Temporal Perception and Prediction in Ego-Centric Video

Yipin Zhou Tamara L. Berg
University of North Carolina at Chapel Hill
Chapel Hill, NC, United States
[yipin,tlberg]@cs.unc.edu

Abstract

Given a video of an activity, can we predict what will happen next? In this paper we explore two simple tasks related to temporal prediction in egocentric videos of everyday activities. We provide both human experiments to understand how well people can perform on these tasks and computational models for prediction. Experiments indicate that humans and computers can do well on temporal prediction and that personalization to a particular individual or environment provides significantly increased performance. Developing methods for temporal prediction could have far reaching benefits for robots or intelligent agents to anticipate what a person will do, before they do it.

1. Introduction

Accurate visual recognition of image and video content is becoming a reality: there have been significant recent advancements in algorithms to recognize content elements, such as, objects [21], scenes [42], or attributes [41]. Much of this work focuses on estimating what's *in the frame*, that is, information that is directly visible within the image or video. Alas, an image captures only a thin slice of reality, and much of true human-level visual understanding is about what happens *beyond the frame*: making inferences about the broader contexts, e.g. spatial, temporal, social, etc, suggested by a given visual input.

In this paper we explore predicting *temporal* context beyond the frame, examining how well humans and computers can make predictions about what will happen next during everyday activities. Such reasoning will be necessary for producing intelligent agents that can understand what a person is doing and anticipate what they are likely to do next. Both types of inference can help produce robots that more naturally interact with humans in our daily lives.

There has been a great deal of previous work on recognizing activities from general video [23, 7, 37, 22, 20] and egocentric video [30, 9, 18, 33, 10]. Some recent work has started to look at the task of predicting what might happen next in video, focusing on specific tasks like predicting future trajectories of cars [36], pedestrians [19], or general moving objects [40].

In our work we introduce two tasks related to temporal prediction. In the first task, given two short video snippets of an activity, the goal is to predict their correct temporal ordering. In the second task, given a longer context video plus two video snippets sampled from before or after the context video, the goal is to predict which video snippet was captured closest in time after the context video. This task models the scenario of predicting what a person will do next in ego-centric video.

These tasks have several advantages. They provide a measure of video understanding that is complementary to standard activity recognition with tasks that do not require semantic labels, but are easy to evaluate. They also provide quantitative measures for temporal prediction, one challenging aspect of general scene understanding. The tasks are also designed so that we can ask (multiple) people to perform the same tasks as the algorithms, allowing us to measure human performance on temporal prediction. In summary, this provides an initial first step toward enabling computers to understand the temporal nature of videos.

We provide a variety of experiments to evaluate the temporal prediction tasks. First we evaluate human performance under several different scenarios. Given our findings, we design several different computational methods for temporal prediction using state of the art deep learning features selected to capture a range of different types of video content. To enable these experiments, we collect a new dataset of ego-centric videos of everyday activities. This dataset allows us to evaluate both general models for temporal prediction and prediction models personalized to a particular individual or environment.

In summary, our contributions are:

- Definition of two new tasks for temporal prediction in video: pairwise ordering and future prediction.
- A new dataset of ego-centric videos of everyday activities, including both individuals and families living in the same location.
- Experiments to evaluate human performance on each of the proposed temporal prediction tasks.
- Evaluation of deep features for object, scene, and motion estimation incorporated into several classification methods for pairwise ordering and future prediction.

The rest of our paper is organized as follows. We first review related work (Sec 1.1). Then we introduce the temporal prediction tasks (Sec 2.1) and our first person personalized activity dataset (Sec 2.2). Next we perform experi-

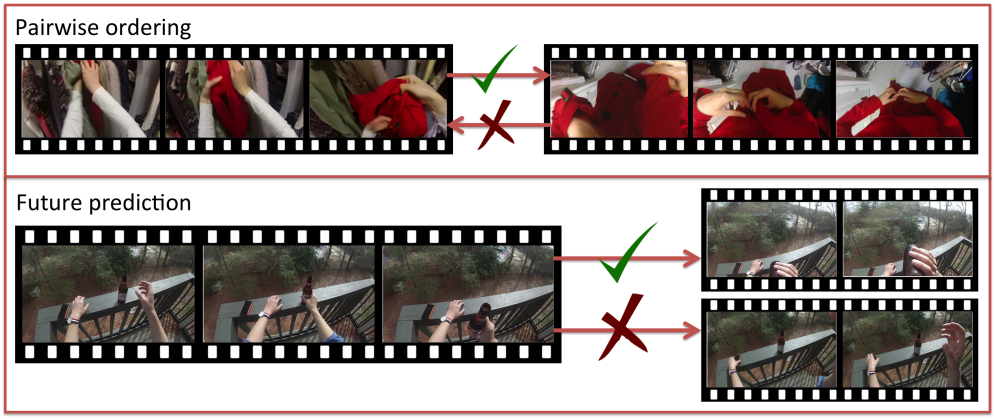


Figure 1: Illustration of our two temporal prediction tasks for ego-centric video of everyday activities. In the pairwise ordering task (**above**) the goal is to provide the correct temporal ordering for two short snippets of video from an activity. In the future prediction task (**below**), given a longer context video of an activity and two video snippets, the goal is to determine which snippet will occur (closest in time) after the context video.

ments to understand human performance on pairwise ordering (Sec 3). Finally, we evaluate performance on pairwise-ordering (Sec 4) and future prediction (Sec 5).

