Recognition of Ongoing Complex Activities by Sequence Prediction over a Hierarchical Label Space

Wenbin Li Mario Fritz
Max Planck Institute for Informatics
wenbinli,mfritz@mpi-inf.mpg.de

Abstract

Human activity recognition from full video sequence has been extensively studied. Recently, there has been increasing interest in early recognition or recognition from partial observation. However, from a small fraction of the video, it might be demanding if not even impossible to make a fine grained prediction of the activity that is taking place. Therefore, we propose the first method to predict ongoing activities over a hierarchical label space. We approach this task as a sequence prediction problem in a recurrent neural network where we predict over a hierarchical label space of activities. Our model learns to realize accuracy-specificity trade-offs over time by starting with coarse labels and proceeding to more fine grained recognition as more evidence becomes available in order to meet a prescribed target accuracy. In order to study this task we have collected a large video dataset of complex activities with long duration. The activities are annotated in a hierarchical label space from coarse to fine. By directly training a sequence predictor over the hierarchical label space, our method outperforms several baselines including prior work on accuracy specificity tradeoffs originally developed for object recognition.

1. Introduction

Strong progress has been achieved in the area of human activity recognition over the last decade ranging from coarse actions [34] to fine-grained activities [30]. The majority of these techniques focus on what has happened. However, many application in surveillance system, crime prevention, assistive technology for daily living, human computer interaction interface, human robot interaction would like to infer which activity is currently going on in order to enable a system to behave proactive or context-aware.

Current system that address early recognition and also activity recognition in general have a relatively short temporal reach. Typically the investigated activities are on the order of a minute – with a few exceptions like cooking activities. This limits the systems in the way they can act and assist to a more immediate, anticipatory response. In order to increase the temporal reach of our systems we have to study a richer set of longer term activities.

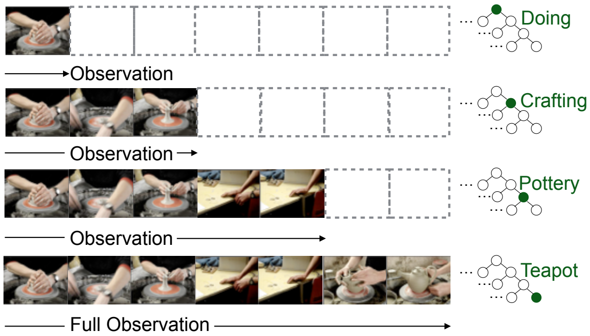


Figure 1: An overview of our approach to inferring goal during the recognition process of a complex activity by predicting with semantic abstraction.

While this goal is desirable, it raises the question how much can be inferred about a complex activity from a very short observation? The natural concern is that in many cases the information might be quite limited until a more substantial fraction is being observed. Therefore we argue that recognition of complex activities from partial observation has to go hand in hand with a mechanism to predict at different semantic granularity in order to deliver a useful prediction at anytime. In this sense, we have to bring coarse and fine-grained recognition together in a single approach.

In order to study this challenging problem, we propose the first dataset of complex activities that have a substantial time span and that are organized in a semantic hierarchy. We study how prediction can be performed from partial observation of ongoing activities in this setting by employing of a semantic back-off strategy. We introduce a new representation that models the uncertainty over different time frames across different levels of the semantic hierarchy. More specifically, we propose a recurrent neural network formulation that learns to directly predict on this hierarchical label space from coarse to fine in order to achieve an accuracy specificity trade off. A performance measure is proposed that evaluates the gain in specificity over time as more observations become available. Our new method outperforms several baselines as well as techniques that we adopt from the literature on accuracy specificity tradeoffs for object recognition.

2. Related Work

Early computer vision research in recognizing human motion and activities can be date back in early 1990's [1]. Researchers have explored various approaches to tackle the task and excellent surveys are available [25, 28, 1, 40]. In this section, we first review related datasets for activity recognition and then discuss some related approaches for analysis from partial observation.

Activity Datasets There is a large number of datasets proposed for activity recognition or detection with various levels of complexity. Early efforts to construct such datasets include the KTH dataset [34] and the Weizmann dataset [3] which feature simple activities like walking and jumping. More complex datasets are introduced later with various backgrounds, slightly longer duration and more importantly, with additional interaction of objects or peoples. Typical examples are answering phone (human-object interaction) in the Hollywood2 [23], golf swinging (human-object interaction) in the UCF Sports [29] and hand shaking (human-human interaction) in the UT-Interaction [33]. However, activities covered in these datasets usually consist only very few types of (in most cases only one) interactions, hence their structure are still relatively simple and individual video's length is rather short. In contrast to this, there are also long-duration datasets collected from surveillance video like VIRAT [26] or daily living recordings like TUM Kitchen [38], CMU-MMAC [8], URDAL [24], TRECVID [27], ego-centric datasets [11, 36], and MPII Cooking [30]. Videos from these datasets show complex activities that comprise multiple interactions of different type. Considering the example of making a dish in the cooking dataset [30], one needs to interact with different tools and ingredients to complete a recipe.

One limitation about current long-duration datasets is that they are mostly used for recognition of elementary activities within the long sequence instead of the recognizing the overall process. In addition, they are limited to a particular domain and do not provide a hierarchical label space. Our dataset exactly aims to fill this gap by providing dataset of complex activities across multiple domains that are organized in a semantic label hierarchy. A related dataset was built in [10] where activities are grouped with respect to social interactions and places where the activity usually takes places. There are several recent works collecting videos from YouTube [22, 19, 35]. We also collect our dataset from web sources, as it is a practical way to collect a sizable dataset of realistic videos.

