Anticipating Daily Intention using On-Wrist Motion Triggered Sensing

Tz-Ying Wu*, Ting-An Chien*, Cheng-Sheng Chan, Chan-Wei Hu, Min Sun
Dept. of Electrical Engineering,
National Tsing Hua University, Taiwan.
{gina9726, tingan0206, james121506, huchanwei1204, aliensunmin}@gmail.com

Abstract

Anticipating human intention by observing one's actions has many applications. For instance, picking up a cellphone, then a charger (actions) implies that one wants to charge the cellphone (intention) (Fig. 1). By anticipating the intention, an intelligent system can guide the user to the closest power outlet. We propose an on-wrist motion triggered sensing system for anticipating daily intentions, where the on-wrist sensors help us to persistently observe one's actions. The core of the system is a novel Recurrent Neural Network (RNN) and Policy Network (PN), where the RNN encodes visual and motion observation to anticipate intention, and the PN parsimoniously triggers the process of visual observation to reduce computation requirement. We jointly trained the whole network using policy gradient and cross-entropy loss. To evaluate, we collect the first daily "intention" dataset consisting of 2379 *videos with* 34 *intentions and* 164 *unique action sequences (paths in Fig. 1). Our method achieves* 92.68%, 90.85%, 97.56% *accuracy on three users while processing only* 29% *of the visual observation on average.*

1. Introduction

Thanks to the advance in Artificial Intelligence, many intelligent systems (e.g., Amazon Echo, Google Home.) have become available on the markets. Despite their great ability to interact with humans through a speech interface, they are currently not good at proactively interacting with humans. Thus, we argue that the key for proactive interaction is to anticipate user's intention by observing their actions. Given the anticipated intention, the intelligent system may provide service to facilitate the intention. More specifically, the ability to anticipate a large number of daily intentions will be the key to enable a proactive intelligent system.

Many researchers have tackled tasks related to intention anticipation. [11, 28, 18] focus on early activity prediction

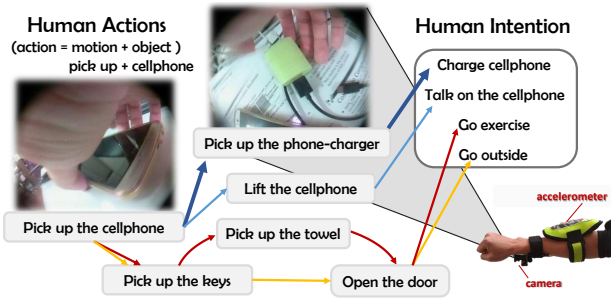

Figure 1. Illustration of intention anticipation. An action sequence (i.e., an ordered list of actions) is a strong cue to anticipate intention – predicting an intention before it occurs. For instance, the actions on the dark blue path (i.e., pick-up the cellphone; then, pick-up cellphone charger) strongly imply "charge cellphone". The task is challenging since (1) the same action (pick-up the cellphone) can lead to different intentions (talk on the cellphone vs. charge cellphone), and (2) multiple paths can lead to the same intention (see Fig. 5). Bottom-right panel: actions are recorded by our on-wrist sensors including a camera and an accelerometer.

– predicting actions before they have completed. However, the time-to-action-completion of this task is typically very short. Hence, there are only a few scenarios that intelligent systems may take advantage of the predicted activity. Kitani et al. [15] propose to forecast human's trajectory. Forecasting trajectory is very useful, but it does not directly tell you the "intention" behind a trajectory. [3, 12, 13] anticipate the future events on the road such as making a left turn or involving in an accident. Although these events can be considered as intentions, only few intentions (at most five) are studied. Moreover, none of the work above leverages heterogeneous sensing modalities to reduce computing requirement.

In this work, we anticipate a variety of daily intentions (e.g., "go outside", "charge cellphone", in Fig. 1) by sensing motion and visual observation of actions. Our method is unique in several ways. Firstly, we focus on *On-Wrist* sensing: (1) an on-wrist camera (inspired by [24, 2]) is used to observe object interactions reliably, and (2) an on-wrist accelerometer is used to sense 3D hand motion efficiently. Since both on-wrist sensors are unconventional, we collect auxiliary object appearance and motion data to pre-train two

*indicates equal contribution

encoders: (1) a Convolutional Neural Network (CNN) to classify daily objects, and (2) a 1D-CNN to classify common motions. Secondly, we leverage heterogeneous sensing modalities to reduce computing requirement. Note that visual data is very informative but costly to compute. In contrast, motion data is less informative but cheap to compute. We propose a Policy Network to determine when to peek at some images. The network will trigger the camera only at some important moments while continuously analyzing the motions. We call this as *Motion Triggered* sensing. Finally, we propose to use a Recurrent Neural Network (RNN) to model important long- and short-term dependency of actions. Modeling this dependency properly is the key of accurate anticipation, since daily action sequences are subtle and diverse. For instance, while multiple action sequences leading to the same intention, the same subset of actions can lead to different intention as well (see "go exercise" and "go outside" in Fig. 1).

In order to evaluate our method, we collect the first daily intention dataset from on-wrist sensors. It consists of 2379 videos with 34 intentions and 164 unique action sequences. For pre-training encoders, we collect an object dataset by manipulating 50[1] daily objects without any specific intention, and a 3D hand motion dataset with six motions performed by eight users. On the intention dataset, our method achieves $92.68\%, 90.85\%, 97.56\%$ accuracy while processing only 29% of the visual observation on average.

Our main contributions can be summarized as follows. (1) We adapt on-wrist sensors to reliably capture daily human actions. (2) We show that our policy network can effectively select the important images while only slightly sacrificing the anticipation accuracy. (3) We collected and will release one of the first daily intention dataset with a diverse set of action sequence and heterogeneous on-wrist sensory observations.

2. Related Work

We first describe works related to anticipation. Then, we mention other behavior analysis tasks. Finally, we describe a few works using wearable sensors for recognition.