1.1. Related Work

Prediction: Recently, the idea of predicting the future has been introduced for tasks such as estimating the future trajectories of cars [36], pedestrians [19, 39], or general objects [40] in images or videos. In a related goal, early detection methods, have aimed at predicting human behaviors such as facial expressions or other activities in early stages of the action [16, 32]. More generally, some methods have looked at ideas related to human intent and goal inference for prediction using Stochastic Context Sensitive Grammars [28] or social models of behavior [39].

Temporal Ordering Recovery: Many methods for activity recognition have incorporated temporal information through the use of spatio-temporal descriptors [22, 7, 37]. A more complete survey is provided Wang et al [38]. However, there has been relatively less research on trying to recover temporal orderings. The work in [29] aims to tell whether a video is running forwards or backwards based on low-level visual information. [13] shows that ranking models learned from frame orderings are useful for the activity recognition problem. More relevant to our ordering prediction task, [5, 4] have provided interesting methods for trying to recover the temporal ordering for a set of photos with consistently moving objects using geometric methods. Our work extends these goals to recover temporal orderings for video.

Egocentric video analysis: First person video analysis has attracted increasing attention due to the rich information contained in egocentric visual data and easier access to wearable recording devices. There has been recent work on object recognition [12, 31], activity recognition [30, 9, 33], gaze prediction [25], and interaction detection [10] in first person video. In addition, there has been interest in first person video as a form of life-blogging, leading to methods for egocentric video summarization [24, 26]. A more complete survey of this area is provided by Betancourt et al [2].

2. Temporal Prediction Tasks & Dataset

We first introduce our tasks (Sec 2.1) and a new dataset of first person everyday activity videos (Sec 2.2).

2.1. Temporal Prediction tasks

We design two tasks to evaluate temporal understanding. Our goals for designing these tasks are two-fold: 1) we would like to create tasks that are easy to evaluate, and 2) we would like the tasks to be answerable by people so that we can evaluate both human and computer performance.

The first task is a pairwise ordering scenario. In this task, we are given two short snippets from an ego-centric video of an activity and asked to infer their correct temporal ordering. For example, in Fig 1 the upper box shows two example snippets from the activity of "putting on clothes". Obviously the left snippet occurs before the right snippet in temporal order. Being able to infer the correct temporal ordering between pairs of video snippets is important since it can provide a backbone algorithm for interpreting the temporal information of an entire video sequence or for predicting what might occur next in a video.

The second task directly evaluates our ability to make future predictions for everyday activities. In this task, we are provided with a video showing part of an activity plus two shorter video snippets and asked to predict which of the snippets comes next temporally in the video. Fig 1 (lower box) shows an example of a person grabbing a bottle and raising it toward himself. Temporal predictions should tell us that the drinking action is more likely to occur (closest in time) after the context video than the other snippet.

Figure 2: Example frames of 5 activities from our dataset. People perform everyday activities in various locations, and according to their own preferences, e.g. when putting on shoes people might squat down or stand or use a chair.

2.2. Dataset

There are several existing egocentric datasets [12, 30, 11, 10, 24] designed for a variety of purposes, such as object recognition, activity recognition, social interaction detection and video summarization. Many of these datasets record daily life activities. However, due to the complexity of data collection, most of the datasets contain only one or a few examples of each activity performed by each subject.

One observation that we would like to take advantage of in our models is that we spend much of our daily lives engaged in extremely repetitive activities. Every morning most of us get up, brush our teeth, take a shower, put on clothing, make and eat breakfast, and so on throughout the day. On the other hand, this repetitiveness is offset by the fact that each of us may perform these activities with different variations. For example, one person might prepare cereal for breakfast while someone else may prefer toast. Some people will use a chair to put on their shoes while others may put on their shoes while standing. All of these factors make for wide variation even in common everyday activities. However, one insight is that although these factors, along with variations due to environment, may vary from person to person, a particular person might be quite consistent in how they perform each activity and in the locations in which they perform the activities.

Therefore, along with building general temporal prediction models, we would also like to build models for prediction that can be personalized, either to a particular individual or to a particular environment. To enable this, we have collected a dataset of first person videos of everyday activities, called the First Person Personalized Activities (FPPA) dataset, where each subject has performed every activity many times. We make use of this dataset for evaluating temporal prediction tasks, but it could also be used for general or personalized activity recognition.

2.2.1 Data collection

For data collection, we make use of a GoPro camera mounted on the user's head using a head strap. The GoPro cameras record ego-centric data simulating the wearer's viewpoint. Each camera captures a high-definition video at 1080p (1920x1080 resolution) with a wide field of view (133.6 degrees) at a rate of 30 frames per second.