Dataset	#Class	Avg. #Frames per Seq.
KTH	6	91
UT interaction	6	84
UCF sports	9	58
HMDB51	51	93
MPI cooking	65	157
Youtube action	11	163
Olympics sports	16	234
Hollywood2	12	285
VIRAT	12	357
KSCGR	8	786
Our dataset	48	8.7K

Table 1: Action/activity datasets.

Analysis from Partial Observation There are a few recent works focusing on analysis from partial observation. Early recognition [32] aims to recognize activity from the forepart of videos. The authors use accumulated histogram with respect to the observed frames to represent the activity of interest, and performs sequence matching with templates averaged from training dataset. Following the similar idea of sequence matching, [5] relaxes the location of observed part and addresses recognition from partial observation and applies sparse coding for better representation of the matching likelihood. A slightly different line of work like [13] focuses on early detection, deciding the temporal location of the activity, yet as shown in [4] the algorithm does not work so well in practical recognition task. A more recent work [20] builds the early activity prediction on the human detection and uses the resulted tracks to form a hierarchical representation, yet in real world scenario, the detection and tracking are usually expensive for complex scene involving multiple people and can often be unreliable, not mention in situations like ego-centric video where it is impossible to obtain a detection result for the person doing the activity.

Beyond predicting a label for the sequence, [17] poses a more challenging task to predict activity where a MDP is used for the distribution over all possible human navigation trajectories. The walking path addressed in [17] is still relatively simple, hence [18] further explores indoor activities which involves more complex interactions between human and object. They represent the distribution over incoming states with a set of particles sampled from an augmented CRF. The activities exploited in [18], like taking medicine, already meets our definition of complexities, yet it only applies to a very limited number of objects in very simple scenes under controlled lab conditions. It is not clear how it scale up to real world application and longer time scales.

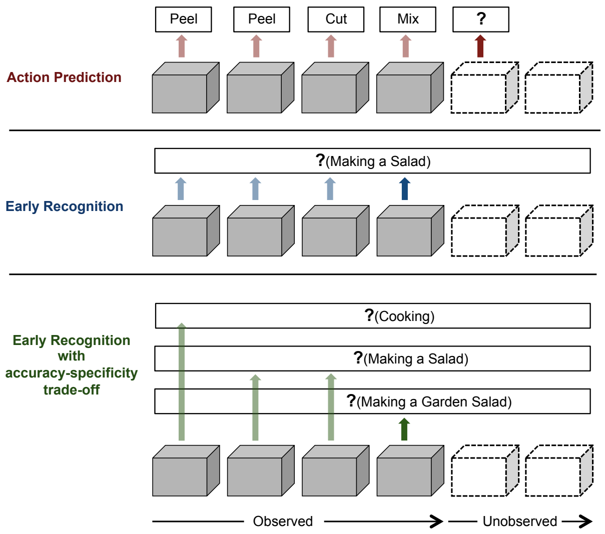


Figure 2: Example to show the difference between action prediction and early recognition. Given a streaming video sequence, (top) action prediction aims to give a label to a incoming local temporal segment (mix or cut); (middle) standard early recognition gives a label for the global video sequence (make a salad); (bottom) our proposed framework to predict in the hierarchy with accuracy-specificity trade-off. The solid arrows mark when the decision is made.

Prediction vs. Early Recognition Here we want to distinguish two different types of problems addressed in related literature, i.e. prediction for unseen action/activity or early recognition on the current observation. This is illustrated in Figure 2 with the example of making a salad which involves multiple activities including peeling, cutting and mixing. As one watches the activity moving on, prediction is to forecast what is the next move after mixing whereas early recognition is to tell the main theme of the activity as cooking regardless of only partial observation of the whole process. [17, 18] are examples of prediction and [32, 5, 20] are for early recognition.

Indeed predicting real world activity is a very difficult task, it generally requires various components including but not limited to pose estimation, object detection and temporal segmentation where each task along is challenging already in real world application not to mention it is nontrivial to combine them for a reasonable solution. Our work focuses on early recognition which due to the long temporal duration of our investigated activities, also extend to the future.

Activity Recognition with Recurrent Neural Networks [2] applies a different variant of RNN to action recognition task. Our work differs from it in that we focus on the early recognition and prediction over a hierarchical label space with a learned accuracy-specificity tradeoff. Our results show the joint training of the sequence model w.r.t. to the accuracy specificity tradeoff is key to our performance improvements.

3. Method

In this section, we first discuss different ways of video representation and formulate the task of early recognition in video sequence. Our goal is to predict at any point in the sequence over a hierarchical label space in order to realize accuracy-specificity trade-offs. In order to have a temporal integration of information over time and learn classifier and the hierarchical tradeoff jointly, we employ recurrent neural networks for modeling videos and extend it to labeling in a hierarchy, and learning accuracy-specificity trade-off in early activity recognition in videos.