2.1. Anticipation

The gist of anticipation is to predict in the future. We describe related works into groups as follows.

Early activity recognition. [11, 28, 18] focus on predicting activities before they are completed. For instance, recognizing a smile as early as the corners of the mouth curve up. Ryoo [28] introduces a probability model for early activity prediction. Hoai et al. [11] proposed a max-margin model to handle partial observation. Lan et al. [18] propose the hierarchical movemes representation for predicting future activities.

Event anticipation. [17, 13, 12, 3, 33] anticipate events even before they appear. Jain et al. [13, 12] propose to fuse multiple visual sensors to anticipate the actions of a driver such as turning left or right. Fu et al. [3] further propose a dynamic soft-attention-based RNN model to anticipate accidents on the road captured in dashcam videos. Recently, Vondrick et al. [33] propose to learn temporal knowledge from unlabeled videos for anticipating actions and objects. However, the early action recognition and anticipation approaches focus on activity categories and do not study risk assessment of objects and regions in videos. Bokhari and Kitani [1] propose to forecast long-term activities from a first-person perspective.

Intention anticipation. Intention has been explored more in the robotic community [35, 17, 16, 22]. Wang et al. [35] propose a latent variable model for inferring human intentions. Koppula and Saxena [17] address the problem by observing RGB-D data. A real robotic system has executed the proposed method to assist humans in daily tasks. [16, 22] also propose to anticipate human activities for improving human-robot collaboration. Hashimoto et al. [8] recently propose to sense intention in cooking tasks via the knowledge of access to objects. Recently, Rhinehart and Kitani [27] propose an on-line approach for first-person videos to anticipate intentions including where to go and what to acquire.

Others. Kitani et al. [15] propose to forecast human trajectory by surrounding physical environment (e.g., road, pavement). The paper shows that the forecasted trajectory can be used to improve object tracking accuracy. Yuen and Torralba [39] propose to predict motion from still images. Julian et al. [34] propose a novel visual appearance prediction method based on mid-level visual elements with temporal modeling methods.

Despite many related works, to the best of our knowledge, this is the first work in computer vision focusing on leveraging a heterogeneous sensing system to anticipate daily intentions with low computation requirement.

2.2. High-level Behavior Analysis

Other than activity recognition, there are a few high-level behavior analysis tasks. Joo et al. [14] propose to predict the persuasive motivation of the photographer who captured an image. Vondrick et al. [33] propose to infer the motivation of actions in an image by leveraging text. Recently, many methods (e.g., [38, 25, 26, 40, 32, 37]) have been proposed to generate sentence or paragraph to describe the behavior of humans in a video.

[1] including a hand free class, which means that hand is not interacting with any objects.

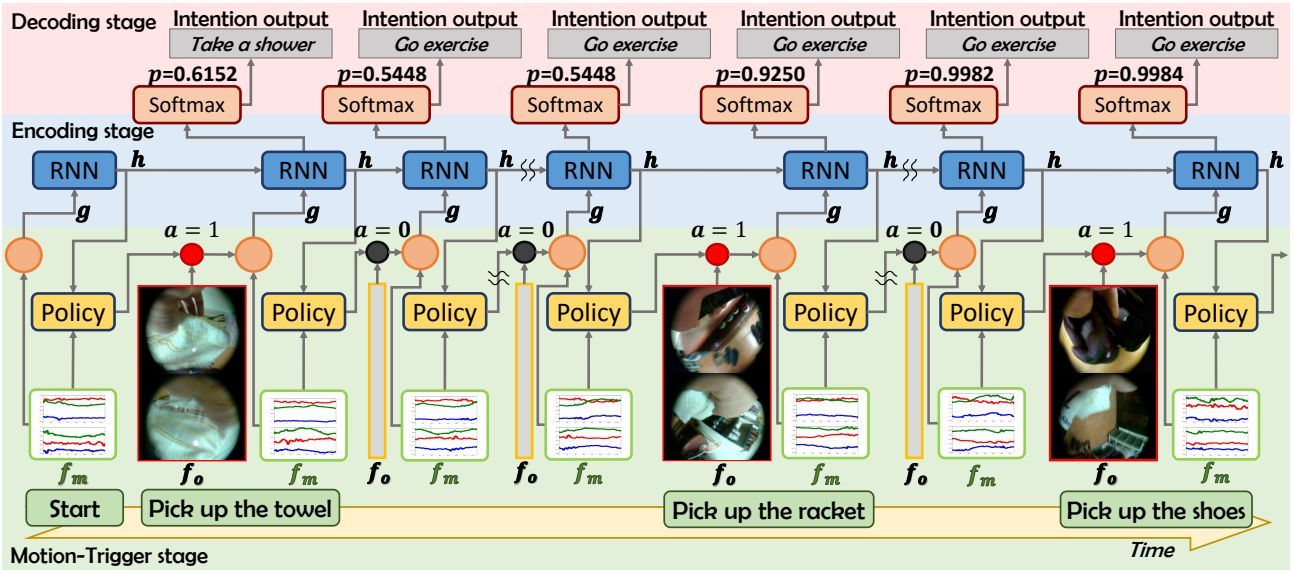

Figure 2. Visualization of our motion-triggered model. Our model consists of an RNN with LSTM cell encoder (blue block) and a Policy Network (yellow block). At each frame, RNN will generate an anticipated intention according to a new embedded representation g and the previous hidden state h of the RNN. The policy will generate the motion-trigger decision a for next frame, based on motion representation f_m and the hidden state h of the RNN. The orange circle represents the fusion operation (details in Sec. 3.2). The red and black circles represent a trigger and non-trigger decision of policy network, respectively (details in Sec. 3.3). When $a = 0$, f_o is empty since it is not processed.