Activities	Avg No.of videos/sub	Avg No.of locs/sub	Totall No.of videos/locs
Wash hands	24.2 (19-34)	3.2 (2-7)	121/16
Put on shoes	22.8 (21-29)	3.0 (2-6)	114/15
Use fridge	26.4 (21-31)	1.6 (1-3)	132/8
Drink water	23.2 (16-31)	3.6 (2-7)	116/18
Put on clothes	21.6 (16-26)	3.4 (2-5)	108/17

Table 1: Statistics of FPPA dataset. The contents in brackets show the minimal to maximal numbers. The total number of video clips in the dataset is 591.

We first provide each subject with a list of 5 daily activities (a list of activities is shown in Table 1). To encourage subjects to act naturally, they are not provided with any more details. Subjects are encouraged to film themselves completing each activity multiple times at different locations where they would normally perform them. Video recordings were captured in the subjects own homes or in public places (gym, lounge) that they usually visit. In total, our data is made up of 5 sets of videos. Two of the sets consist of videos from a single individual (single-subject) while 3 of the sets consist of videos from families, i.e. two or more people living together at the same location (family-subject). We spread the data collection procedure of each set over two months to encourage variability and to gather a large number of videos for each subject.

2.2.2 Characteristics and statistics

The main characteristic of the FPPA dataset is that it is built to enable learning both general and personalized models for temporal prediction. As such we collect a large number of examples of each activity from each subject and for each location. Table 1 shows statistics of our dataset, including the per subject average number of the video clips for each activity, and the average number of locations in which each subject performed the activity. On average subjects have performed each activity approximately 20 times. As habits vary from subject to subject, some subjects have performed an activity in a single location while others have performed them in up to 7 different locations. Figure 2 shows some example frames from activities performed by different subjects.

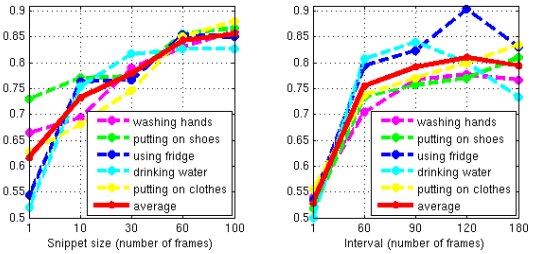


Figure 3: Human performance on pairwise ordering. Left shows performance as snippet size varies. Right shows performance as interval varies.

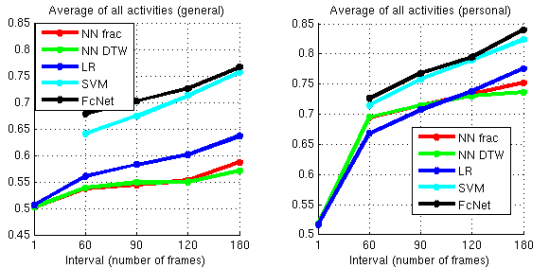


Figure 4: Model performance on pairwise ordering. Left shows prediction performance of general models (trained on other subjects). Right shows personalized model performance (trained on other videos from the same subject).

3. Human experiments

Our main goal is to give computers the ability to predict temporal information for video. Toward this aim, we start with a straightforward pairwise ordering task. However, before we can design even this simple task we would like to know several things: the feasibility of this task for people, and what specific implementation features should be used for the task, e.g. what length snippets should we use, or how far separated in time should the snippets be so that they are still temporally distinguishable? Therefore, we design two human experiments to gain some useful insights into the pairwise ordering task and then configure our pairwise ordering (Sec 4) and future prediction (Sec 5) tasks based on these analyses.

3.1. Snippet size

We design an experiment to evaluate the effect of snippet length on human perceptions of pairwise ordering using Amazon Mechanical Turk (AMT) as our crowdsourcing platform. For each activity we randomly pick 100 pairs of snippets from videos of the activity, where the central frame for each snippet is selected at a random temporal position within the video. We vary snippet size as 1, 10, 30, 60, or 100 frames (snippet size 1 is a static image). For each pair of snippets we ask 3 AMT workers to tell us which snippet should come first in temporal ordering. To limit bias, we randomly reshuffle the left-right placement of snippets in our Turk interface, and only allow workers to see one snippet length for a particular pair of snippets.

Figure 3 (left) shows the effect of snippet size on human performance. One interesting observation is that for the "using fridge" and "drinking water" activities there is an obvious increase in human performance between a snippet size of 1 (static image) and snippet size of 10 frames. One reason for this could be that these activities are relatively symmetric so it may be difficult to tell from a single frame whether the person is opening or closing the fridge, or picking up or putting down a cup. For a video snippet, motion cues help people resolve these ambiguities.

Based on these experiments, we find that human performance for pairwise ordering increases greatly past single frame snippets, but then levels off at about 60 frames. For length 60 frame snippets we find that average performance across users is about 80 to 85% for all activities. Therefore, for the rest of our human and computer experiments we use a snippet length of 60 frames.