3.1. Video Representation

Given a specific descriptor x, a video sequence can then be represented as $[x_1, x_2, ..., x_T]$, where T denotes the frame index. There are several ways to obtain a compact representation for sequences as shown in Figure 3: (1) pooling over the whole sequence; (2) partition the video into separated temporal segments, and combine the representation from the segments into a temporal sequence representation $[x_{1:t}, x_{t+1:2t}, ...x_{T-t+1:T}]$, where t is the number of frames for each segment. This approach has been applied in [4, 37]; (3) sample clips (also temporal segments) from the video to represent the video, each clip is trained as a single instance in the video class, i.e. $[x_{t_1:t_1+t}] \cup [x_{t_2:t_2+t}] \cup ...$, where $t_1, t_2, ...$ are the time stamps when the clips are sampled, and final prediction on the video is based on the majority vote from the sampled clips. This approach is applied in the large-scale video recognition work [16].

We use improved dense trajectory feature [39] as the basic feature to encode spatio-temporal information across frames in the videos with a code book of size 4000 obtained via k-means clustering. It uses a dense representation of extracted trajectories and combines with trajectory-aligned features, including HOG [6], HOF [21], motion boundary histograms (MBF) [7]. The descriptor itself has been shown to achieved the state-of-art performances on several public datasets.

3.2. Early Recognition

Given a video sequence $[x_1, x_2, ..., x_T]$, where T is the last frame index of the video. Early recognition generally refers to the task of classification:

$$y = f([x_1, x_2, ..., x_{\widetilde{t}}]) \ \ ,\widetilde{t} \leq T$$

The standard full video classification can be seen as a special case of this formulation where $\widetilde{t} = T$. It is impor-

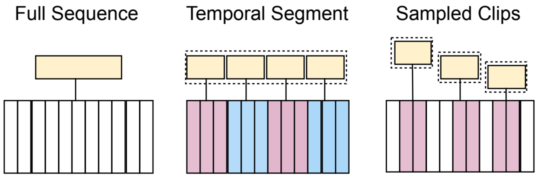


Figure 3: Different ways to represent individual video sequence: full sequence (left), use one representation for the full sequence; temporal segment (middle), partition full sequence into temporal segments and combine representation for individual segments into one representation; sampled clips (right), sample video clips from the sequence to form several representations.

tant to point out the difference between early recognition of activity [32] and human action prediction/anticipation [17, 18, 20] whereas the former is essentially a sequence classification problem that maps sequence in a single label

$$X : [x_1, x_2, ..., x_T] \mapsto y$$

and the latter is a temporal classification problem that maps a input sequence into a target sequence.

$$X : [x_1, x_2, ..., x_T] \mapsto [y_1, y_2, ..., y_L]$$

A specific example is shown in Figure 2 for ongoing video stream of making salad. Assume we already observe the person in video peeled the cucumber, cut it into slice, mix it with other ingredients, early recognition generally aims to output a global label for the sequence based on the available observation, in this case, the label should be ‘making salad’ while the action prediction tries to predict the specific local action label for the incoming video frames, in this case, the correct label would be ‘put the salad in plate’.

3.3. Recurrent Neural Networks

A recurrent neural network (RNN) is a class of artificial neural network that allows connections between units to form a directed cycle. We consider an architecture with one self-connected hidden layer, which can be unrolled in time as shown in Figure 4. One notable merit for RNN is that the recurrent connections allow a ‘memory’ of previous inputs to persist in the network’s internal state which can then be used to influence the network output [12]. This characteristic makes it suitable for sequence analysis, especially in our case, with long duration and complex video sequences.

Given a video sequence composed of temporal segments $x_1, x_2, ..., x_T$, each in R^d, the network computes a sequence of hidden states $h_1, h_2, ..., h_T$, each in R^m and predictions $y_1, y_2, ..., y_T$, each in R^k. The hidden unit integrates the information from the arrived observation and those propagated from previous blocks of the networks:

$$\widetilde{x_i} = W_{xh}x_i + W_{hh}h_{i-1} + b_h$$
$$h_i = \tanh(\widetilde{x_i})$$

Each prediction unit represents the class label up to the current observation and is activated by the hidden unit via a softmax function:

$$\widetilde{h_i} = W_{hy}h_i + b_y$$
$$y_i = \text{softmax}(\widetilde{h_i})$$

where W_{xh}, W_{hh}, W_{hy} are the weight matrices and b_h, b_y are the biases. The training is done by BackPropagation Through Time (BPTT) [31] and the parameter is learned by conjugate gradient descent method.

3.4. Early Recognition in a Semantic Hierarchy

Deng et al. [9] first introduce the concept of optimizing an accuracy-specificity tradeoff for hierarchical image classification with the DARTS algorithm. Here we briefly recap the formulation of the concept and discuss how we extend it to our settings.