2.3. Recognition from Wearable Sensors

Most wearable sensors used in computer vision are first-person (i.e., ego-centric) cameras. [23, 31, 6, 19] propose to recognize activities. [21, 7] propose to summarize daily activities. Recently, two works [24, 2] focus on recognition using on-wrist camera and show that it outperforms ego-centric cameras. Inspired by them, we adapt a similar on-wrist sensor approach.

3. Our Approach

We first define the problem of intention anticipation. Next, we introduce our RNN model encoding sequential observations and fusing multiple sensors' information from both hands. Then, we talk about our novel motion-triggered process based on a policy network. Finally, we describe how we pre-train the representation from auxiliary data.

3.1. Problem Formulation

Observations. At frame t, the camera observes an image I_t, and the motion sensor observes the 3D acceleration of hands $A_t \in R^3$.

Representations. The image I and 3D acceleration A are raw sensory values which are challenging to be used directly for intention anticipation, especially when lacking training data. Hence, we propose to learn visual object (referred to as object) $f_{o,t}$ and hand motion (referred to as motion) $f_{m,t}$ representations from other tasks with a larger number of training data. Note that, for all the variables, we use superscript to specify left or right hand (when needed). For instance, $f^L_{o,t}$ indicates left-hand object representation.

Goal. At frame t, our model predicts the future intention $y_t \in \mathcal{Y}$ based on the observations, where $\mathcal{Y}$ is the set of intention indices. Assuming the intention occurs at frame T, we not only want the prediction to be correct but also to predict as early as possible (i.e., $T - t$ to be large).

3.2. Our Recurrent and Fusion Model

Intention anticipation is a very challenging task. Intuitively, the order of observed objects and hand motions should be a very strong cue. However, most orders are not strict. Hence, learning composite orders from limited training data is critical.

Recurrent Neural Network (RNN) for encoding. We propose to use an RNN with two-layers of Long-Short-Term-Memory (LSTM) cell to handle the variation (Fig. 2-Top) as follows,

$$g_t = \text{Emb}(W_{emb}, \text{con}(f_{m,t}, f_{o,t})) \,, \quad (1)$$
$$h_t = \text{RNN}(g_t, h_{t-1}) \,, \quad (2)$$
$$p_t = \text{Softmax}(W_y, h_t) \,, \quad (3)$$
$$y_t = \arg\max_{y \in \mathcal{Y}} p_t(y) \,, \quad (4)$$

where $p_t \in R^{|\mathcal{Y}|}$ is the softmax probability of every intention in $\mathcal{Y}$, W_y is the model parameter to be learned, h_t is the learned hidden representation, and g_t is a fixed dimension output of Emb$(\cdot)$. W_{emb} is the parameter of embedding function Emb$(\cdot)$, con$(\cdot)$ is the concatenation operation, and Emb$(\cdot)$ is a linear mapping function (i.e.,

$g = W_{emb} \cdot \text{con}(f_m, f_o, 1)$. RNN has the advantage of learning both long- and short-term dependency of observation which is ideal for anticipating intentions.

Fusing left and right hands. Since tasks in real life typically are not completed by only one hand, we allow our system to observe actions on both hands simultaneously. We concatenate the right (i.e., the dominant hand) and left-hand observations in a fixed order to preserve the information of which hand is used for certain actions more frequently. The fused observation is $f_i = \text{con}(f_i^R, f_i^L)$, where $i \in \{o, m\}$.

Training for anticipation. Since our goal is to predict at any time before the intention happened, anticipation error at different time should be panelized differently. We use exponential loss to train our RNN-based model similar to [12]. The anticipation loss L^A is defined as,

$$\sum_{t=1}^{T} L_t^A = \sum_{t=1}^{T} -\log p_t(y^{\text{gt}}) \cdot e^{\log(0.1)\frac{T-t}{T}} , \qquad (5)$$

where y^{gt} is the ground truth intention and T is the time when intention reached. Based on this definition, the loss at the first frame (t=0) is only 10% of last frame (t=T). This implies that anticipation error is panelized less when it is early, but more when it is late. This encourages our model to anticipate the correct intention as early as possible.

The current RNN considers both motion f_m and object f_o representations as shown in Eq. 1. It is also straightforward to modify Eq. 1 such that RNN considers only motion or only object representation. However, the RNN needs to consider the same type of representation at all times. In the following section, we introduce the Motion-Triggered sensing process, where the RNN considers different representations at different frames depending on a learned policy.

3.3. RL-based Policy Network

We propose a policy network π to determine when to process a raw image observation I into an object representation f_o. The network continuously observes motion $f_{m,t}$ and hidden state of RNN h_t to parsimoniously trigger the process of computing $f_{o,t+1}$ as follows,

$$a_t = \arg\max_a \pi(a \mid (h_t, f_{m,t}); W_p) \in \{0, 1\} , \qquad (6)$$

$$\hat{f}_{o,t+1} = (1 - a_t) \cdot \hat{f}_{o,t} + a_t \cdot f_{o,t+1}(I_{t+1}) , \qquad (7)$$

$$g_{t+1} = \text{Emb}(W_{emb}, \text{con}(f_{m,t+1}, \hat{f}_{o,t+1})) , \qquad (8)$$

where a_t is the decision of our policy network to trigger ($a_t = 1$) or not trigger ($a_t = 0$), W_p is the parameters of the policy network, the policy π outputs a probability distribution over trigger ($a_t = 1$) or non-trigger ($a_t = 0$), and $\hat{f}_{o,t+1}$ is the modified object representation. As shown in Eq. 7, when $a_t = 1$, the visual observation at frame $t + 1$ will be updated ($\hat{f}_{o,t+1} = f_{o,t+1}(I_{t+1})$) with high cost on CNN inference. When $a_t = 0$, the previous representation will simply be kept ($\hat{f}_{o,t+1} = \hat{f}_{o,t}$). The modified object representation $\hat{f}_{o,t+1}$ will influence the embedded representation g_{t+1} as shown in Eq. 8.