3.2. Snippet Interval

Our next experiment explores how the temporal distance between two snippets affects human pairwise ordering performance. For this experiment, we keep the snippet size fixed to 60 frames, but vary the time interval between the selected snippet pairs. In particular, for each activity we select 100 video snippet pairs. For this sampling, we first select a random video, and a random first snippet. Then we extract the second snippet for the pair from later in the same video with intervals between center frames of 1, 60, 90, 120, or 180 frames. We limit Turker bias in the same manner as the previous experiment.

The results of this experiment are shown in Fig 3 (right). Agreeing with intuition, when the temporal offset between two snippets is extremely small (e.g., 1 frame), it is difficult for people to predict the correct pairwise ordering between snippets. As the interval between snippets increases, human performance also increases, but then levels off or even drops as the interval between the snippets gets very long (e.g., 180 frames). For activities such as "putting on clothes" which have clear steps in a relatively lengthy procedure, larger intervals tend to increase performance. For activities like drinking water, the accuracy initially rises with larger intervals then decreases for longer intervals. There are several potential reasons for this: (1) the most distinctive parts of the procedure may be very short, or (2) this activity is periodic (people may repeat the sipping portion of the action multiple times in one drinking activity). Both reasons can cause ambiguity for longer intervals.

4. Pairwise Ordering Task

We now design our computational pairwise ordering task. For this task we investigate several different representations for video snippets, to extract information about depicted objects, scenes, and motion (Sec 4.1). We also evaluate performance of 4 different classification methods (Sec 4.2) for building general pairwise ordering models

Methods	Washing hands	Putting on shoes	Using fridge	Drinking water	Putting on clothes	Average
NN frac(O)	0.5707	0.5257	0.5494	0.5459	0.5604	0.5504
NN frac(OS)	0.5879	0.5402	0.5468	0.5432	0.5421	0.5521
NN frac(OSM)	0.5694	0.5299	0.5884	0.4938	0.5807	0.5525
NN DTW(O)	0.5835	0.5302	0.5400	0.4709	0.5501	0.5350
NN DTW(OS)	0.5610	0.5447	0.5387	0.5335	0.5533	0.5462
NN DTW(OSM)	0.5738	0.5257	0.5884	0.4855	0.5758	0.5499
LR(O)	0.5635	0.5502	0.6296	0.5844	0.6034	0.5862
LR(OS)	0.5550	0.5853	0.6233	0.5732	0.5923	0.5858
LR(OSM)	0.5536	0.5964	0.6583	0.6039	0.5950	0.6014
SVM(O)	0.6392	0.7026	0.7351	0.6848	0.6967	0.6917
SVM(OS)	0.6609	0.7270	0.7435	0.6845	**0.7044**	0.7041
SVM(OSM)	0.6749	0.6945	0.7402	0.7538	0.7006	0.7128
FcNet(O)	0.6506	0.7166	0.7648	0.7045	0.6710	0.7015
FcNet(OS)	0.6857	0.7188	0.7045	0.6650	0.6943	0.6936
FcNet(OSM)	**0.6775**	**0.7274**	**0.7473**	**0.7828**	0.6966	**0.7263**

Table 2: Accuracy of pairwise temporal ordering using general prediction models (trained and tested on different subjects) for interval size 120. Here O indicates object features, S scene features, and M motion features

Methods	Washing hands	Putting on shoes	Using fridge	Drinking water	Putting on clothes	Average
NN frac(O)	0.7398	0.7043	0.7491	0.6602	0.7659	0.7239
NN frac(OS)	0.7417	0.7142	0.7626	0.6671	0.7774	0.7326
NN frac(OSM)	0.7304	0.6867	0.7782	0.6934	0.7806	0.7339
NN DTW(O)	0.7547	0.7138	0.7369	0.6616	0.7703	0.7280
NN DTW(OS)	0.7534	0.7010	0.7525	0.6681	0.7784	0.7307
NN DTW(OSM)	0.7288	0.6841	0.7705	0.6942	0.7731	0.7302
LR(O)	0.7118	0.7551	0.8029	0.7233	0.7773	0.7541
LR(OS)	0.6903	0.7425	0.7784	0.6911	0.7567	0.7318
LR(OSM)	0.6872	0.7235	0.8018	0.7248	0.7528	0.7380
SVM(O)	0.7548	0.7822	0.8136	0.8275	0.8187	0.7994
SVM(OS)	0.7526	0.7583	0.8208	0.8264	0.8184	0.7953
SVM(OSM)	0.7142	0.7433	0.8214	**0.8568**	0.8129	0.7897
FcNet(O)	**0.7915**	**0.7845**	0.8238	0.8343	0.8224	**0.8113**
FcNet(OS)	0.7880	0.7668	**0.8239**	0.8143	**0.8253**	0.8036
FcNet(OSM)	0.7447	0.7555	0.8220	0.8318	0.8217	0.7951

Table 3: Accuracy of pairwise temporal ordering using personalized prediction models (trained and tested on different videos from the same subject) for interval size 120. Here O indicates object features, S scene features, and M motion features

(Sec 4.3), and models personalized to a particular subject or environment (Sec 4.4).