Optimizing Accuracy-Specificity Trade-Offs The key idea behind the accuracy-specificity trade-off is to make cost-sensitive prediction over the hierarchy, where predictions at upper level (fine-grained level categories) of the hierarchy get penalized more than those at lower level (coarse level categories). By optimizing over both the specificity and accuracy an optimal tradeoff is found. More formally, given a classifier $f : X \mapsto Y$, with accuracy $\Phi(f)$ defined as,

$$\Phi(f) = E[f(X) \in \pi(Y)]$$

where $\pi(Y)$ is the set of all possible correct predictions, $[P]$ is the Iverson bracket, i.e., 1 if P is true and 0 otherwise. The preference for specific class labels over general class labels at each node v, i.e. the specificity is encoded with information gain (decrease in entropy) by

$$r_v = log_2|Y| - log_2 \sum_{y \in Y} [v \in \pi(y)]$$

The total reward for the classifier f is hence defined as

$$R(f) = E(r_f(X)[f(X) \in \pi(Y)])$$

The optimal trade-off between accuracy and specificity is then formulated as maximizing the reward given an accuracy guarantee $1 - \epsilon \in (0, 1]$:

$$\begin{aligned} \underset{f}{\text{minimize}} \quad & R(f) \\ \text{subject to} \quad & \Phi(f) \geq 1 - \epsilon \end{aligned}$$

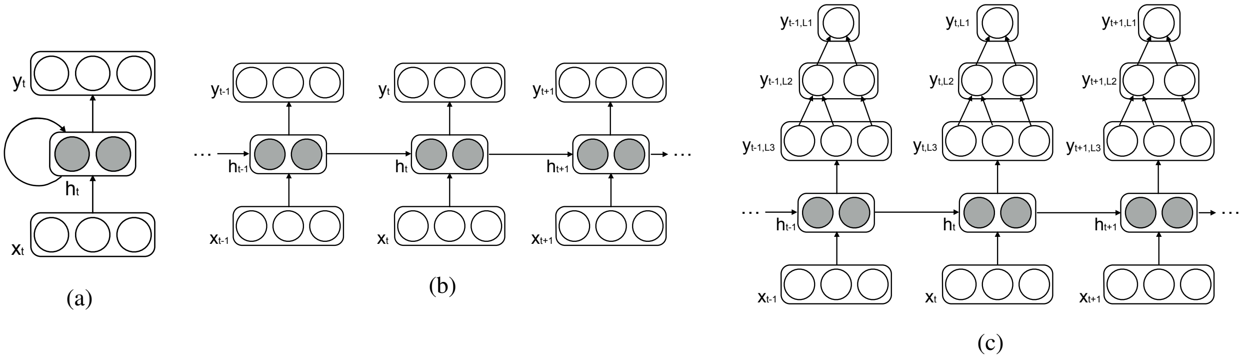


Figure 4: Graphical model for recurrent neural networks. (a) The recurrent neural networks with a single, self-connected hidden layer. (b) The unrolled model with respect to discrete time steps from (a). (c) The recurrent network with structural output over the hierarchy.

Accuracy-specificity Trade-off Over Time Motivated by ideas from incremental feature computation and anytime recognition [14, 15], we extend this concept with a temporal dimension to early recognition and model the decision process when analyzing an ongoing video stream. At each time step, we have to predict a label in the hierarchy and hereby trade-off between accuracy and specificity.

$$R(f,t) = E(r_f(X_t)[f(X_t) \in \pi(Y)])$$

The intuition behind this is that when only observing a small portion of the video, we have little evidence to accurately predict at a fine-grained level but may still be able to give a sensible coarse-level class label. By considering the total cost of possible wrong prediction at fine-grained level and probably correct prediction at coarse-level, together with our preference to predict at different levels, we are likely to give prediction at coarse-level given little observed data. When observing more and more parts of the video, we become more certain about our prediction at the fine-grained level, and by the same mechanism, we start to predict at a more fine-grained level. Figure 2 shows a concrete example. By summing up the term over time T,

$$\sum_t^T R(f,t) = \sum_t^T E(r_f(X_t)[f(X_t) \in \pi(Y)])$$

we can evaluate the efficiency for accuracy-specificity from a classifier f.

Structured Output RNN for Predictions Over Hierarchical Label Spaces We propose an RNN optimizing the objective from above by predicting over a structured output space – our hierarchy H. We denote labels at top-layer, middle-layer and bottom layer in the hierarchy as Y_1, Y_2, Y_3 (coarse to fine). As shown in Figure 4, our RNN model directly predicts an output layer $y_{3,i}$ representing the posterior probabilities over the fine grained labels in the bottom layer.

$$y_{3,i} = p(Y_3 = i|x), i = 1, ..., K_3$$

where K_3 is the number of classes within the layer. On top of this layer, we introduce an additional layer to represent the middle layer, where the connections between these two layers are defined according to the hierarchy H, i.e. if class i in bottom layer belongs to class j in middle layer, node i and j are connected or $(i,j) \in H$. Accordingly, the middle layer activations are defined as follows:

$$\begin{aligned} y_{2,j} &= p(Y_2 = j|x), j = 1, ..., K_2 \\ &= \sum_i p(Y_2 = j|Y_3 = i)p(Y_3 = i|x) \end{aligned}$$

Similarly we define another layer above this layer to represent top layer prediction (coarse labels):

$$\begin{aligned} y_{1,k} &= p(Y_1 = k|x), k = 1, ..., K_1 \\ &= \sum_j p(Y_1 = k|Y_2 = j)p(Y_2 = j|x) \\ &= \sum_j \sum_i p(Y_1 = k|Y_2 = j)p(Y_2 = j|Y_3 = i)p(Y_3 = i|x) \end{aligned}$$

The complete model is shown in Figure 4. Based on the this prediction over the hierarchy we define a structured loss in order to optimize for the desired accuracy specificity trade-off:

$$\begin{aligned} Loss(\theta, D) = -\sum_i &\log p(Y_1 = y_1^{(i)}|x^{(i)}, \theta) \\ &+ \log p(Y_2 = y_2^{(i)}|x^{(i)}, \theta) \\ &+ \log p(Y_3 = y_3^{(i)}|x^{(i)}, \theta) \end{aligned}$$

where D denotes the training set$\{X, Y_1, Y_2, Y_3\}$, and i indexes the i-th data instance in D.