Reward. We set our reward to encourage less triggered operation ($a = 1$) while maintaining correct intention anticipation ($y = y^{\text{gt}}$) as shown below.

$$R = \begin{cases} p_t(y^{\text{gt}}) \cdot R^+ \cdot (1 - \dfrac{n}{T}), & \text{if } y = y^{\text{gt}} \\ p_t(y^{\text{gt}}) \cdot R^- \cdot \dfrac{n}{T}, & \text{if } y \neq y^{\text{gt}} \end{cases} \qquad (9)$$

where y^{gt} is the ground truth intention, y is the predicted intention, n is the number of triggered operations in T frames of the video, p_t is the probability of anticipated intention, R^+ is a positive reward for correct intention anticipation, and R^- is a negative reward for incorrect intention anticipation. Note that, when the trigger ratio n/T is higher, the positive reward is reduced and the negative reward gets more negative.

Policy loss. We follow the derivation of policy gradient in [36] and define a policy loss function L^P,

$$L^P = -\frac{1}{KT} \sum_{k=1}^{K} \sum_{t=1}^{T} \log(\pi(a_t^k \mid (h_t^k, f_{m,t}^k); W_p)) \cdot R_t^k , \qquad (10)$$

where $\{a_t^k\}_t$ is the k^{th} sequence of trigged patterns sampled from $\pi(\cdot)$, K is the number of sequences, and T is the time when intention reached. R_t^k is the reward of the k^{th} sampled sequence at time t computed from Eq. 9. Please see Sec.2 of the supplementary material for the derivation.

Joint training. The whole network (Fig. 2) consists of a RNN and a policy network. We randomly initialize the parameters W_p of policy network. The parameters of RNN is initialized by the RNN encoder trained on both representation f_o and f_m. This initialization enables the training loss to converge faster. We define the joint loss $L = L^P + \lambda L^A$ for each training example, where λ is the weight to balance between two loss. Similar to the standard training procedure in deep learning, we apply stochastic gradient decent using mini-batch to minimize the total joint loss.

3.4. Learning Representations from Auxiliary Data

Due to the limited daily intention data, we propose to use two auxiliary datasets (object interaction and hand motion) to pre-train two encoders: an object Convolutional Neural Network (CNN) and a hand motion 1D-CNN. In this way, we can learn a suitable representation of object and motion.

Object CNN. It is well-known that ImageNet [5] pre-trained CNN performs well on classifying a variety of objects. However, Chan et al. [2] show that images captured by on-wrist camera are significantly different from images in ImageNet. Hence, we propose to collect an auxiliary image dataset with 50 object categories captured by our on-wrist camera, and fine-tuned on Imagenet [5] pre-trained

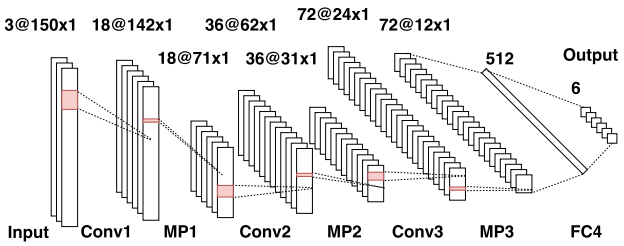

Figure 3. Illustration of our 1D-CNN pre-trained to classify six motions. Conv, MP, FC stand for Convolution, Max Pooling, and Fully Connected, respectively. $3@150 \times 1$ denotes that there are three 150×1 matrices. Since the second dimension is always one, it is a 1D-CNN. Our model has three stacks of Conv+MP layers and a FC layer at the end.

ResNet-based CNN [9]. After the model is pre-trained, we use the model to extract object representation f_o from the last layer before softmax.

Hand motion 1D-CNN. Our accelerometer captures acceleration in three axes ($s \in R^3$) with a sampling rate of 75Hz. We calibrate our sensor so that the acceleration in 3 axes are zero when we placed it on a flat and horizontal surface. We design a 1D-CNN to classify every 150 samples (2 seconds) into six motions: lift, pick up, put down, pull, stationary, and walking. The architecture of our model is shown in Fig. 3. Originally, we plan to mimic the model proposed by [4], which is a 3-layer 2D-CNN model with 1 input channel. Considering that there are no stationary properties among three acceleration values for each sample, we adjust the input channel number to 3 and define the 1D-CNN. For training the model, we have collected an auxiliary hand motion data with ground truth motions (Sec. 4). After the model is trained, we use the model to extract motion representation f_m at the FC4 layer (see Fig. 3).

3.5. Implementation Details

Intention anticipation model. We design our intention anticipation model to make a prediction in every half second. All of our models are trained using a constant learning rate 0.001 and 256 hidden states.

Policy Network. Our policy network is a neural network with two hidden layers. For joint training, we set learning rate 0.001, λ 0.1 for joint loss. The reward of R^+ and R^- are 100 and -100, respectively.

Object CNN. Following the setting of [2], our object CNN aims at processing 6 fps on NVIDIA TX1. This frame rate is enough for daily actions. Since most of the actions will last a few seconds, it's unnecessary to process at 15 or 30 fps. We take the average over 6 object representations as the input of our model. Different from [2], our on-wrist camera has a fish-eye lens to ensure a wide field-of-view capturing most objects. For fine-tuning the CNN model on our dataset, we set maximum iterations 20000, step-size 10000, momentum 0.9, every 10000 iteration weight decay 0.1, and learning rate 0.001. We also augment our dataset by horizontal flipping frames.

