Predictive-Corrective Networks for Action Detection

Achal Dave Olga Russakovsky Deva Ramanan
Carnegie Mellon University

1. Introduction

Computer vision is undergoing a period of rapid progress. While the state-of-the-art in image recognition is disruptively increasing, the same does not quite hold for video analysis. We believe one reason is that many architectures and optimization techniques used for video are largely inspired by those for static images (e.g., "two-stream" models [5]), though notable exceptions that directly process spatio-temporal volumes exist.

Recurrent models: An attractive solution to the above are *state-based* models that implicitly process large spatio-temporal volumes by maintaining a hidden state over time. Classic temporal models based on Hidden Markov Models (HMMs) or Kalman filters do exactly this. Their counterpart in the world of neural nets are recurrent models, which are relatively less explored for video-based learning. We posit that this could be because temporal data streams are highly correlated, while most SGD solvers rely heavily on *i.i.d.* data for efficient training. Typical methods for ensuring uncorrelated data (random data permutations) remove the very temporal structure that we are trying to exploit!

Our approach: We rethink both the underlying network architecture and stochastic learning paradigm, drawing inspiration from classic theory on linear dynamic systems for time-series learning models. By extending such iconic models to include nonlinear hierarchical mappings, we derive a series of novel recurrent neural networks that work by making top-down *predictions* about the future and *correct* those predictions with bottom-up observations (Fig. 1).

Prediction: From a biological perspective, we leverage the insight that the human vision system relies heavily on continuously predicting the future and then focusing on the unexpected. This serves two goals: (1) achieves consistency in predicted actions, and (2) reduces the computational burden when no significant changes are observed.

Correction: More importantly, explicitly modeling appearance predictions allows the model to focus on unexpected events. By explicitly focusing on these corrections, our model is able to identify action transitions much more reliably. Further, from a statistical perspective, focusing on changes addresses a key challenge in learning from sequential data: it reduces correlations between consecutive samples. While consecutive video frames are highly correlated, *changes* between frames are not, increasing the diversity of training samples.

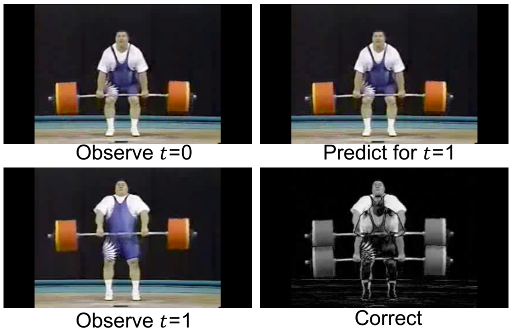

Figure 1. Our model first **predicts** the future and then updates its predictions with **corrections** from observing subsequent frames.

Contributions: We introduce a lightweight, intuitive and interpretable model for temporal action localization. By making predictions about future frames and subsequently correcting its predictions, the model is able to achieve significant improvements in both recognition accuracy and computational efficiency. We demonstrate action localization results on three benchmarks: THUMOS [2], MultiTHUMOS [6] and Charades [4]. Our model is competitive with the two-stream network [5] on all three datasets without the need for computationally expensive optical flow. Further, it (marginally) outperforms the state of the art MultiLSTM model on MultiTHUMOS [6].

2. Predictive-Corrective Model

We provide intuition for our model motivated by Kalman Filters. Our model smoothly updates its memory over consecutive frames with residual corrections based on frame changes, yielding an accurate and efficient framework.

2.1. Linear Dynamic Systems

Consider a single-shot video sequence that evolves continuously over time. For a video frame at time t, let $\mathbf{x}_t$ denote the underlying semantic representation of the state. For example, actions can be decomposed into mini muscle motions, and $\mathbf{x}_t$ can correspond to the extent each of these motions is occurring at time t. The action detection model observes the pixel frame appearance $\mathbf{y}_t$ and is tasked with making an accurate semantic prediction $\hat{\mathbf{x}}_t$ of the state $\mathbf{x}_t$.

Dynamics: Consider modeling the video sequence as a lin-

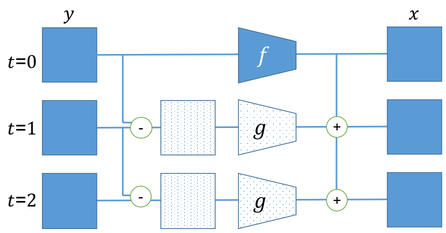

Figure 2. An instantiation of our predictive-corrective block.

ear dynamic system, evolving according to $\mathbf{x}_t = \mathbf{A}\mathbf{x}_{t-1} + \textit{noise}$, and $\mathbf{y}_t = \mathbf{C}\mathbf{x}_t + \textit{noise}$. That is, the semantic state $\mathbf{x}_t$ is a noisy linear function of the semantic state at the previous time step $\mathbf{x}_{t-1}$, and the pixel-level frame appearance $\mathbf{y}_t$ is a noisy linear function of the semantic action state $\mathbf{x}_t$.