4.1. Video snippet representation

There has been an increasing amount of work trying to understand the nature of ego-centric video. Different from videos taken from a third person point of view, egocentric videos record not only the inherent environment seen from the wearable device, but also the body or head motion of the wearer [1]. Quite a few previous works have used techniques such as, hand detection, object detection, or motion features such as optical flow to represent ego-centric videos [30, 9, 18, 33, 10], but how to represent first person videos effectively for temporal prediction tasks is an open question. Recently the use of features learned by deep networks have achieved success in many different vision tasks such as object recognition and detection [8, 27, 14], scene classification [42], or activity recognition [35]. In this work, we represent video snippets using learned deep features for objects, scenes, and motion.

Object representation: For each frame of a video snippet, we extract the 4096 dimension fc6 layer of the VGG model [3], pre-trained to recognize 1000 ImageNet [6] object categories. Then we apply max-pooling over a 10-frame window around the frame to implicitly capture some temporal information about objects within the snippet.

Scene representation: For each frame of a video snippet, to model scene/environment information for video snippets, we extract the 4096 dimension fc7 layer of the Caffe reference model [17], pre-trained on the scene-centric Places dataset [42]. Again, we apply max-pooling in a 10-frame window to capture temporal scene information.

Motion representation: Inspired by recent work on deep networks for activity recognition, we re-implement the Temporal Convnet approach of Simonyan and Zisserman [35]. Their method takes a two stream approach to activity recognition using both object and optical flow features. Our reimplementation of their method achieves an accuracy of 78% on the UCF-101 dataset compared to their reported result of 81%. From the optical flow portion of the Temporal Convnet, we extract the 4096 dimensional fc6 layer as our motion representation.

In our experiments, we evaluate the use of object features in isolation or combining object features with scene

or scene and motion features. For the combined features we simply concatenate features for classification. For our experiments we use video snippets of size 60 frames. For each snippet we uniformly sample 6 frames from the snippet, compute the above features and then do max-pooling over each feature dimension to get the final representation.

4.2. Prediction methods

We evaluate several methods for predicting pairwise temporal ordering. The first two are nearest neighbor based methods. The intuition behind these methods is that if two video snippets have similar appearance and motion then we can directly transfer temporal information from one video to the other. One challenge for nearest neighbor methods is that activities can be completed at different rates, therefore we experiment with temporal warping methods to align video snippets. Next we present a regression method that tries to directly predict the time of a video snippet. All three of these models attempt to estimate when a video snippet occurred within a larger activity. Given two snippets we can then predict their pairwise ordering based on their relative estimated times. Our final two methods are trained to directly predict pairwise ordering. Given two video snippets, A and B, we train an SVM and a fully-connected network to predict whether or not snippet A occurred temporally before B. For all of these methods we assume that we know what activity is occurring to focus our efforts on the task of temporal prediction. These methods could be incorporated into a broader system to estimate both activity recognition and temporal predictions.

NN Frac: We first represent the temporal information of all snippets as a real value computed as the temporal position of the snippet relative to the length of the entire action. For each query snippet, using one or more of our feature representations, we retrieve its nearest neighbor snippet from the training set and transfer the nearest neighbor's relative time to the query. Similarity is measured as cosine similarity.

NN DTW: We perform nearest neighbor prediction as before, but a priori first align all training videos for an activity temporally using Dynamic Time Warping (DTW) [34]. DTW is a dynamic programming algorithm that keeps track of the cost of the best path of the alignment. Here the cost function is defined as the Euclidean distance between each pair of video snippets.

LR: We train a Linear Regression model to estimate the temporal position of a video snippet, relative to the length of the entire activity. Inputs to the regressor are one or more of our feature representations described in Sec 4.1.

SVM: We train a linear SVM model directly for the pairwise prediction task. Input to the model are concatenated features from a pair of snippets, A and B. Output of the model is a binary prediction $\in \{1, -1\}$ where the model predicts 1 to indicate that A temporally occurs before B and -1 to indicate that A occurs after B. To train this model, we randomly sample pairs of snippets from activities with intervals between the snippets ranging from 60 to 300 frames. The learning parameter is set using cross-validation.

FcNet: Inspired by the metric network architecture introduced in [15], we train a three layer fully-connected network to predict pairwise ordering. The scenario is the same as the SVM method, that is, we model the task as a binary classification problem. The first two layers of this network have 512 units and use Rectified Linear Unit (ReLU) as the non-linear activation function. The last layer has 2 output units, estimating the probability that snippet A occurs before or after B. During training we apply mini-batch gradient decent and cross-entropy loss. We set the learning rate to be 0.001, dropout rate 0.5, momentum 0.9, and train for 30000 iterations. The hyper-parameters are decided using a validation set.

