocentric videos. We extract a feature vector concatenating 4098 features obtained from the activations of the sixth hidden layer of the CNN used in the above methods. We then classify a feature vector for each egocentric image into an activity class using the C4.5 decision tree classifier [24]. We evaluate this method using "leave-one-session-out (LOSO)" cross validation and "leave-one-environment-out (LOEO)" cross validation.

C. Results

1) Activity recognition accuracy: Table II shows the activity recognition accuracies for the methods. Among the unsupervised methods, *WN*+ achieved the best performance. The precision of *WN*+ was higher than 60% and *WN*+ achieved almost the same performance as existing supervised activity recognition methods using egocentric videos, which were introduced in the related work section. While *SL* (LOSO) outperformed *WN*+, *WN*+ does not require training data collected and labeled by a user. In contrast, *SL* (LOEO) does not use labeled training data collected in an environment of interest. The performance of *SL* (LOEO) was poorer than that of *WN*+ because image features observed in different environments were also different.

2) Web search engine and lexical database: We first focus on the methods that do not use the activity name expansion. When we do not use the the activity name expansion, *WN* outperformed *WMI*, *WJ*, and *WD*. In the result of *WN*, many instances related to diet such as "cooking," "eating," "making coffee," and "making tea" were mistakenly classified into the "washing dishes" class. As for the "washing dishes" class, the "dish" synset was used to compute semantic similarity, *i.e.,* the "dish" synset is an expanded object for the "washing dishes" class. Because the distance between the "dish" synset and an object used in these activities such as cups and dishes was short, these instances were mistakenly classified into the "washing dishes" class.

TABLE III
RELATIONSHIP BETWEEN k_e AND F-MEASURE

k_e	*WN*+	*WMI*+	*WJ*+	*WD*+
1	0.524	0.350	0.376	0.374
2	0.592	0.381	0.563	0.558
3	0.568	0.460	0.536	0.527
4	0.577	0.411	0.486	0.490
5	0.549	0.311	0.553	0.556

TABLE IV
ACCURACIES WHEN WE DO NOT PERFORM SALIENCY PREDICTION

	Avg. precision	Avg. recall	Avg. F-measure
WN+	0.565	0.597	0.529
WMI+	0.464	0.388	0.289
WJ+	0.597	0.567	0.503
WD+	0.601	0.554	0.497

In the results of *WMI*, *WJ*, and *WD*, many instances were mistakenly classified into the "playing with pet" class. This is because "playing" is a common term and frequently co-occurs with names of many objects in the Web. Although we assume that activity names can be freely defined by a user, directly using activity names to compute semantic similarities does not work well. (In this evaluation, we use activity names defined in other activity recognition papers.)

3) Effect of activity name expansion: A large performance improvement of *WN* was observed by use of the activity name expansion. This is because the activity name expansion permits us to compute the distance between a detected object and an object expected to be used in an activity. As mentioned above, the distance between a synset corresponding to an object and a synset corresponding to an action (activity) is large in WordNet.

4) Number of expanded object names: Here we investigate the effect of k_e, *i.e.,* number of objects in A_t. Table III shows the relationship between k_e and F-measure. When $k_e = 2$, *WN*+, *WJ*+, and *WD*+ achieved the best performances. Also, the accuracies when $k_e = 1$ were poor because using only one object name for each activity degraded the classification accuracies for activities involving the use of multiple objects such as making coffee and tea.

5) Effect of saliency prediction: Table IV shows the classification accuracies when we do not perform the saliency prediction. As is seen in the results, a large performance improvement about 6-10% was observed by use of the saliency prediction.

6) Number of objects extracted from images: Here we investigate the effect of k_o, *i.e.,* number of objects in O_t. Table V shows the relationship between k_o and F-measure. As shown in the results, the effect of k_o was small. This may be because each object in O_t is associated with its score, and the effects of objects with low scores on the semantic similarity computation were small.

7) Effect of smoothing: Table VI shows the classification accuracies when we do not perform smoothing after computing

TABLE V
RELATIONSHIP BETWEEN k_o AND F-MEASURE

k_o	*WN*+	*WMI*+	*WJ*+	*WD*+
1	0.582	0.418	0.544	0.538
2	0.582	0.415	0.542	0.535
3	0.593	0.403	0.552	0.547
4	0.593	0.394	0.563	0.560
5	0.592	0.381	0.563	0.558

TABLE VI
ACCURACIES WHEN WE DO NOT PERFORM SMOOTHING

	Avg. precision	Avg. recall	Avg. F-measure
WN+	0.573	0.584	0.540
WMI+	0.589	0.471	0.396
WJ+	0.555	0.530	0.491
WD+	0.552	0.515	0.483

semantic similarities. As shown in the table, a performance improvement about 5% was observed by use of the smoothing.