Kalman filter: Under this linear dynamic system assumption, the posterior estimate of the action state $\mathbf{x}_t$ is $\hat{\mathbf{x}}_t = \hat{\mathbf{x}}_{t|t-1} + \mathbf{K}(\mathbf{y}_t - \hat{\mathbf{y}}_{t|t-1})$, where $\hat{\mathbf{x}}_{t|t-1}$ and $\hat{\mathbf{y}}_{t|t-1}$ are the prior prediction of $\mathbf{x}_t$ and $\mathbf{y}_t$ respectively given observations up to previous time step $t-1$, and $\mathbf{K}$ is the Kalman gain matrix. We analyze this posterior estimate in detail.

State approximation: To make predictions of the semantic action space $\hat{\mathbf{x}}_{t|t-1}$ and of appearance $\hat{\mathbf{y}}_{t|t-1}$, we rely on the fact that the actions and pixel values of a video evolve *slowly* over time. Thus, we can approximate $\hat{\mathbf{x}}_{t|t-1} \approx \hat{\mathbf{x}}_{t-1}$, and approximate $\hat{\mathbf{y}}_{t|t-1} \approx \mathbf{y}_{t-1}$. The posterior estimate simplifies to:

$$\hat{\mathbf{x}}_t = \underbrace{\hat{\mathbf{x}}_{t-1}}_{\text{prediction}} + g\underbrace{(\mathbf{y}_t - \mathbf{y}_{t-1})}_{\text{correction}} \tag{1}$$

where g is a learned function, helping compensate for the imperfect assumptions made here.

Learning: What remains is learning the non-linear function g from differences in frame appearance to differences in action state. We call this a *predictive-corrective block* (as in Fig. 2) and it forms the basis of our model. We refer to our paper [1] for more details about placement.

3. Experiments

We evaluate our predictive-corrective model on three challenging benchmarks: MultiTHUMOS [6], THUMOS [2], and Charades [4]. We refer to our full paper [1] for implementation details, and will release source code for training and evaluating our models.

3.1. Datasets

THUMOS, MultiTHUMOS: THUMOS contains 20 annotated action classes; MultiTHUMOS includes 45 additional action classes annotated on the THUMOS videos. We train models on the training and validation videos for all the MultiTHUMOS actions jointly, evaluating on the THUMOS test videos by computing the per-frame mAP over the 20 THUMOS and 65 MultiTHUMOS action classes.

Charades: Charades consists of common actions in homes, and is a more challenging test bed: it contains 157 actions and is constructed to decorrelate actions from scenes.

Method	Multi-THUMOS	THUMOS	Charades
Single-frame [6]	25.4	34.7	7.9
LSTM (on RGB)	28.1	39.3	7.7
Two-Stream [2][5]	27.6	36.2	**8.9**
Ours	**29.7**	38.9	**8.9**

Table 1. Comparison of our model with prior work. (Per-frame mAP on MultiTHUMOS, THUMOS, and Charades test sets.)

3.2. Results

We report results in Table 1. On MultiTHUMOS, we show that our model compares favorably with the state-of-the-art. On THUMOS our model outperforms baselines, but is not yet on par with MultiLSTM [6], possibly because the LSTM-based model can better handle the longer actions due to a longer (though less interpretable) memory. At the cost of efficiency, we can further improve our model by running it in a dense sliding window fashion, achieving 30.8% mAP on MultiTHUMOS (outperforming MultiLSTM's 29.6%) and 40.9% on THUMOS (only 0.4% behind MultiLSTM at 41.3%). On the Charades dataset, our predictive-corrective model improves over the single-frame and LSTM baselines, and matches the accuracy of the two-stream network, without needing expensive optical flow. [1]

References

[1] A. Dave, O. Russakovsky, and D. Ramanan. Predictive-corrective networks for action detection. *CVPR*, 2017. 2

[2] Y. Jiang, J. Liu, A. R. Zamir, G. Toderici, I. Laptev, M. Shah, and R. Sukthankar. Thumos challenge: Action recognition with a large number of classes. In *ECCV Workshop*, 2014. 1, 2

[3] G. A. Sigurdsson, S. Divvala, A. Farhadi, and A. Gupta. Asynchronous temporal fields for action recognition. *arXiv preprint arXiv:1612.06371*, 2016. 2

[4] G. A. Sigurdsson, G. Varol, X. Wang, A. Farhadi, I. Laptev, and A. Gupta. Hollywood in homes: Crowdsourcing data collection for activity understanding. In *ECCV*, 2016. 1, 2

[5] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *NIPS*, 2014. 1, 2

[6] S. Yeung, O. Russakovsky, N. Jin, M. Andriluka, G. Mori, and L. Fei-Fei. Every moment counts: Dense detailed labeling of actions in complex videos. *IJCV*, 2015. 1, 2

[1] For completeness, we note that the model does not yet match state-of-the-art results on the Charades benchmark: [3] achieves 12.5% mAP using global cues and post-processing.

[2] The two-stream number on MultiTHUMOS and THUMOS is reported from [6], which uses a single optical flow frame for the flow stream.