4.3. General Pairwise Ordering Prediction

We evaluate performance of our models on the general pairwise ordering task using a leave-one-out strategy, training on all subjects except one and then predicting on the held out subject. Fig 4 (left) shows accuracy averaged over activities and subjects for each prediction method, using a combination of all feature types. Similar to humans, the computational methods do not provide accurate predictions when the interval between snippets is too small, but as the interval increases performance improves. For the NN method applying DTW does not help a great deal, probably due to the high variance of the data. Future work could consider better alignment mechanisms. We also observe that the SVM and FcNet models trained to directly predict temporal ordering outperform the other methods significantly. Table 2 shows accuracy for each activity, classifier choice, and feature choice (for interval size 120). For some activities, optimal performance is achieved by combining all of the features while performance in others favored the combination of object and scene features.

4.4. Personalized Pairwise Ordering Prediction

We evaluate two types of personalized models: models personalized to a particular subject or to a particular location. First, we evaluate personalization where we use different video clips from a single subject for training and testing by applying the leave-one-out method. Since the amount of personalized data is quite limited, for the FcNet, to prevent overfitting, we fine-tune the general network using personalized data for another 25000 iterations (the hyper-parameters remain unchanged). Figure 4 (right) shows averaged results across activity, subject, and interval for models personalized to a subject. In these experiments, we see improved performance on the pairwise ordering task compared to general prediction models trained on other subjects, indicating that the personalized models are able to better adapt to a particular individual's habits and daily en-

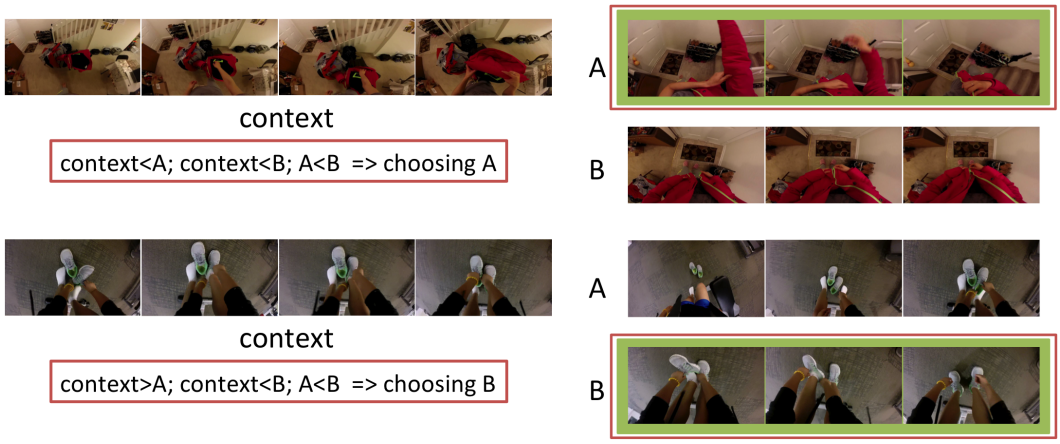


Figure 7: Visualization results of computer-based future prediction. Text within red borders are the pairwise ordering results generated by our method. Right shows algorithm proposed future prediction (red border) and ground truth (green border).

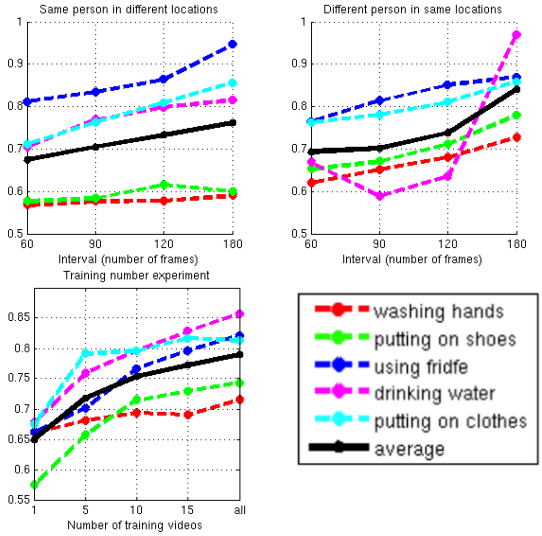


Figure 5: Additional personalization experiments for pairwise ordering. Top left shows performance for same person, different locations. Top right shows performance for different people, same location, and bottom left evaluates performance as training set size varies.

vironments. Table 3 shows personalized pairwise ordering accuracy for each activity type and feature combination for interval size of 120. Unlike the general prediction model, the models achieve best performance with the object representation based model, but for some activities incorporating additional features is helpful.