V. CONCLUSION

This paper presented an activity recognition method based on egocentric videos without using training data prepared by a user in her environment. Our method first detects objects used in the user's activity using a publicly-available CNN for object recognition. The method then estimates an activity of the user using the object detection result, computing semantic similarity between a list of the detected object names and a name of each activity class based on knowledge on the Web. Our method could achieve about 60% precision *without* using training data collected in an environment of interest.

ACKNOWLEDGMENT

This work is partially supported by JST CREST JP-MJCR15E2, JSPS KAKENHI Grant Number JP16H06539 and JP17H04679.

REFERENCES

[1] E. M. Tapia, S. S. Intille, and K. Larson, "Activity recognition in the home using simple and ubiquitous sensors," in *Pervasive 2004*, 2004, pp. 158–175.

[2] M. Philipose, K. P. Fishkin, M. Perkowitz, D. J. Patterson, D. Fox, H. Kautz, and D. Hähnel, "Inferring activities from interactions with objects," *IEEE Pervasive Computing*, vol. 3, no. 4, pp. 50–57, 2004.

[3] T. Van Kasteren, A. Noulas, G. Englebienne, and B. Kröse, "Accurate activity recognition in a home setting," in *Ubicomp 2008*, 2008, pp. 1–9.

[4] T. Maekawa, Y. Yanagisawa, Y. Sakurai, Y. Kishino, K. Kamei, and T. Okadome, "Web searching for daily living," in *SIGIR 2009*, 2009, pp. 27–34.

[5] ——, "Context-aware web search in ubiquitous sensor environment," *ACM Transactions on Internet Technology (ACM TOIT)*, vol. 11, no. 3, pp. 12:1–12:23, 2012.

[6] L. Bao and S. S. Intille, "Activity recognition from user-annotated acceleration data," in *Pervasive 2004*, 2004, pp. 1–17.

[7] J. Lester, T. Choudhury, and G. Borriello, "A practical approach to recognizing physical activities," in *Pervasive 2006*, 2006, pp. 1–16.

[8] T. Maekawa and S. Watanabe, "Unsupervised activity recognition with user's physical characteristics data," in *International Symposium on Wearable Computers (ISWC 2011)*, 2011, pp. 89–96.

[9] J. Korpela, K. Takase, T. Hirashima, T. Maekawa, J. Eberle, D. Chakraborty, and K. Aberer, "An energy-aware method for the joint recognition of activities and gestures using wearable sensors," in *International Symposium on Wearable Computers (ISWC 2015)*, 2015, pp. 101–108.

[10] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *NIPS 2012*, 2012, pp. 1097–1105.

[11] T. Maekawa, Y. Yanagisawa, Y. Kishino, K. Ishiguro, K. Kamei, Y. Sakurai, and T. Okadome, "Object-based activity recognition with heterogeneous sensors on wrist," in *Pervasive 2010*, 2010, pp. 246–264.

[12] T. Maekawa, Y. Kishino, Y. Yanagisawa, and Y. Sakurai, "Wristsense: wrist-worn sensor device with camera for daily activity recognition," in *2012 IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops)*, 2012, pp. 510–512.

[13] T. Maekawa, Y. Kishino, Y. Sakurai, and T. Suyama, "Recognizing the use of portable electrical devices with hand-worn magnetic sensors," in *Pervasive 2011*, 2011, pp. 276–293.

[14] T. Maekawa, Y. Kishino, Y. Yanagisawa, and Y. Sakurai, "Recognizing handheld electrical device usage with hand-worn coil of wire," in *Pervasive 2012*, 2012, pp. 234–252.

[15] H. Pirsiavash and D. Ramanan, "Detecting activities of daily living in first-person camera views," in *CVPR 2012*, 2012, pp. 2847–2854.

[16] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1627–1645, 2010.

[17] M. Ma, H. Fan, and K. M. Kitani, "Going deeper into first-person activity recognition," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[18] P.-H. Tseng, R. Carmi, I. G. Cameron, D. P. Munoz, and L. Itti, "Quantifying center bias of observers in free viewing of dynamic natural scenes," *Journal of Vision*, vol. 9, no. 7, pp. 4–4, 2009.

[19] T. Judd, K. Ehinger, F. Durand, and A. Torralba, "Learning to predict where humans look," in *ICCV 2009*, 2009, pp. 2106–2113.

[20] T. Judd, F. Durand, and A. Torralba, "A benchmark of computational models of saliency to predict human fixations," Tech. Rep., 2012.

[21] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *ACM Multimedia 2014*, 2014, pp. 675–678.

[22] C. Luo, B. Ni, J. Wang, S. Yan, and M. Wang, "Manipulated object proposal: A discriminative object extraction and feature fusion framework for first-person daily activity recognition," *arXiv preprint arXiv:1509.00651*, 2015.

[23] K. W. Church and P. Hanks, "Word association norms, mutual information, and lexicography," *Computational linguistics*, vol. 16, no. 1, pp. 22–29, 1990.

[24] J. R. Quinlan, *C4.5: Programs for Machine Learning*. Morgan Kaufmann, 1993.