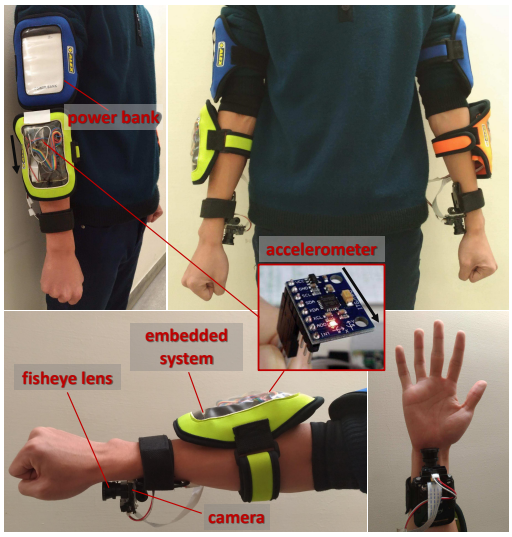

Figure 4. Our on-wrist sensing system. The fisheye camera is below the wrist. The embedded system and motion sensor are on the forearm. Both hands are equipped with the same system.

Hand motion 1D-CNN. Motion representation is extracted for a 2-second time segment. Hence, at every second, we process a 2-second time segment overlapped with previous processed time segment for 1 second. For training from scratch, we set the base learning rate to 0.01 with step-size 4000, momentum 0.9 and weight decay 0.0005. We adjust the base learning rate to 0.001 when fine-tuning.

4. Setting and Datasets

We introduce our setting of on-wrist sensors and describe details of our datasets.

4.1. Setting of On-wrist Sensors

Following similar settings in [24, 2], our on-wrist camera[2] and accelerometer[3] are mounted as shown in Fig. 4. Both camera and accelerometer are secured using velcro. We use the fisheye lens to ensure a wide field-of-view. We list some simple rules to be followed by users. First, the camera is under the arm, toward the palm. Second, the camera should roughly align the center of the wrist. This ensures that camera can easily record the state of the hand.

4.2. Datasets

We collect three datasets[4] for the following purposes. (1) Daily Intention Dataset: for training our RNN model to anticipate intention before the intention occurs. (2) Object Interaction Dataset: for pre-training a better object interaction encoder to recognize common daily object categories. (3) Hand Motion Dataset: for pre-training a better motion encoder to recognize common motions.

[2]fisheye lens mounted on noIR camera module with CSI interface.
[3]MPU-6050.
[4]Our dataset and code can be downloaded from `http://aliensunmin.github.io/project/intent-anticipate`

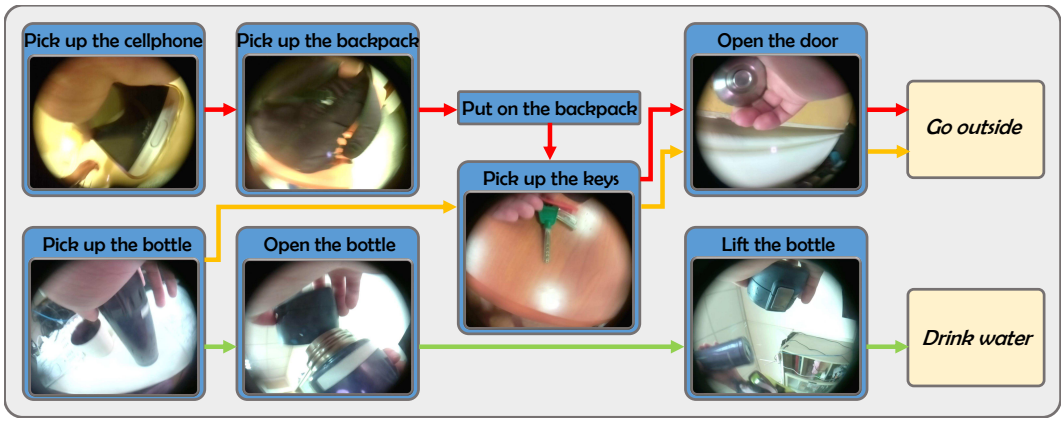

Figure 5. Daily intention dataset. We show examples of two action sequences (red line and yellow line) reaching to the same intention (go outside). In yellow line and green line, we show that the same object (bottle) involves in two intentions (go outside vs. drink water).

User	A	B	C
# of action sequences	1540	358	481
avg. per sequence	9.4	2.2	2.9

Table 1. Statistics of our intention dataset.

4.2.1 Daily Intention Dataset

Inspired by Sigurdsson et al. [29], we select 34 daily intentions such as charge cellphone, go exercise, etc. Note that each intention is associated with at least one action sequence, and each action consists of a motion and an object (e.g., pick up+wallet). We propose two steps to collect various action sequences fulfilling 34 daily intentions.

Exploring stage. At this stage, we want to observe various ways to fulfill an intention (Fig. 1). Hence, we ask a user (referred to as user A) to perform each intention as different as possible. At this step, we observed 164 unique action sequences.

Generalization stage. At this stage, we ask user A and other users (referred to as user B and user C) to follow 164 action sequences and record multiple samples[5] for each action sequence. This setting simulates when an intelligent system needs to serve other users. We show by experiment that our method performs similarly well on three users.

In Table 1, We summarize our intention dataset. Note that the number of action sequences recorded by user A is much more than others. Since we will train and validate on user A, selecting the proper hyper-parameters (e.g., design reward function). Next, we'll apply the same setting to the training process of all users, and evaluate the result. This can exam the generalization of our methods. Design of reward function is described in the Sec.3 of the supplementary material.

4.2.2 Object Interaction Dataset.

We select 50[6] object categories and collect a set of 940 videos corresponding to 909 unique object instances[7]. Each video records how an object instance is interacted by a user's hand. We sample 362 frames on average in each video. At the end, we collected an auxiliary dataset consisting of 340,218 frames in total to pre-train our object encoder. Example frames of the dataset are shown in Fig. 6.

[5] 10, 2, 3 times for user A, B, C, respectively
[6] including a hand-free category.
[7] not counting "free" as an instance.