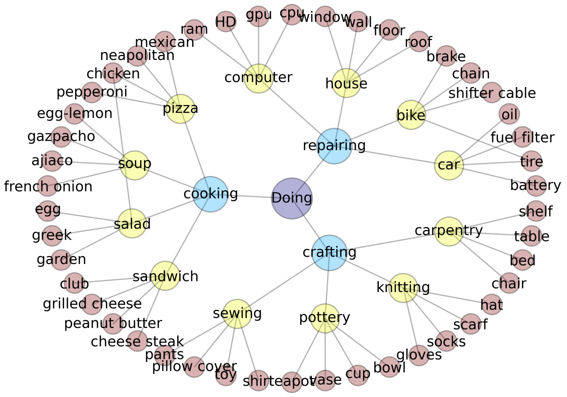


Figure 5: 3-layer hierarchy defined in our dataset.

4. Experiment

We first present our new dataset of complex activities with a hierarchical label space and afterwards perform a quantitative comparison of our proposed method and compare to several baselines.

4.1. Datasets

We explore various complex activities composed of multiple interactions. Recording videos for a large set of diverse classes is difficult, considering we need to find experts in different fields to perform the activities and capture multiple videos for a single class. In addition, this would eliminate the challenge of different capture devices. Hence, we build our dataset on videos from web to create our dataset. In the following part, we briefly discuss how we collect videos, pre-process the data and tackle the associated challenges of building such a dataset.

Data Collection We begin by defining a 3-layer semantic hierarchy (we count the root node of "Doing something" as layer 0) with 3 nodes in the first layer (cooking, crafting, repairing), then for each node, we select 4 specific derived categories as nodes to form the following layer, and for each node in the middle layer, we select 4 more specific classes as leaf nodes to form the bottom layer. For example, we include making "pizza", "soup", "salad" and "sandwich" as the second layer for "cooking", and for "salad", we consider making 4 different kinds of salads, namely "egg salad", "garden salad", "chicken salad" and "greek salad". Overall, we obtain a tree for complex activities with total $3 \times 4 \times 4 = 48$ activity classes as leaf nodes. The full hierarchy is shown in Figure 5. 10 videos are collected for each leave node from YouTube and eHow, which sum up to 480 video clips with total length of more than 41 hours or more than 4.18 million frames. The dataset is available online[1]. Sample frames of the videos are shown in Figure 6. We use half number of videos for training and the rest for testing.

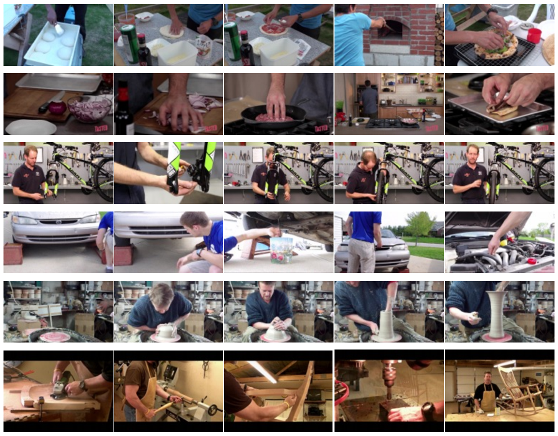


Figure 6: Some sample frames from our dataset.From top to bottom: make neapolitan pizza, make cheese steak sandwich, repair bike brake, change car oil, make vase, build chair.

These videos from the web differ from the ones recorded from lab: while the latter record the whole process for each activity with good controlled conditions, the former are often edited (adding head leader, tail leader, titles, flashback, etc) from the uploaders under various conditions (different point of view, shooting skills, etc). Hence such data is very noisy and exposes many of the aforementioned challenges of realistic videos. We rescale video into 360p for further processing (640×360).

4.2. Full Video Recognition

First we consider a setting where we classify full video sequences. This is particularly interesting since our collected videos are significant more complex than previous datasets on both the temporal scale and internal structure. This provides a relative measure of difficulty for activity recognition on our database. We train and predict on full sequences with a SVM classifier. While a χ^2 kernel is usually applied to integrate different descriptors in a multichannel fashion as in [39], we find out that a linear kernel gets slightly better results on our dataset. Therefore we use the linear SVM for all our experiments. As can be seen from Table 2, the performance reaches 25.7% at layer-3 (fine-grained), which suggests that we have established indeed a very difficult task at this detailed level.

An alternative to training on full sequence is to use sampled clips to represent the whole video [16]. Accordingly, we randomly sample 20% of each video in the training set for training, and test on the full video sequences in the test set. Note here we predict directly on the entire video sequence instead of the average on the prediction of sampled clips from each test sequence. As shown in 2, this approach is also valid on our dataset and is only slightly worse than training on full video sequence.