We provide several more experiments to further understand personalization. To evaluate generalization for individuals between locations, we train models for the single-subjects with different locations in training vs testing. To evaluate generalization between people in the same location, we train models for family-subjects with different family members in training vs. testing. And we also evaluate the effect of training set size. Due to the limited data,

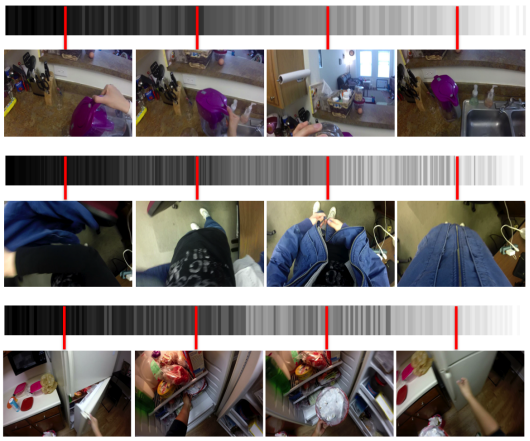


Figure 6: Inferring temporal information for an entire video sequence. Colorbar indicates the reordering of original time information (black=start, white=end).

for each experiment we apply the SVM method on object, scene, and motion features. (Figure 5 shows quantitative results). For the individual and location experiments, accuracies are still reasonable compared to the previous personalization experiment. For the training size experiment, we find that the amount of training data required for accurate prediction varies, with some activities benefiting from larger training sizes (e.g. "using fridge") and others achieving surprisingly good accuracy with only 5 samples (e.g. "putting on clothes").

Finally, as mentioned above, pairwise ordering can be used a backbone algorithm for inferring the temporal information of an entire video sequence. To demonstrate this potential, we reshuffle the video sequence of an activity and use our personalized regression model to predict a temporal value for each frame. Then we reorder the frames based on predicted time. Results are visualized in Fig 6 where the colorbar shows the reordering of original time

Activities	SVMg	SVMp	FcNetg	FcNetp	Human
Wash hands	0.6350	0.7550	0.6350	0.7900	0.7816
Put on shoes	0.7000	0.7250	0.7600	0.7700	0.8733
Use fridge	0.6100	0.7100	0.6600	0.7350	0.9284
Drink water	0.6500	0.7300	0.6350	0.7500	0.8717
Put on clothes	0.7100	0.8350	0.6950	0.8650	0.8866
Average	0.6630	0.7510	0.6770	0.7820	0.8686

Table 4: Future prediction task accuracy by computational methods and people, where 'g' indicates general model, and 'p' personalized model.

(ground truth) information (black=start in original video, white=end), with sampling of reordered keyframes below.

5. Future Prediction Task

Next we design a future prediction task where we are provided with a 3 second context video, C, of an activity and two video snippets, A and B, and we are asked to predict which will occur (soonest in time) after C. A is sampled from soon after the context video (randomly selected, but no more than 3 seconds in the future) and B is randomly sampled, either from further in the future or from the time period prior to C. A correct prediction will predict snippet A as happening next.

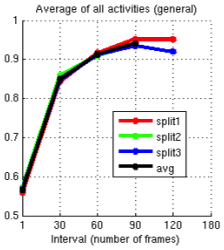
Computer prediction: Given an algorithm to predict pairwise orderings between snippets, it is straightforward to extend this algorithm to the future prediction task. We compute all pairwise orderings between A, B, and C, and then select the snippet that is most likely to follow after C in temporal order. For these experiments we use combined object, scene, and motion features (Fig 7 shows examples).

Human prediction: We also evaluate how well humans can make future predictions when provided with a longer context video. In particular, we show 3 AMT workers the context video plus the two video snippets and ask the worker to identify which will follow soonest in time after the video.

Table 4 shows accuracies of human and computer predictions. Personalized models outperform the general models significantly, achieving an average accuracy of 75%. Human performance on this task is also quite good (87%). We also want to understand how well algorithms can perform on the future prediction given snippets from two different videos. Here we first ask humans to perform the future prediction task and then select data with high inter-subject agreement. On this data, using the human predictions as ground truth, the general SVM and FcNet models achieve 66.22% and 66.99% accuracy respectively.

6. Additional experiments

Pairwise ordering on UCF101: For comparison, we also evaluate our pairwise ordering task on a subset of a wildly-used third person action recognition dataset, UCF101. We select 10 categories of action with reasonably long durations and non-repetative movements. Each category contains more than 100 video clips. In this experiment we evaluate our SVM method and keep other settings (snippet size, feature) the same as our previous experiments. We run 3 train/test splits provided by the UCF offical project webpage. Fig 8 (left) shows performance. We see that for third-person activity videos, our method can achieve even better performance than for first person videos.



Activities	Our method	FlowWords
wash hands	76.03%	76.86%
put on shoes	69.30%	72.81%
use fridge	79.67%	72.73%
drink water	57.52%	66.09%
put on clothes	80.37%	63.55%
average	72.58%	70.41%

Figure 8: Left is the pairwise ordering accuracy of subset of UCF101 dataset. Note that video clips in UCF101 are very short for some actions, the largest interval is only 90. Right is the forward/backward classification accuracy of our method and Flow-Words method in [29] testing on our dataset

Arrow of time: Finally, we use our pairwise ordering method as a backbone to evaluate the task proposed by [29] on our dataset. The goal of [29] is to tell whether a video is runing forward or backward. We implement their Flow-Words classification method which is based on a SIFT-like descriptor and linear SVM. For specific information and parameter settings, please refer to [29]. Predicting the temporal direction of video clips can be solved by our pairwise ordering predictions. Our method achieves comparable performance on this task. For each testing video clip, we sample all its snippet pairs with interval 90 and 120 along the video and apply our general SVM model to classify the ordering, then we do majority vote to decide the flow direction of the video. Fig 8 (right) shows the average accuracy for Flow-Words and our method.