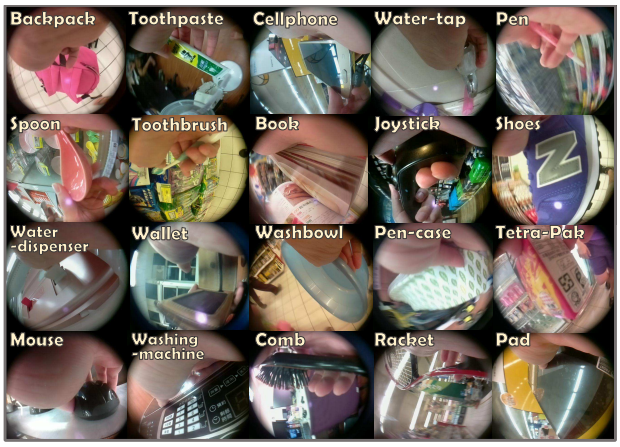

Figure 6. Auxiliary object dataset. Sample images overlaid with their ground truth categories.

4.2.3 Hand Motion Dataset

Inspired by [4], we select six motions. We ask eight users to collect 609 motion sequences from the right hand and one user to collect 36 motion sequences from the left hand. For the right-hand data collected by eight users, we aim at testing cross users generalizability. For the left-hand data, we aim at testing cross hand generalizability.

5. Experiments

We first conduct pilot experiments to pre-train object and hand motion encoders. This helps us to select the appropriate encoders. Next, we conduct experiments for intention anticipation with policy network and evaluate our method in various settings. Finally, we show typical examples to highlight the properties of our method.

Model	Training Acc.	Testing Acc.	Speed
VGG-16	98.58%	77.51%	4 fps
ResNet-50	**99.92**%	80.77%	**6** fps
ResNet-101	97.45%	82.79%	4 fps
ResNet-152	96.83%	**83.09**%	3 fps

Table 2. Classification accuracy and processing speed of different models. We highlight the best performance using bold font.

Model	Training Acc.	Testing Acc.
1ch-3layer [4]	100.00%	81.41%
3ch-1layer	100.00%	77.21%
3ch-2layer	100.00%	78.37%
3ch-3layer	100.00%	**83.92**%
left		52.78%
left-*flip*		**83.33**%

Table 3. Motion classification accuracy of different models. We highlight best performance using bold font.

5.1. Preliminary Experiments

Object pre-training. We evaluate multiple Convolution Neural Network (CNN) architectures on classifying 50 object categories in our object intention auxiliary dataset. These architectures include VGG-16 [30] and Residual Net (ResNet) [10] with 50, 101, 152-layers. We separate the whole dataset into two parts: 80% of object instances for training and 20% for testing. The testing accuracy is reported on Table. 2. Our results show that deeper networks have slightly higher accuracy. Another critical consideration is the speed on the embedded device. Hence, we report the processed frames per second (fps) on NVIDIA TX1 in the last column of Table. 2. Considering both accuracy and speed, we decide to use ResNet-50 since we designed our system to process at 6 fps similar to [2].

For hand motion, We describe two experiments to (1) select the best model generalizing across users, and (2) select the pre-processing step generalizing to the left hand.

Generalizing across users. Given our dataset collected by eight different users, we conduct a 4-fold cross validation experiment and report the average accuracy. We compare a recent deep-learning-based method [4] (1ch[8]-3layer model) with our 3ch models trained from scratch in Table. 3. The results show that our 3ch-3layer model generalizes the best across different users. At the end, we pre-train our 3-layer model on data collected by [20][9] to leverage more data. Then, we fine-tune the model on our auxiliary data.

Generalizing across hands. We propose the following pre-process to generalize our best model (3ch-3layer trained on right hand data) to handle left hand. We flip the left hand samples by negating all values in one channel (referred to as *flip*). This effectively flips left-hand samples to look similar to right-hand samples. In the last two rows of Table. 3, we show the accuracy of left-hand data. Our method with *flip* pre-processing achieves better performance. In the intention anticipation experiment, we use "3ch-3layer" and apply *flip* pre-process on left hand.

[8]ch stands for number of input channels

[9]Their data is collected by cellphone's accelerometer while the cellphone is in user's pocket.

5.2. Motion Triggered Intention Anticipation

For intention anticipation, we evaluate different settings on all three users. In the following, we first introduce our setting variants and the evaluation metric. Then, we compare their performance in different levels of anticipation (e.g., observing only the beginning X percent of the action sequence).

Setting variants.

(1) Object-only (**OO**): RNN considering only object representation f_o.

(2) Motion-only (**MO**): RNN considering only motion representation f_m .

(3) Concatenation (**Con.**): RNN considering both object f_o and motion f_m representations.

(4) Motion-Triggered (**MTr.**): RNN with policy network, where the input of RNN is determined by the policy network. In this setting, we also report the ratio of triggered moments (referred as to **Ratio**). The lower the ratio the lower the computation requirement.

Metric. We report the intention prediction accuracy when observing only the beginning 25%, 50%, 75%, or 100% of the action sequence in a video.

Comparisons of different variants on all users (referred to as user A, B, and C) are shown in Table. 4. We summarize our findings below. Object-only (**OO**) outperforms Motion-only (**MO**). This proves that object representation is much more influential than motion representation for intention anticipation. We also found that concatenating motion and object (**Con.**) does not consistently outperform Object-only (**OO**). Despite the inferior performance of **MO**, the tendency of **MO** under different percentage of observation is pretty steady. This implies that there are still some useful information in the motion representation. Indeed, **MTr.** can take advantage of motion observation to reduce the cost of processing visual observation to nearly 29% while maintaining a high anticipation accuracy $(92.68\%, 90.85\%, 97.56\%)$.

In Fig. 8, we control the ratio of triggered moments and change the anticipation accuracy by adjusting the threshold of motion triggers. The results show that increasing the ratio of triggered moments leads to higher accuracy on intention anticipation. Most interesting, the accuracy only decrease slightly when the ratio is larger than 20%. Note that the default threshold is 0.5, which means the policy will decide to trigger when the probability of trigger is larger than non-trigger. Some quantitative results are described in Sec.4 of the supplementary material.

