# Supplementary Material:Hierarchical Modeling for Task Recognition and Action Segmentation in Weakly-Labeled Instructional Videos

Reza Ghoddoosian Saif Sayed Vassilis Athitsos
Vision-Learning-Mining Lab, University of Texas at Arlington
{reza.ghoddoosian, saififtekar.sayed}@mavs.uta.edu, athitsos@uta.edu

## 1. Overview

In this supplementary material, we show comparisons of I3D [1] and iDT [14] features in task recognition on two datasets, and present comparison results on the original 10 classes of the Breakfast dataset [7]. We also provide a glossary of terms and a table of symbols we use in the paper.

## 2. I3D and iDT Feature Comparison in Task Recognition of Weakly-Labeled Videos

In this section, we compare I3D and iDT features for the purpose of task recognition in weakly-labeled instructional videos. Specifically, we present results of existing models using I3D (Table 1) and iDT (Table 2) features on the MPII Cooking 2 dataset [13] as well as the first split of the Breakfast dataset [7].

We used the Fisher vectors of iDT features as in [7, 11]. The Fisher vectors for each frame are extracted over a sliding window of 20 frames. They are first projected to a 64-dimensional space by PCA, and then normalized along each dimension. Also, we extracted the I3D features of the Cooking 2 dataset using TV-L1 optical flow [16] on a moving window of 32 frames with a stride of 2, and the precomputed I3D features of the Breakfast dataset were obtained from [2]. Furthermore, we applied PCA to the extracted I3D features to reduce the dimensionality of RGB and optical flow channels from 1024 to 128. We fed the same features to all competitors except [15] in Table 3 whose code is not publicly available, so we compare with their reported result on ResNet101 [4] features.

In the Cooking 2 dataset, we train the models on the training split and test on the test split. However as [8] and [12] take a long time to train and infer the segments, in Tables 1 and 2, we only use the first split of the Breakfast dataset to evaluate the difference in performance of all models when using I3D and iDT features as input. Note that the reported task recognition results on the Breakfast dataset in the paper are the average of all four splits using the best case for each method.

Explicit task classification methods, e.g., ours, W-TALC [10] and 3C-Net [9], consistently perform better with I3D features on both datasets, whereas the bottom-up inference of tasks in NNViterbi [12] and CDFL [8] produces mixed result. In particular, the performance of [8] and [12] on the Breakfast dataset considerably improves upon using iDT features. Overall, the more significant presence of object information in I3D features helps to classify top-level tasks more accurately, while detecting fine-grained actions seems to be less affected by such appearance information.

| | Breakfast (1st split) (%) | | Cooking (%) | |
|---|---|---|---|---|
| Models | t-acc | t-mAP | t-acc | t-mAP |
| NNViterbi[12]* | 57.14 | - | 23.80 | - |
| CDFL[8]* | 66.26 | - | 28.57 | - |
| W-TALC[10]* | 75.79 | 78.96 | 33.33 | 43.07 |
| 3C-Net[9]* | 75.39 | 78.50 | 30.95 | 46.30 |
| Timeception[5]* | 79.50 | 82.53 | 21.43 | 25.14 |
| VideoGraph[6]* | 80.06 | - | 23.80 | - |
| Our Method | 81.74 | 88.30 | 45.24 | 54.49 |

Table 1. Task classification results of state-of-the-art methods using I3D features on the Cooking 2 dataset and the first split of the Breakfast dataset. (* results obtained using the author's source code).

| | Breakfast (1st split) (%) | | Cooking (%) | |
|---|---|---|---|---|
| Models | t-acc | t-mAP | t-acc | t-mAP |
| NNViterbi[12]* | 71.03 | - | 16.66 | - |
| CDFL[8]* | 77.38 | - | 21.42 | - |
| W-TALC[10]* | 53.17 | 54.96 | 19.04 | 25.85 |
| 3C-Net[9]* | 56.74 | 60.36 | 14.28 | 27.38 |
| Timeception[5]* | 65.87 | 71.73 | 9.52 | 14.36 |
| VideoGraph[6]* | 58.93 | - | 14.28 | - |
| Our Method | 60.31 | 61.72 | 23.80 | 27.66 |

Table 2. Task classification results of state-of-the-art methods using iDT features on the Cooking 2 dataset and the first split of the Breakfast dataset. (* results obtained using the author's source code).

| Models | t-acc | Feature | Test Split |
|---|---|---|---|
| Timeception[5] | 71.3 | 3D-ResNet [3] | Last 8 subjects |
| Timeception[5] | 69.3 | I3D (*pre pooling*) | Last 8 subjects |
| Timeception[5]* | 76.6 | I3D (*post pooling*) | Split 1 |
| VideoGraph[6] | 69.5 | I3D (*pre pooling*) | Last 8 subjects |
| VideoGraph[6]* | 79.9 | I3D (*post pooling*) | Split 1 |
| RhyRNN[15] | 44.3 | ResNet101 [4] | Split 1 |
| Our Method | 81.5 | I3D (*post pooling*) | Split 1 |
| Our Method | 85.2 | I3D (*post pooling*) | Last 8 subjects |

Table 3. Task classification results (t-acc) of state-of-the-art methods on the Breakfast dataset for 10 classes. (* results re-implemented using the author's source code).