[1] http://www.mpii.de/ongoing-activity

Layer	Clips-training(%)	Full-training(%)
bottom-layer	25.3	25.7
middle-layer	52.3	59.1
top-layer	76.4	78.1

Table 2: Classification results on full video sequence

4.3. Recognition from Partial Observation

We proceed by examining the case where the video is only partially observed.We simulate two types of video segments to represent incremental arrival of video frames, (i) frames from the beginning with different observation ratios $[10\%, 20\%, ..., 100\%]$ where we explore different strategies for early recognition (ii) continuous frames taking up 10% of the full observation starting at different temporal location. As a result a video is represented as a temporal sequence of these frame segments $[0-10\%, 10-20\%, ..., 90-100\%,]$. This setting simulates the practical setting where the observer starts from arbitrary position and perceive a part of videos and wants to infer the current activity class. We refer to this as *online recognition*.

Early Recognition We evaluate the following strategies for early recognition: (1) train single model on full video and predict at different observation ratios; (2) train with augmented data i.e the combination of full video and video segments of different observation ratios and make predictions; (3) train on sampled video segments and test on partial observation, this is inspired by the result from our full-video recognition experiment and we would like to investigate how models trained on sampled clips can discriminate activity classes based on partial observation. We use the same pipeline as in the full video recognition and also test on differ layers in the label hierarchy. The results are shown in Figure 7. As we see from the plots, while the training on sampled clips gets slightly worse results than the other two settings, i.e. training on full sequence and on both full sequence and augmented data is still feasible. Comparing training on full sequence with and without the augment dataset, we observe improvement on upper two layer. At the lower-level, it helps when the observation ratio is below 60% but degrades the performance slightly afterwards.

In addition to early recognition setting, we also evaluate the accuracy-specificity trade-off over time as shown in Figure 8. We compare to our adaption of the DARTS algorithm as described above as a baseline. As we can see from the expected information gain at fixed target accuracy, training with full sequence and augment data (red curve) most of the time achieve the best reward over the other two, i.e. training on full sequence (green curve) and training on clips (blue curve). To help better understand the concept, Figure 8 also shows examples of prediction a distribution over the hierarchy and observation ratios at several target accuracy. The proportion of predictions at lower level grows with time, which means the classifier gets more certain about the activity class over time. When specifying a lower target accuracy, there are more predictions at lower levels in the hierarchy, with higher target accuracy the other way around. In order to reach the target accuracy, the prediction has to move to higher layer that has better confidence.

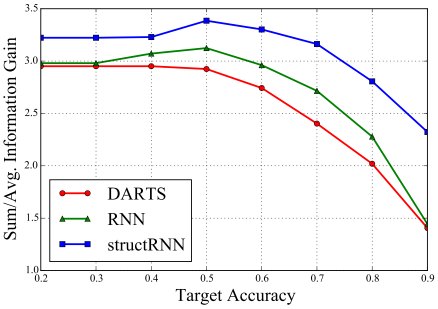


Figure 9: Results for online recognition.

Online Recognition We evaluate online recognition using our proposed RNN model that performs a structured prediction over the label hierarchy (structRNN) and compare it to the DARTS algorithm and a plain RNN. For RNN and struct RNN, we use 50 units for the hidden layer and use a L_2 regularizer. Parameters were selected based on the validation set. We perform dimensionality reduction by using the decision value from linear classifiers trained over different layers in order to reduce the raw feature vector of $20K$ dimension. In a preliminary study we have observed that this generally gives better performance on the expected reward for accuracy-specificity trade-off. The final results are show in Figure 9. Compared to the baseline DARTS algorithm, the RNN achieves better performance. This is due to the connectivity between the hidden units that improve the propagation of information along time. The structRNN further improves the RNN performance as we enforce the structural loss with respect to the hierarchy. In addition, we note that our structRNN even outperforms the previously investigated early recognition settings. Figure 10 shows some example predictions for videos.

5. Conclusion

In this paper, we proposed a new challenging dataset with hierarchical labels to study the recognition of long-duration, complex activities and temporal accuracy-specificity trade-offs. We propose a new method based on recurrent neural networks that learns to predicts over this hierarchy and realizes accuracy specificity tradeoffs. Our method outperforms several baselines on this new challenge including an adaptation of hierarchical prediction from object recognition.

Acknowledgements This work was partially supported by the DFG collaborative research center SFB 1223.

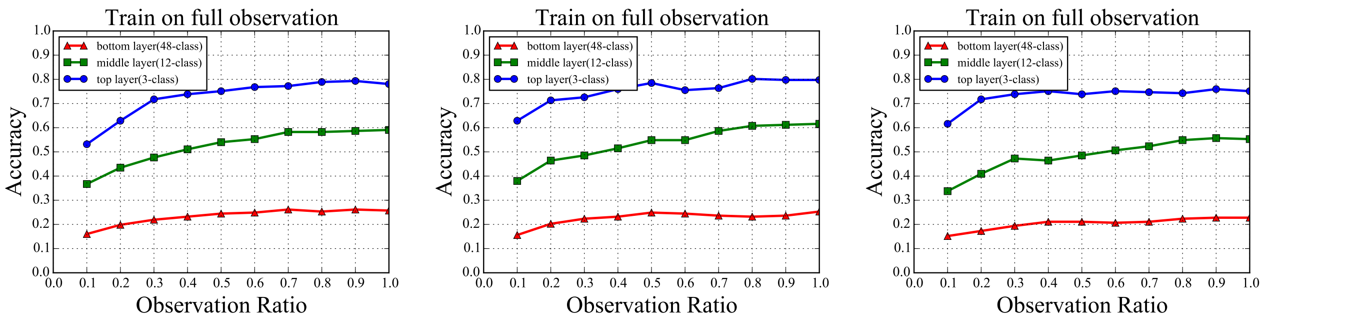


Figure 7: (Left) train SVM model from full video and predict at different observation ratios; (middle) train model from full video sequences and augmented dataset; (right) train model from sampled clips.