7. Conclusions

We have introduced two tasks for evaluating temporal understanding of ego-centric videos of everyday activities: pairwise ordering and future prediction. We have evaluated both human performance on these tasks and computational models under general and personalized training scenarios. We find that models trained directly on the pairwise ordering task outperform models trained to predict the time at which a video snippet occurred. We also find that personalized models significantly outperform general models, suggesting that to build an accurate predictor for an individual, we should capture data specific to that person.

Acknowledgements We thank David Forsyth for ideas and discussions related to the prediction problem and Vicente Ordonez for useful discussions. We also thank Eunbyung Park for help training the Temporal Convnet. This research is supported in part by NSF grant #1445409.

References

[1] O. Aghazadeh, J. Sullivan, and S. Carlsson. Novelty detection from an ego-centric perspective. In *CVPR*, 2011.

[2] A. Betancourt, P. Morerio, C. S. Regazzoni, and M. Rauterberg. The evolution of first person vision methods: A survey. *CoRR*, 2014.

[3] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman. Return of the devil in the details: Delving deep into convolutional nets. In *BMVC*, 2014.

[4] T. Dekel, Y. Moses, and S. Avidan. Space-time tradeoffs in photo sequencing. In *ICCV*, 2013.

[5] T. Dekel (Basha), Y. Moses, and S. Avidan. Photo sequencing. *IJCV*, 2014.

[6] 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.

[7] P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. In *VS-PETS*, 2005.

[8] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. *CoRR*, 2013.

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

[10] A. Fathi, J. Hodgins, and J. Rehg. Social interactions: A first-person perspective. In *CVPR*, 2012.

[11] A. Fathi, Y. Li, and J. Rehg. Learning to recognize daily actions using gaze. In *ECCV*. 2012.

[12] A. Fathi, X. Ren, and J. Rehg. Learning to recognize objects in egocentric activities. In *CVPR*, 2011.

[13] B. Fernando, E. Gavves, J. Oramas, A. Ghodrati, and T. Tuytelaars. Modeling video evolution for action recognition. In *CVPR*, 2015.

[14] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014.

[15] X. Han, T. Leun, Y. Jia, R. Sukthankar, and A. C. Berg. Matchnet: Unifying feature and metric learning for patch-based matching. In *CVPR*, 2015.

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

[17] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. B. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *CoRR*, 2014.

[18] K. Kitani, T. Okabe, Y. Sato, and A. Sugimoto. Fast unsupervised ego-action learning for first-person sports videos. In *CVPR*, 2011.

[19] K. Kitani, B. Ziebart, J. Bagnell, and M. Hebert. Activity forecasting. In *ECCV*. 2012.

[20] A. Kläser, M. Marszałek, and C. Schmid. A spatio-temporal descriptor based on 3d-gradients. In *BMVC*, 2008.

[21] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*. 2012.

[22] I. Laptev and T. Lindeberg. Space-time interest points. In *Computer Vision*, 2003.

[23] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *CVPR*, 2008.

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

[25] Y. Li, A. Fathi, and J. Rehg. Learning to predict gaze in egocentric video. In *ICCV*, 2013.

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

[27] M. Oquab, L. Bottou, I. Laptev, and J. Sivic. Learning and transferring mid-level image representations using convolutional neural networks. In *CVPR*, 2014.

[28] M. Pei, Y. Jia, and S.-C. Zhu. Parsing video events with goal inference and intent prediction. In *ICCV*, 2011.

[29] L. C. Pickup, Z. Pan, D. Wei, Y. Shih, C. Zhang, A. Zisserman, B. Schölkopf, and W. T. Freeman. Seeing the arrow of time. In *CVPR*, 2014.

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

[31] X. Ren and C. Gu. Figure-ground segmentation improves handled object recognition in egocentric video. In *CVPR*, 2010.

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

[33] M. Ryoo and L. Matthies. First-person activity recognition: What are they doing to me? In *CVPR*, 2013.

[34] H. Sakoe and S. Chiba. Dynamic programming algorithm optimization for spoken word recognition. *ICASSP*, 1978.

[35] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. *CoRR*, 2014.

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

[37] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu. Action Recognition by Dense Trajectories. In *CVPR*, 2011.

[38] H. Wang, M. M. Ullah, A. Klaser, I. Laptev, and C. Schmid. Evaluation of local spatio-temporal features for action recognition. In *BMVC*, 2009.

[39] K. Yamaguchi, A. Berg, L. Ortiz, and T. Berg. Who are you with and where are you going? In *CVPR*, 2011.

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

[41] N. Zhang, M. Paluri, M. Ranzato, T. Darrell, and L. Bourdev. Panda: Pose aligned networks for deep attribute modeling. In *CVPR*, 2014.

[42] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva. Learning Deep Features for Scene Recognition using Places Database. *NIPS*, 2014.