## 3. Task Classification Results on 10 Classes of the Breakfast Dataset

Timception [5], VideoGraph [6] and RhyRNN [15] are the latest state-of-the-art methods to classify tasks in minutes-long videos and are the closest competitors to our work. We compared the standard four fold cross validated results of Timeception and VideoGraph over 9 classes of the Breakfast dataset in Table 1 of the paper, however, we could not compare our method to RhyRNN because the source code of RhyRNN is not publicly available to adjust that model to our evaluation settings. Hence, in Table 3, we present comparison results of our method with the reported accuracy of this method and different versions of other models over the original 10 classes of the Breakfast dataset. For a direct comparison with RhyRNN , we show results on the first split as reported in RhyRNN.

Furthermore, Table 3 shows the original reported results of Timeception and VideoGraph, which are lower than our re-implemented versions in both cases. Contrary to the standard splitting rule of the Breakfast dataset, both works have used the last 0.15% of subjects in the dataset (8 subjects) to test their performance. Our result on this split significantly outperforms previous methods (Table 3). [5] and [6] also use the output before the last average pooling layer (*pre pooling*) in the I3D network as features, unlike us, where we use the features after the pooling layer (*post pooling*). The results in Table 3 suggest the superiority of the latter, because the lower dimension after pooling allows each network to be given more features as input, which increases their input temporal range.

Interestingly, the task accuracy for most models, including ours, hardly drops upon evaluation on 10 classes and our method is still superior than different versions of state-of-the-art.

## 4. Glossary of Terms and Symbols

As there are similar terms and many symbols used in the paper, here, we provide specific definitions of terms (Table 5) and symbols (Table 4) for readers to refer to.

| Symbol | Definition |
|---|---|
| $\mathbb{A}$ | The set of all attributes |
| $a_j$ | Attribute j |
| $a_{i,j}$ | Attribute j of video i |
| $\mathbb{A}_i$ | The set of attributes in video i |
| $\vec{a}_i$ | Muiltihot ground-truth attribute vector of video i |
| $\vec{a}_i^w$ | TF-IDF weighted ground-truth attribute vector of video i |
| $A^T B$ | Matrix multiplication of A transposed and B |
| $a \cdot b$ | Scalar multiplication of a and b |
| $\beta$ | Importance factor of $\mathcal{L}_{th}$ in the total loss |
| $\mathbb{C}$ | The set of all tasks |
| $c_i$ | Task label for video i |
| $\vec{c}_i$ | One-hot task ground-truth vector of video i |
| $d$ | Stage duration |
| $F$ | Dimension of the feature encoding $\phi(\mathbf{x})$ |
| $f_i^c$ | Final fused classification logits |
| $g(\mathbf{x})$ | The fully connected layer to produce encoding $\phi(\mathbf{x})$ |
| $\mathcal{H}_x(\,)$ | Heaviside step function shifted to x |
| $h_\kappa$ | Feature summary of stage $\kappa$ |
| K | Number of stages in the THS stream |
| $\mathbf{k}_\phi$ | Temporal convolution kernels to produce $\phi(\mathbf{x})$ |
| $k_i$ | Number of selected frames of video i from the $\mathrm{top_k}$ operation |
| L | Kernel length of $\mathbf{k}_\phi$ |
| $\boldsymbol{l}_1^S$ | Sequence of S action durations in a video |
| $\mathcal{L}_{sh}$ | Loss function for the SHS stream |
| $\mathcal{L}_{th}$ | Loss function for the THS stream |
| $\mathcal{L}_f$ | Loss function of the fused streams |
| $M_i$ | Number of attributes in video i |
| $\mathbf{M}_{tfidf}$ | TF-IDF mask |
| $\mathcal{M}_x^a$ | Mapping function from features to attributes |
| $\mathcal{M}_a^c$ | Mapping function from attributes to tasks |
| N | Number of videos in the training set/batch |
| $S$ | Number of segments in a video |
| $\mathfrak{s}[\,]$ | Softmax operation |
| $\mathcal{S}_\kappa$ | Classifier for stage $\kappa$ in the THS stream |
| $s$ | The parameter used in the $\mathrm{top_k}$ operation |
| $T_i$ | Number of frames in video i |
| $\mathcal{T}$ | Stage aggregation function in the THS stream |
| $\tau$ | Task variable |
| $\vartheta_\kappa$ | Task prediction logits of stage $\kappa$ |
| $\vartheta_{total}$ | Stage-aggregated task prediction logits |
| $\mathbb{V}$ | Set/Batch of training videos |
| $\mathbf{v}_i$ | Video $i$ |
| $\mathbf{W}_{tfidf}$ | TF-IDF weights |
| $\mathbf{x}_i$ | Input feature vector for video i |
| $\phi(\mathbf{x})$ | Learned video feature encoding |
| $\psi_i^a$ | Attribute score vector of video i in the SHS stream |
| $\psi_i^c$ | Task score vector of video i in the SHS stream |
| $\Psi_i^a$ | T-CAM of video i |
| $\boldsymbol{\delta}_1^S$ | Sequence of S action labels in a video |
| $\lambda$ | Design parameter in $\mathcal{L}_{sh}$ |
| $\Pi(\tau)$ | Set of all action sequences in the training set given task $\tau$ |
| $\sigma(\,)$ | Sigmoid operation |
| $\Gamma(\,)$ | Stage-wise drop out in the stage aggregation function |
| $\mathbb{1}(\,)$ | Indicator function |
| $\odot$ | Element-wise product operation |

