2019 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC)

Activity Forecasting in Routine Tasks by Combining Local Motion Trajectories and High-level Temporal Models

Yanxia Zhang, Andreas Girgensohn and Yulius Tjahjadi
FX Palo Alto Laboratory
Palo Alto, USA
{yzhang| $andreasg$ | $yulius$}@$fxpal.com$

Abstract

Human activity forecasting from videos in routine-based tasks is an open research problem. There are numerous applications in robotics, visual monitoring and skill assessment. Currently, activity forecasting has many challenges because human actions are not fully observable from continuous recording. Additionally, a large number of activities involve fine-grained articulated human motions that are hard to capture using frame-level representations. To overcome these challenges, we propose a method that forecasts human actions by learning the dynamics of local motion patterns. We employed dense trajectories to extract local atomic action patterns and Long Short-Term Memory (LSTM) neural networks for high level action dependencies modeling. The experiments on a public dataset validated the effectiveness of our proposed method in activity forecasting and demonstrated large improvements over the baseline two-stream end-to-end model. We learned that human activity forecasting benefits from learning both the short-range motion patterns and long-term dependencies between actions.

Human activity forecasting, dense trajectories, temporal convolutional network (TCN), long short-term memory (LSTM), vision-based skill assessment

1. Introduction

Continuous recognition of human activities from video cameras is an open research problem that has a broad range of applications in manufacturing, healthcare and human-computer interaction [1–3]. For example, in factory settings, workflow tasks usually involve a fixed routine set of structured actions such as the worker picking up the screwdriver, moving it to a component, and tightening screws there. It is desirable to design intelligent systems that can sense the ongoing human activities and anticipate the next step for timely assistance by a robot, an external human expert, or a digital interface. To develop such systems, temporal analysis of human actions and continuous activity recognition and segmentation of routine tasks needs to be performed.

Action forecasting predicts what action is occurring in a video stream [4, 5]. The action labels are inferred before the entire action execution is fully observed, possibly at an early stage when only very few frames of the action are observed. In a variety of real-world applications such as workflow monitoring, video streams contain sequences of multiple routine actions that are long, complex and repetitive [3–6]. For example, when changing tires, the task involves fine-grained motions with hard-to-detect fasteners and tools such as lug nut and lug wrench. The actions of tightening and loosening the lug nut are similar in their appearances and vary mostly in their motion pattern dynamics. Additionally, loosening the lug nuts can only happen before unscrewing the lug nuts. Similarly, factory workers need to follow the workflow designed for the assembly line and frequently use fasteners and tools such as screws and screw drivers. Video-based surgical skills evaluation systems is another rapidly growing field [3, 6]. Surgeons are trained to first master basic surgical skills such as suturing and knot tying that involve hand movements in a repetitive pattern before they perform complex tasks. Currently, these activities are video recorded offline and annotated manually by experts that is time-consuming and costly. Forecasting human actions in these scenarios is an important step to design intelligent systems for skills training and improving workflow efficiency.

This work addresses the challenges of analyzing routine tasks such as workflow videos for skill training. We propose a novel system that forecasts human actions by combining local motion trajectories and long-term action dependencies. We investigate the benefits of combining hand-crafted features and deep learning based predictive models for action forecasting in rou-

978-1-7281-1664-8/19/$31.00 ©2019 IEEE
DOI 10.1109/CSE/EUC.2019.00089

tine tasks that involve fine-grained motion dynamics and have little training data.

We perform empirical evaluations to compare our method with the state-of-art two -stream end-to-end approach on a public available dataset. We analyze the effectiveness of using local motion trajectories while considering dependencies between actions and the performance of different feature vectors. Using dense trajectories and Fisher vector representations improved the accuracy by 5% with a Support Vector Machine (SVM) model. Taking into account the temporal effect with a Temporal Convolutional Network (TCN) model improved the prediction performance by 10%. The best results were achieved by using a Long Short-Term Memory (LSTM) model that significantly outperformed the baseline method by over 15%. This is explained by the fact that the LSTM model takes into account the transitions between high-level actions, thus making smooth and continuous predictions.