	User A				User B				User C			
	25%	50%	75%	100%	25%	50%	75%	100%	25%	50%	75%	100%
Con.	88.41%	90.24%	92.07%	93.29%	90.85%	92.68%	94.51%	95.12%	97.56%	97.56%	98.17%	98.17%
OO	89.63%	92.68%	92.68%	94.51%	91.46%	94.51%	94.51%	95.73%	96.95%	96.95%	98.17%	98.17%
MO	65.85%	70.73%	75.61%	75.61%	62.20%	66.46%	69.51%	72.56%	71.34%	79.88%	85.37%	87.20%
Mtr.	86.58%	90.24%	92.07%	92.68%	84.75%	88.41%	88.41%	90.85%	94.51%	96.34%	97.56%	97.56%
Ratio	34.00%	32.34%	30.72%	28.42%	31.13%	33.23%	30.88%	29.67%	33.40%	33.88%	30.89%	29.17%

Table 4. Intention anticipation comparison. **OO** stands for object-only observation. **MO** stands for motion-only observation. **Con.** stands for concatenating f_o and f_m. **Mtr.** stands for motion-triggered. **Ratio** stands for triggered ratio. In the second row, 25% denotes only the beginning 25% of the action sequence is observed. All methods are evaluated on A, B, and C users. Note that **Mtr.** is significantly better than **MO** and only slightly worse than **Con.** while processing only about 29% of the frames.

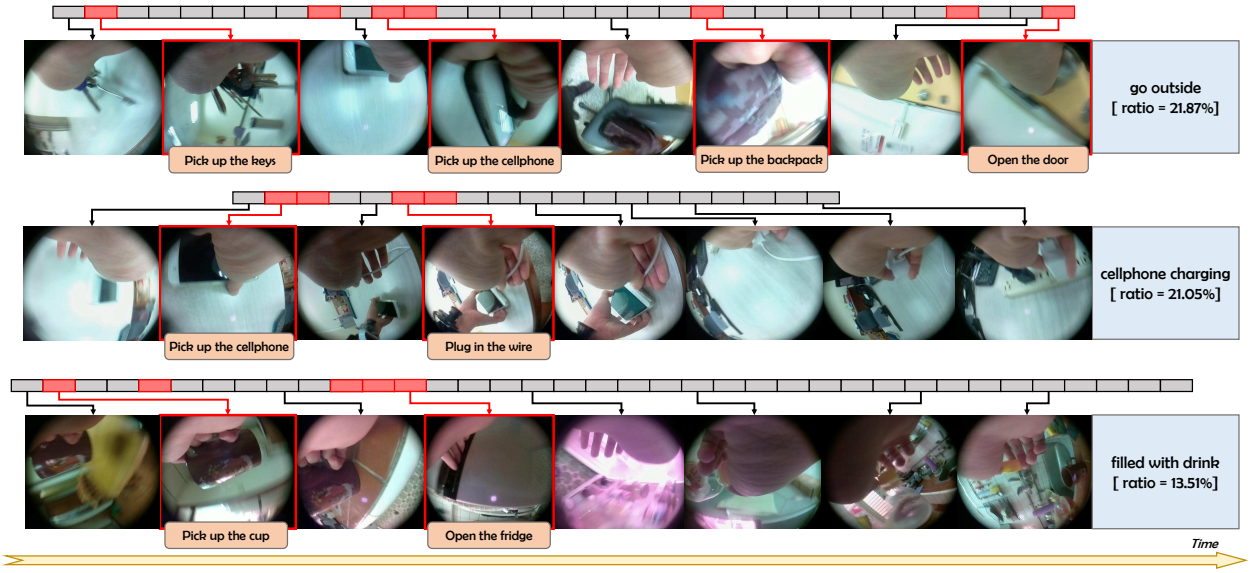

Figure 7. Typical Examples. In each row, we show an example of our motion-triggered method selecting visual observations. The gray block represents non-triggered frames, and red block represents triggered frames. Each block stands for half second. A few triggered (red boxes) and non-triggered (regular boxes) frames are visualized. At the end of each example, We show the trigger ratio and the correctly predicted intention. More results are shown in the Sec.1 of the supplementary material.

5.3. Typical Examples

We show typical examples in Fig. 7. In the first example, our Policy Network (PN) efficiently peeks at various objects (e.g., keys, cellphone, backpack, etc.). In other examples, PN no longer triggers after some early peeks. Specifically, in the second example, once the cellphone is observed and the wire is plugged in, PN is confident enough to anticipate cellphone charging without any further triggered operation.

6. Conclusion

We propose an on-wrist motion triggered sensing system for anticipating daily intentions. The core of the system is a novel RNN and policy networks jointly trained using policy gradient and cross-entropy loss to anticipate intention as early as possible. On our newly collected daily intention dataset with three users, our method achieves impressive anticipation accuracy while processing only 29% of the visual observation. In the future, we would like to develop an on-line learning based method for intention anticipation in the wild.

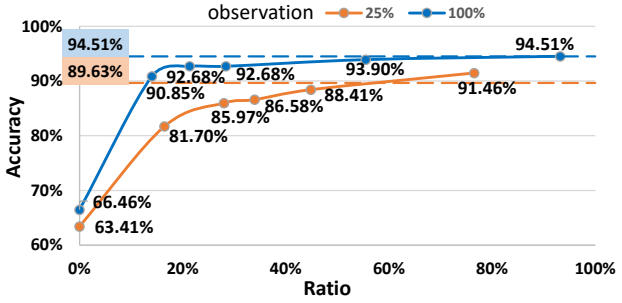

Figure 8. Anticipation accuracy (vertical axis) of our motion-triggered process on user A for sensing the beginning 25% (orange solid curves) and 100% (blue solid curves) of the action sequence. The horizontal axis is the triggered ratio from 0% (equals to motion-only process) to 100% (equals to motion-object combined process). We also show the accuracy of object-only process using dash curves.