Table 4. Definitions of symbols used in the paper.

| Term | Definition |
|---|---|
| Action | Lower level actions happening in the form of segment sequence in instructional videos. |
| Action alignment | Partitioning the video into sequence of action segments given a sequence of action labels. |
| Action detection | Classify and localize occurrences of, typically, a single action in the video among considerable background frames. |
| Action segmentation | Partitioning the video into sequence of action segments. |
| Attribute | Set of actions or the set of verb/object components of actions. |
| Fully-supervised classification | Task classification using frame-level and video-level labels. |
| Instructional videos | Videos with a top-level task and a sequence of fine-grained actions to carry out the underlying task. |
| Task | The single top-level composite activity present in the video. |
| Task recognition | Classifying the top-level task in long instructional videos. |
| Weakly-labeled videos | Videos with no frame-level annotations. In our case, only sequence of video-level action labels is available. |
| Weakly-supervised classification | Task classification without access to frame-level annotation. We use the term "weak" to distinguish from fully-supervised methods. |

Table 5. Definitions of technical terms used in the paper.

## References


[1] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017.

[2] Yazan Abu Farha and Jurgen Gall. Ms-tcn: Multi-stage temporal convolutional network for action segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3575–3584, 2019.

[3] Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet? In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 6546–6555, 2018.

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[5] Noureldien Hussein, Efstratios Gavves, and Arnold WM Smeulders. Timeception for complex action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 254–263, 2019.

[6] Noureldien Hussein, Efstratios Gavves, and Arnold WM Smeulders. Videograph: Recognizing minutes-long human activities in videos. *arXiv preprint arXiv:1905.05143*, 2019.

[7] Hilde Kuehne, Ali Arslan, and Thomas 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*, pages 780–787, 2014.

[8] Jun Li, Peng Lei, and Sinisa Todorovic. Weakly supervised energy-based learning for action segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6243–6251, 2019.

[9] Sanath Narayan, Hisham Cholakkal, Fahad Shahbaz Khan, and Ling Shao. 3c-net: Category count and center loss for weakly-supervised action localization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 8679–8687, 2019.

[10] Sujoy Paul, Sourya Roy, and Amit K Roy-Chowdhury. W-talc: Weakly-supervised temporal activity localization and classification. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 563–579, 2018.

[11] Alexander Richard, Hilde Kuehne, and Juergen Gall. Action sets: Weakly supervised action segmentation without ordering constraints. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5987–5996, 2018.

[12] Alexander Richard, Hilde Kuehne, Ahsan Iqbal, and Juergen Gall. Neuralnetwork-viterbi: A framework for weakly supervised video learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7386–7395, 2018.

[13] Marcus Rohrbach, Anna Rohrbach, Michaela Regneri, Sikandar Amin, Mykhaylo Andriluka, Manfred Pinkal, and Bernt Schiele. Recognizing fine-grained and composite activities using hand-centric features and script data. *International Journal of Computer Vision*, 119(3):346–373, 2016.

[14] Heng Wang and Cordelia Schmid. Action recognition with improved trajectories. In *Proceedings of the IEEE international conference on computer vision*, pages 3551–3558, 2013.

[15] Tianshu Yu, Yikang Li, and Baoxin Li. Rhyrnn: Rhythmic rnn for recognizing events in long and complex videos. In *European Conference on Computer Vision*, pages 127–144. Springer, 2020.

[16] Christopher Zach, Thomas Pock, and Horst Bischof. A duality based approach for realtime tv-l 1 optical flow. In *Joint pattern recognition symposium*, pages 214–223. Springer, 2007.