2. Related Work

2.1. Activity Recognition

A primary research focusing on human action recognition has been with videos from consumer, entertainment, and sports programs [7–9]. These research activities deal with large-scale action classification where each action significantly differs from each other. For instance, the Kinectics human action video dataset [9] contains around 300k videos of 400 actions such as *shaking hands*, *riding a bike*, *brushing hair*, etc. Hundreds of videos are available for each action type. Similarly, the ActivityNet dataset [8] includes around 20k videos of 200 actions. The action recognition task deals with the problem of detecting whether the video contains certain actions. These video clips are trimmed to around 10 seconds and labeled with only one class although it may contain multiple action classes.

The state-of-art method in the large-scale action classification task is based on the two-stream model introduced by Simonyan and Zisserman [10] that has been demonstrated to achieve high performance. Their method utilizes two replicates of a ImageNet-pretrained ConvNet: one spatial stream taking in a single RGB frame; the other flow stream taking in a stack of 10 precomputed optical flow frames. Multiple snapshots are sampled from the video and feed into the two stream network to make prediction. The final action prediction is the average of the output from the multiple snapshots. A detailed review of current action classification methods can be found in [9].

In summary, these methods usually require a large amount of training data with ground truth annotations to train the architectures from scratch. The collection of such large-scale annotated dataset is difficult in various rapidly growing AI application areas such as healthcare and manufacturing industries. In large-scale action classification, datasets contain mostly manually pre-segmented videos of coarse-grained activities that exhibit high, inter-class variability. The focus of existing action recognition work has been on capturing object and scene-level information from short image sequences that is different from learning long-range, temporal actions in streaming videos.

2.2. Activity Forecasting

The goal of forecasting human actions in structured task is to predict the action in time given a continuous video stream. The majority of prior works assume the presence of a single action [7–10]. In contrast, we deal with a realistic setup where video streams contain multiple sequential actions. Specifically, we focus on routine activities that contain multiple actions that are often subtly different from one another and follow a specific order. For example, picking up a table leg should precede screwing it in when assembling Ikea furniture. Daily cooking and shopping activities are commonly studied scenarios [11].

Prior research proposed techniques to forecast future activities in continuous videos based on a Markov process. The assumption is that the next action only depends on the previous one. Kuehne et al. [11] used Hidden Markov Models combined with grammar defining action transitions to model cooking activities. For feature representations, they used a bag-of-words (BoW) approach over space-time interest points (STIP) detected using Harris operators. Chakraborty and Roy-Chowdhury [2] adopted the same STIP+BoW approach to extract action features from video frames. They addressed the forecasting problem by inference on a Markov Random Field (MRF) model defined on the human activity sequence graph. Recently, Han et al. [4] proposed deep architectures for this task. The model in [4] is end-to-end and learn action features from a two-stream ConvNet architecture [10].

Our work differs from prior works [2, 4, 11] in two aspects. First, in terms of action feature representations, we use dense trajectories to extract local spatial-temporal patterns instead of frame-level representations. A Gaussian Mixture Model (GMM) is used to build a code book for the descriptors of the dense trajectories in each feature categories (Trajectory, HOG, HOF and MBH). Then, a short sequence of the video is represented using Fisher vectors from the trained code

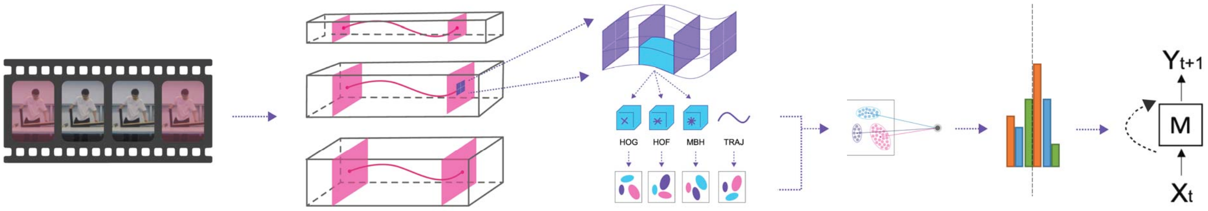

Figure 1. Pipeline of our method: (1) Feature extraction - Dense trajectories around interest points sampled at different scales are extracted from each frame. (2) Feature Encoding - Features along the trajectories from the video sub-sequence are represented as Fisher vectors (FV). (3) Prediction - The final representation is fed as input to a prediction model.

book. Second, we combine the short-range action feature representations with high-level temporal models such as TCN and LSTM to learn long-range action dependencies.