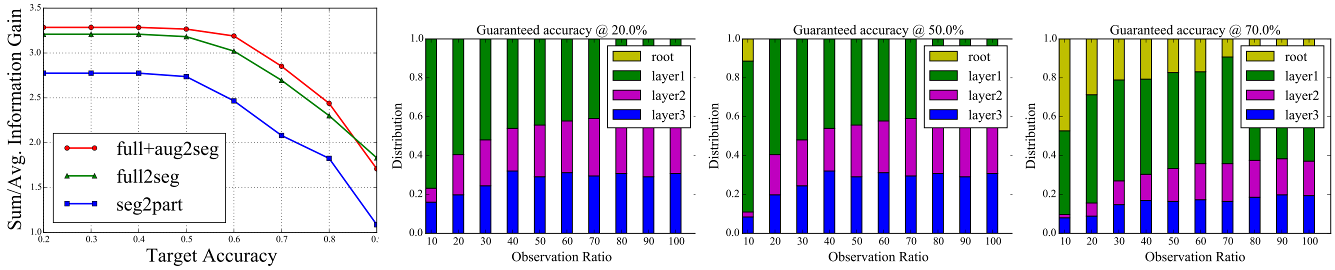
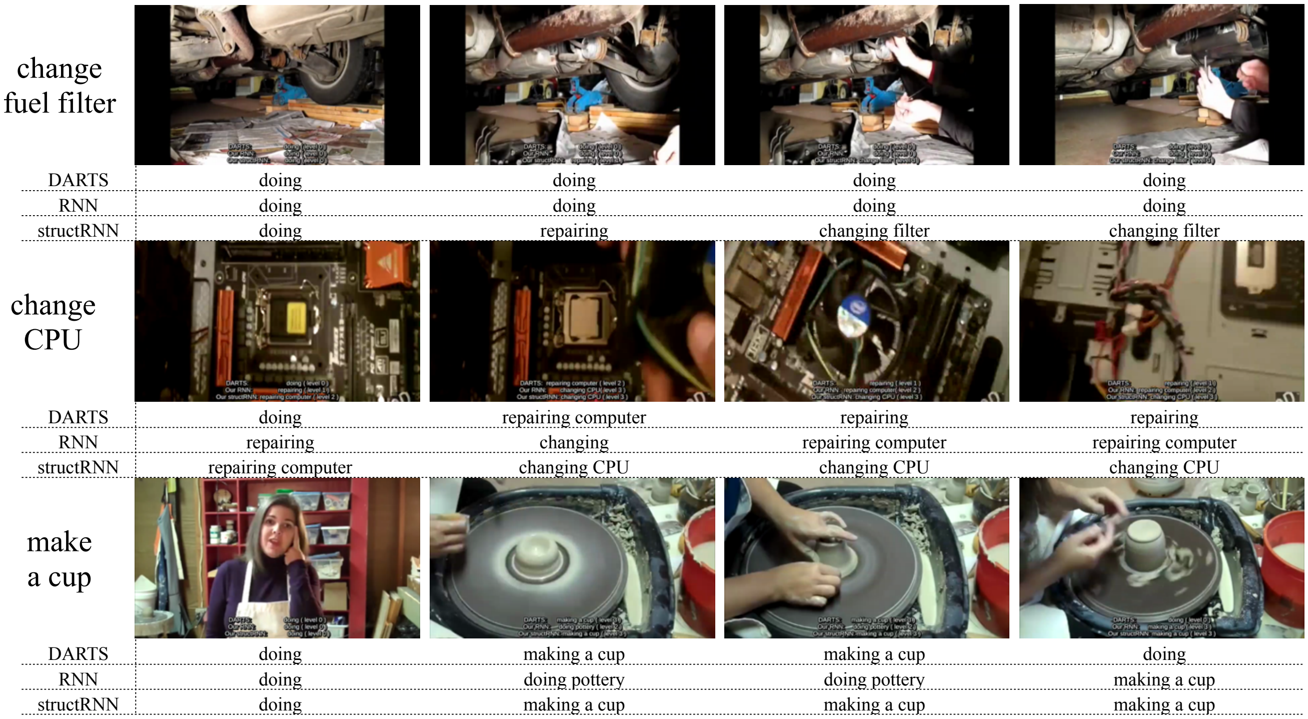


Figure 8: (Left)the expected information over time given specific accuracy.(right) Example of prediction distribution over time based on the optimization over accuracy-specificity trade-off for train on full and augment segments.



change fuel filter	DARTS	doing	doing	doing	doing
	RNN	doing	doing	doing	doing
	structRNN	doing	repairing	changing filter	changing filter
change CPU	DARTS	doing	repairing computer	repairing	repairing
	RNN	repairing	changing	repairing computer	repairing computer
	structRNN	repairing computer	changing CPU	changing CPU	changing CPU
make a cup	DARTS	doing	making a cup	making a cup	doing
	RNN	doing	doing pottery	doing pottery	making a cup
	structRNN	doing	making a cup	making a cup	making a cup

Figure 10: Example prediction over three activities in the video, from top to bottom: change car filter, change a CPU and make a cup.