7. Acknowledgement

We thank MOST 104-2221-E-007-089-MY2 and MediaTek for their support.

References

[1] S. Z. Bokhari and K. M. Kitani. Long-term activity forecasting using first-person vision. In ACCV, 2016.

[2] C.-S. Chan, S.-Z. Chen, P.-X. Xie, C.-C. Chang, and M. Sun. Recognition from hand cameras: A revisit with deep learning. In ECCV, 2016.

[3] F.-H. Chan, Y.-T. Chen, Y. Xiang, and M. Sun. Anticipating accidents in dashcam videos. ACCV, 2016.

[4] Y. Chen and Y. Xue. A deep learning approach to human activity recognition based on single accelerometer. In SMC, 2015.

[5] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In CVPR, 2009.

[6] A. Fathi, A. Farhadi, and J. M. Rehg. Understanding egocentric activities. In ICCV, 2011.

[7] J. Ghosh, Y. J. Lee, and K. Grauman. Discovering important people and objects for egocentric video summarization. In CVPR, 2012.

[8] A. Hashimoto, J. Inoue, T. Funatomi, and M. Minoh. Intention-sensing recipe guidance via user accessing objects. International Journal of Human-Computer Interaction, 2016.

[9] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In CVPR, 2015.

[10] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In CVPR, 2016.

[11] M. Hoai and F. De la Torre. Max-margin early event detectors. In CVPR, 2012.

[12] A. Jain, H. S. Koppula, B. Raghavan, S. Soh, and A. Saxena. Car that knows before you do: Anticipating maneuvers via learning temporal driving models. In ICCV, 2015.

[13] A. Jain, A. Singh, H. S. Koppula, S. Soh, and A. Saxena. Recurrent neural networks for driver activity anticipation via sensory-fusion architecture. In ICRA, 2016.

[14] V. Joo, W. Li, F. F. Steen, and S.-C. Zhu. Visual persuasion: Inferring communicative intents of images. In CVPR, 2014.

[15] K. M. Kitani, B. D. Ziebart, J. A. D. Bagnell, and M. Hebert . Activity forecasting. In ECCV, 2012.

[16] H. S. Koppula, A. Jain, and A. Saxena. Anticipatory planning for human-robot teams. In ISER, 2014.

[17] H. S. Koppula and A. Saxena. Anticipating human activities using object affordances for reactive robotic response. PAMI, 38(1):14–29, 2016.

[18] T. Lan, T.-C. Chen, and S. Savarese. A hierarchical representation for future action prediction. In ECCV, 2014.

[19] Y. Li, Z. Ye, and J. M. Rehg. Delving into egocentric actions. In CVPR, 2015.

[20] J. W. Lockhart, G. M. Weiss, J. C. Xue, S. T. Gallagher, A. B. Grosner, and T. T. Pulickal. Design considerations for the wisdm smart phone-based sensor mining architecture. In Proceedings of the Fifth International Workshop on Knowledge Discovery from Sensor Data, 2011.

[21] Z. Lu and K. Grauman. Story-driven summarization for egocentric video. In CVPR, 2013.

[22] J. Mainprice and D. Berenson. Human-robot collaborative manipulation planning using early prediction of human motion. In IROS, 2013.

[23] K. K. Minghuang Ma. Going deeper into first-person activity recognition. In CVPR, 2016.

[24] K. Ohnishi, A. Kanehira, A. Kanezaki, and T. Harada. Recognizing activities of daily living with a wrist-mounted camera. In CVPR, 2016.

[25] P. Pan, Z. Xu, Y. Yang, F. Wu, and Y. Zhuang. Hierarchical recurrent neural encoder for video representation with application to captioning. In CVPR, 2016.

[26] Y. Pan, T. Mei, T. Yao, H. Li, and Y. Rui. Jointly modeling embedding and translation to bridge video and language. In CVPR, 2016.

[27] N. Rhinehart and K. M. Kitani. First-person activity forecasting with online inverse reinforcement learning. ICCV, 2017.

[28] M. S. Ryoo. Human activity prediction: Early recognition of ongoing activities from streaming videos. In ICCV, 2011.

[29] G. A. Sigurdsson, G. Varol, X. Wang, A. Farhadi, I. Laptev, and A. Gupta. Hollywood in homes: Crowdsourcing data collection for activity understanding. In ECCV, 2016.

[30] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. CoRR, abs/1409.1556, 2014.

[31] S. Singh, C. Arora, and C. V. Jawahar. First person action recognition using deep learned descriptors. In CVPR, 2016.

[32] S. Venugopalan, M. Rohrbach, J. Donahue, R. J. Mooney, T. Darrell, and K. Saenko. Sequence to sequence - video to text. In ICCV, 2015.

[33] C. Vondrick, D. Oktay, H. Pirsiavash, and A. Torralba. Predicting motivations of actions by leveraging text. In CVPR, 2016.

[34] J. Walker, A. Gupta, and M. Hebert. Patch to the future: Unsupervised visual prediction. In CVPR, 2014.

[35] Z. Wang, M. Deisenroth, H. Ben Amor, D. Vogt, B. Schölkopf, and J. Peters. Probabilistic modeling of human movements for intention inference. In RSS, 2012.

[36] R. J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. Machine Learning, 8(3):229–256, 1992.

[37] L. Yao, A. Torabi, K. Cho, N. Ballas, C. Pal, H. Larochelle, and A. Courville. Describing videos by exploiting temporal structure. In ICCV, 2015.

[38] H. Yu, J. Wang, Z. Huang, Y. Yang, and W. Xu. Video paragraph captioning using hierarchical recurrent neural networks. In CVPR, 2016.

[39] J. Yuen and A. Torralba. A data-driven approach for event prediction. In ECCV, 2010.

[40] K.-H. Zeng, T.-H. Chen, J. C. Niebles, and M. Sun. Title generation for user generated videos. In ECCV, 2016.