3. Method

Our activity forecasting method predicts the next action given the previous and current observations from video streams (see Figure 1). The problem is defined as follows: given a set of n videos $V = V_1, V_2, ..., V_n$, each video V_i contains K actions belonging to a total number of C types. Let $S = \{f_t\}_{t=1}^{M}$ be a sub-sequence of V containing M consecutive frames. The goal is to predict an action label at time t denoted as y_t where $t > M$ and $y \in \{1, ..., C\}$. The prediction target y is encoded as 1 for the true class and 0 for all others. Figure 1 illustrates the pipeline of our method. The first step is to extract local motion trajectories from each frame within a sliding window at different scales. Then features from the video sub-sequence are represented as Fisher vectors (FV) and used as input to the prediction model.

3.1. Extracting Local Motion Trajectories

Our work leverages work on dense trajectories by Wang et al. [7]. Their approach uses optical flow to detect trajectories across a fixed number of video frames. In addition to the coordinates of the trajectories (TRAJ), descriptors such as histograms of optical flow (HOF), histograms of orientation of gradients (HOG), and motion boundary histograms (MBH) are computed along the spatio-temporal volume encapsulating the trajectories. We used each of those descriptors of human activity forecasting and compared their performance.

We made several modifications to customize Wang et al.'s implementation [7]: 1) video frames are extracted at 30 fps, 2) high-definition videos are normalized to have a frame height of 360 pixels. We used the same trajectory length of $L = 15$ frames. Points are sampled in a $W \times W$ neighborhood where $W = 5$ in our experiment. Descriptors are computed along the trajectory in a 32×32 pixels image patch size that is divided into a grid of 2×2 spatial cells and 3 temporal cells. The dimensions of the descriptors are 30 for TRAJ, 96 for HOG, 108 for HOF and 96 for MBHx and MBHy. Given a video sequence of M frames $S = \{f_t\}_{t=1}^{M}$, we sample and track interesting points at different scales and extract thousands of trajectories around these points.

3.2. Feature Encoding

We use Fisher vectors (FV) to aggregate the extracted dense features into a high-dimensional vector representation [12]. For each type of the descriptors (TRAJ, HOF, HOG, MBH), we use principal component analysis (PCA) to reduce the dimensions to half of the original dimensions. For each of the feature types (TRAJ, HOF, HOG, MBH), we fit a Gaussian Mixture Model (GMM) with diagonal covariances and set the number of Gaussians to K (K=128, 256, 512). The feature instances extracted from the training set are used to learn the codebook. For a large dataset, a subset of the features can be randomly sampled to fit the GMM. During the evaluation, we used $K = 256$ components of Gaussians that has demonstrated good performance on action localization in prior work [13].

The features are encoded by the derivatives of the log-likelihood of the GMM model with respect to its parameters. The FV computes the derivatives with respect to the Gaussian mean and variances that represents the first and second order differences between features and centers of the GMM. Finally, we apply the power normalization and then the L2 normalization to obtain improved Fisher vectors as introduced by [12]. Each subsequence $S = \{f_t\}_{t=1}^{M}$ is represented by a Fisher vector x with respect to the parameters of a pre-trained diagonalcovariance GMM. The dimension of Fisher vector representation x is $K \times D$, where D is the dimension of each feature descriptor.

3.3. Activity Prediction

For consecutive M frames from a video subsequence $S = \{f_t\}_{t=1}^{M}$, the local motion patterns are represented by a set of Fisher vectors denoted as the input X to a prediction model. Given the input Fisher vector X_t for time step t, we aim to predict the action label for each frame denoted by Y_t.