References

[1] J. Aggarwal and M. S. Ryoo. Human activity analysis: A review. *ACM Computing Surveys (CSUR)*, 2011.

[2] M. Baccouche, F. Mamalet, C. Wolf, C. Garcia, and A. Baskurt. Sequential deep learning for human action recognition. In *Human Behavior Understanding*. Springer, 2011.

[3] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. In *ICCV*, 2005.

[4] L. Cao, Y. Mu, A. Natsev, S.-F. Chang, G. Hua, and J. R. Smith. Scene aligned pooling for complex video recognition. In *ECCV 2012*. 2012.

[5] Y. Cao, D. Barrett, A. Barbu, S. Narayanaswamy, H. Yu, A. Michaux, Y. Lin, S. Dickinson, J. M. Siskind, and S. Wang. Recognizing human activities from partially observed videos. In *CVPR*, 2013.

[6] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005.

[7] N. Dalal, B. Triggs, and C. Schmid. Human detection using oriented histograms of flow and appearance. In *ECCV*. 2006.

[8] F. De la Torre, J. Hodgins, A. Bargteil, X. Martin, J. Macey, A. Collado, and P. Beltran. Guide to the carnegie mellon university multimodal activity (cmu-mmac) database. 2008.

[9] J. Deng, J. Krause, A. C. Berg, and L. Fei-Fei. Hedging your bets: Optimizing accuracy-specificity trade-offs in large scale visual recognition. In *CVPR*, 2012.

[10] B. G. Fabian Caba Heilbron, Victor Escorcia and J. C. Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *CVPR*, 2015.

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

[12] A. Graves. *Supervised sequence labelling with recurrent neural networks*. Springer, 2012.

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

[14] S. Karayev, T. Baumgartner, M. Fritz, and T. Darrell. Timely object recognition. In *NIPS*, 2012.

[15] S. Karayev, M. Fritz, and T. Darrell. Anytime recognition of objects and scenes. In *CVPR*, 2014.

[16] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In *CVPR*, 2014.

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

[18] H. S. Koppula and A. Saxena. Anticipating human activities using object affordances for reactive robotic response. RSS, 2013.

[19] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. Hmdb: a large video database for human motion recognition. In *ICCV*, 2011.

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

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

[22] J. Liu, J. Luo, and M. Shah. Recognizing realistic actions from videos in the wild. In *CVPR*, 2009.

[23] M. Marszalek, I. Laptev, and C. Schmid. Actions in context. In *CVPR*, 2009.

[24] R. Messing, C. Pal, and H. Kautz. Activity recognition using the velocity histories of tracked keypoints. In *ICCV*, 2009.

[25] T. B. Moeslund, A. Hilton, and V. Krüger. A survey of advances in vision-based human motion capture and analysis. *Computer vision and image understanding*, 2006.

[26] S. Oh, A. Hoogs, A. Perera, N. Cuntoor, C.-C. Chen, J. T. Lee, S. Mukherjee, J. Aggarwal, H. Lee, L. Davis, et al. A large-scale benchmark dataset for event recognition in surveillance video. In *CVPR*, 2011.

[27] P. Over, G. Awad, J. G. Fiscus, B. Antonishek, M. Michel, W. Kraaij, A. F. Smeaton, and G. Quénot. Trecvid 2010 - an overview of the goals, tasks, data, evaluation mechanisms and metrics. In *TRECVID*, 2010.

[28] R. Poppe. A survey on vision-based human action recognition. *Image and vision computing*, 2010.

[29] M. D. Rodriguez, J. Ahmed, and M. Shah. Action mach a spatio-temporal maximum average correlation height filter for action recognition. In *CVPR*, 2008.

[30] M. Rohrbach, S. Amin, M. Andriluka, and B. Schiele. A database for fine grained activity detection of cooking activities. In *CVPR*, 2012.

[31] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning internal representations by error propagation. Technical report, DTIC Document, 1985.

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

[33] M. Ryoo and J. Aggarwal. Ut-interaction dataset, icpr contest on semantic description of human activities(sdha), 2010.

[34] C. Schuldt, I. Laptev, and B. Caputo. Recognizing human actions: a local svm approach. In *ICPR*, 2004.

[35] G. Sergio, N. Krishnamoorthy, G. Malkarnenkar, T. Darrell, R. Mooney, and K. Saenko. Youtube2text: Recognizing and describing arbitrary activities using semantic hierarchies and zero-shoot recognition. In *ICCV*, 2013.

[36] S. Stein and S. J. McKenna. Combining embedded accelerometers with computer vision for recognizing food preparation activities. In *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing*. ACM, 2013.

[37] K. Tang, L. Fei-Fei, and D. Koller. Learning latent temporal structure for complex event detection. In *CVPR*, 2012.

[38] M. Tenorth, J. Bandouch, and M. Beetz. The tum kitchen data set of everyday manipulation activities for motion tracking and action recognition. In *ICCV Workshops*, 2009.

[39] H. Wang, A. Klaser, C. Schmid, and C.-L. Liu. Action recognition by dense trajectories. In *CVPR*, 2011.

[40] D. Weinland, R. Ronfard, and E. Boyer. A survey of vision-based methods for action representation, segmentation and recognition. *Computer Vision and Image Understanding*, 2011.