TCN model: Similar to Convolutional Neural Network (CNN) models, a Temporal Convolutional Network (TCN) consists of repeated blocks of convolutions followed by non-linear activations. In a TCN model, the l^{th} convolution layer with a temporal window of d_l includes N_l filters where each filter has weight $W^{(i)}$ ($i \in [1, N_l]$). The output X_l of the l^{th} layer is defined as:

$$X_l = f(W * X_{l-1}) \tag{1}$$

where f is a non-linear activation function and X_{l-1} represents the output of the previous layer. In our implementation, the network architecture consists of a 1D convolutional layer with ReLU activations ($N_l = 64$, $d_l = 25$) followed by a time-distributed, fully-connected layer. A spatial-dropout layer with rate 0.3 is added to the network to improve performance. For action prediction, we use a softmax layer with output neurons equal to the number of classes.

LSTM model: We apply one type of a recurrent neural network, Long Short-Term Memory (LSTM) to predict action types. Compared to TCN, LSTM utilizes the sequential information of the input data and processes current sub-sequences given information extracted from previous sub-sequence with the use of memory cells. Therefore, it can learn both short-term and long-term dependency patterns from input features. At each time step t, the LSTM is trained to predict the next action at $t+1$, given a set of Fisher vector representations X_t from the sub-sequences within the sliding window. Our network architecture is based on a bi-directional LSTM that is made up of two reversed uni-directional LSTMs. The model implemented in our experiment consists of a bi-directional LSTM layer with 64 memory units followed by a time-distributed, dense layer. Finally the outputs of the time-distributed, dense layer are fed into the output softmax layer for prediction.

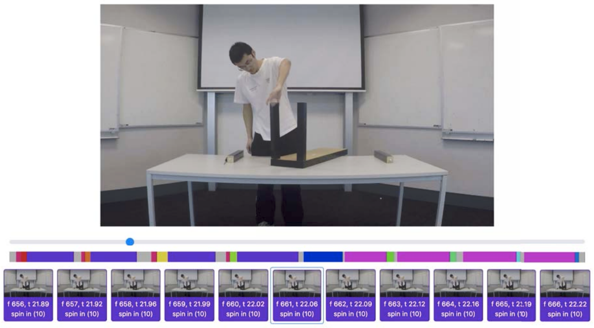

Figure 2. Illustration of the Ikea furniture assembly dataset. Each video contains a sequence of actions that an individual assembles and disassembles the table either on a workbench or on the ground.

To train the networks, we used the categorical cross-entropy loss with Stochastic Gradient Descent and ADAM step updates. All models were optimized by training for 200 epochs. The TCN and LSTM networks were implemented using the deep-learning library *Keras* with a *TensorFlow* backend.

We also used a linear SVM with $C = 1$ to perform independent frame-wise prediction. A one-vs-rest strategy is used for multi-class classification. The SVM model used the implementation from *Scikit-learn*.

4. Experiments

4.1. Dataset

We evaluate our method on the Ikea Furniture Assembly Dataset (IkeaFA) [4]. The dataset consists of 101 videos in total. The reason we selected this dataset is that it contains fine-grained human activities from structured tasks. One of the main characteristics of these activities focusing on human skills in sensory-motor control. The activities vary largely due to motion patterns rather than changes in objects or scenes. Compared to existing popular action recognition datasets such as UCF Sports and UCF 101 that contain exactly one action per video, this dataset contains videos that continuous recorded the tasks that consists of several subtasks. The un-trimmed characteristic makes this dataset more challenging and realistic so that we can explore the temporal dynamics in modeling human actions.

Each video in the dataset is about 5-7 minutes long containing a single person assembling an Ikea furniture, captured by a stationary GoPro camera in HD quality. The frame rate ranges from 45 to 60Hz. Each video recorded a person being instructed to assemble and disassemble an Ikea table. The procedures are that a person first picks up the table-top and then screws the four legs onto the table-top. At the beginning of each video,

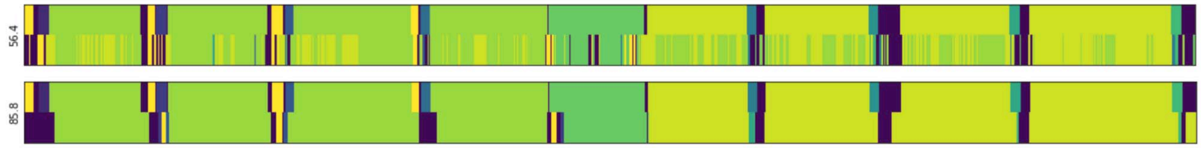

Figure 3. Frame-level action predictions using high level temporal models TCN (top) and LSTM (bottom). Modeling long-term action dependencies with deep recurrent models such as LSTM results in smooth and continuous action predictions.

the person starts with the components lying on either the table or floor and finishes with legs and table-top on the side. The temporal action annotation is available on each frame. There are 13 classes of actions in this task, such as *pick leg*, *attach leg X*, *detach leg X*, *flip table*, *screw in*, *screw out*, and a *null* action, where *X* is a number ranging from 1 to 4. The *null* action indicates that the current time interval does not belong to any other action class; such *null* actions appear between useful actions.

4.2. Evaluations

We compare the performance of our method to state-of-art end-to-end model. We investigate which high-level, temporal model prediction models perform the best and also the effectiveness of different descriptor types. We used the same data-splits as that from [4]. The IkeaFA dataset contains videos from 14 actors. We used videos from 11 actors for training and validation, and videos from the remaining 3 actors (actor id number 9, 11 and 13) for testing.

4.2.1. Visualize Extracted Motion Patterns. An example of extracted dense trajectories is shown in Figure 4. The red dots are the end points of the trajectory on the current frame. The green tracks show the densely-sampled, local-interest points that are tracked using optical flow with an applied medium filter.

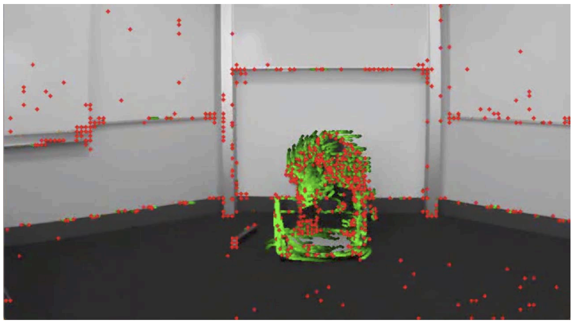

Figure 4. Illustration of the extracted dense trajectory on the Ikea furniture assembly dataset.

4.2.2. Comparison to the Baseline Model. We compared the performance of our method to the baseline, end-to-end, two-stream model [4]. Han et al. [4] applied the state-of-art, two-stream model for action recognition to the application of activity forecasting. Their method uses the features learned from a two-stream, residual CNN network for generating local action features. Table 1 presents the comparison results between the baseline CNN representations and Fisher vector representations based on dense trajectories. The baseline model is trained end-to-end from scratch. The results are evaluated individually on two setups of assembling furniture on a work bench (row 1 in table 1) and on the ground (row 2 in table 1). In contrast, we perform the evaluations without regard to the setups. A temporal sliding window size of 30 frames is used in the experiment. We compared the average prediction accuracy over all frames and weighted mean average precision from all action classes.

The evaluation results show that our proposed model outperforms the baseline model significantly. Using the TCN model for prediction achieved an accuracy of 57.1% and an average precision of 60.0%, compared to an accuracy of 47% and an average precision of 47% from [4]. Using the LSTM model for prediction improved the performance further by another 8%, with an accuracy of 65.2% and an average precision of 71.0%. When we examine the predictions made by the TCN and LSTM model on the test videos, we observed that LSTM makes smooth and continuous predictions compared to TCN as illustrated in Figure 3. The recurrent models may have learned the long-term action dependencies and maintained the knowledge of temporal evolution of actions, i.e., screwing-in spanning over tens of seconds.

4.2.3. Effects of Different Feature Types. Figure 5 presents the accuracies of using different descriptor types contained in the local trajectory patterns. When using an SVM model, MBH descriptors that capture the changes of motion perform the best by themselves with an accuracy of 53.6%. TRAJ descriptors that describe the shape of the motion of the trajectories achieved the

Table 1. Performance compared to the baseline method. MAP (mean average precision)

Method	Accuracy	MAP
Two-Stream IkeaBench [4]	47.1	47.0
Two-Stream IkeaGround [4]	40.8	28.6
Ours TCN	57.1	60.0
Ours LSTM	**65.2**	**71.0**

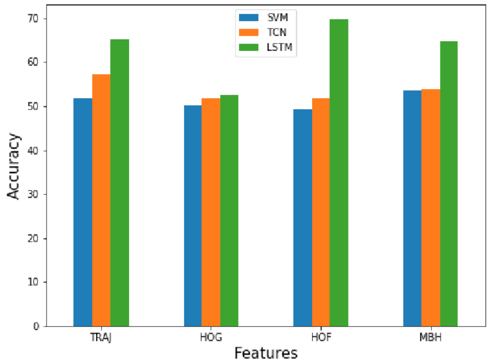

Figure 5. Accuracy of different feature types and prediction models

best accuracy of 57.1% when combined with TCN. The combination of HOF descriptors and LSTM gives the overall best results of 69.6%. Although there is no prevalent feature category, motion descriptors consistently outperform appearance-based HOG descriptors. These results indicate that the importance of motion information in short-range, local patterns.

5. Conclusion

We investigated whether combining hand-crafted features and deep-learning predictive models would benefit activity forecasting in routine tasks that involve fine-grained motion dynamics and have little training data. The result is promising and suggests that combining appropriate, hand-crafted features with deep-learning models can overcome the challenges when annotated data are sparse and difficult to obtain. Our technique is particularly applicable to human activities that involve fine-grained motion and follow specific orders. Our activity-forecasting system may find promising applications in areas of healthcare and manufacturing industries.

References

[1] D. Weinland, R. Ronfard, and E. Boyer, “A survey of vision-based methods for action representation, segmentation and recognition,” *Computer vision and image understanding*, vol. 115, no. 2, pp. 224–241, 2011.

[2] A. Chakraborty and A. K. Roy-Chowdhury, “Context-aware activity forecasting,” in *Asian Conference on Computer Vision*. Springer, 2014, pp. 21–36.

[3] A. Zia, Y. Sharma, V. Bettadapura, E. L. Sarin, T. Ploetz, M. A. Clements, and I. Essa, “Automated video-based assessment of surgical skills for training and evaluation in medical schools,” *International journal of computer assisted radiology and surgery*, vol. 11, no. 9, pp. 1623–1636, 2016.

[4] T. Han, J. Wang, A. Cherian, and S. Gould, “Human action forecasting by learning task grammars,” *arXiv:1709.06391*, 2017.

[5] C. Lea, M. D. Flynn, R. Vidal, A. Reiter, and G. D. Hager, “Temporal convolutional networks for action segmentation and detection,” in *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 156–165.

[6] A. Jin, S. Yeung, J. Jopling, J. Krause, D. Azagury, A. Milstein, and L. Fei-Fei, “Tool detection and operative skill assessment in surgical videos using region-based convolutional neural networks,” in *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2018, pp. 691–699.

[7] H. Wang and C. Schmid, “Action recognition with improved trajectories,” in *Proceedings of the IEEE international conference on computer vision*, 2013, pp. 3551–3558.

[8] F. Caba Heilbron, V. Escorcia, B. Ghanem, and J. Carlos Niebles, “Activitynet: A large-scale video benchmark for human activity understanding,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 961–970.

[9] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev *et al.*, “The kinetics human action video dataset,” *arXiv preprint arXiv:1705.06950*, 2017.

[10] K. Simonyan and A. Zisserman, “Two-stream convolutional networks for action recognition in videos,” in *Advances in neural information processing systems*, 2014, pp. 568–576.

[11] H. Kuehne, A. Arslan, and T. Serre, “The language of actions: Recovering the syntax and semantics of goal-directed human activities,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 780–787.

[12] F. Perronnin, J. Sánchez, and T. Mensink, “Improving the fisher kernel for large-scale image classification,” in *European conference on computer vision*. Springer, 2010, pp. 143–156.

[13] J. C. Van Gemert, M. Jain, E. Gati, C. G. Snoek *et al.*, “Apt: Action localization proposals from dense trajectories.” in *BMVC*, vol. 2, 2015, p. 4.