Self-Supervised Temporal Event Segmentation Inspired by Cognitive Theories

Ramy Mounir[1], Sathyanarayanan Aakur[2], Sudeep Sarkar[1]
[1]Computer Science and Engineering, University of South Florida, Tampa, USA
[2]Computer Science, Oklahoma State University, Stillwater, USA

Chapter Points

We can use cognitive science theories in event segmentation to design highly effective computer vision algorithms for Spatio-temporal segmentation of events in videos that do not require any labeled data.

We discuss three perceptual prediction models from EST in three progressive versions: temporal segmentation using perceptual prediction framework, temporal segmentation along with event working models based on attention maps, and finally spatial and temporal localization of events.

The approaches can learn robust event representations from only a single-pass through an unlabeled streaming video.

They show state-of-the-art performance in unsupervised temporal segmentation and spatial-temporal action localization while offering competitive performance with fully supervised baselines that require extensive amounts of annotation.

Abstract

We consider the event segmentation problem, i.e., how to mark the event's temporal boundaries and spatially locate them in the image. We draw heavily from cognitive science research to define the problem of event segmentation and to design highly effective computer vision algorithms for spatio-temporal segmentation of events in videos. These approaches do not require any annotated data. They can process streaming video data while learning robust representations for segmenting events. First, we introduce the Event Segmentation Theory (EST) model based on the perceptual prediction model to compute the event boundaries proposed by Zacks *et al.* [66]. Then we present our computer vision solutions based on the perceptual prediction model from EST in three progressive versions: temporal segmentation using perceptual prediction framework, temporal segmentation along with event working models based on attention maps, and finally spatial and temporal localization of events.

As in the EST, a perceptual processing unit (a CNN stack) sends the extracted features from the current frame (conditioned on the working event model) to a prediction unit (LSTM), which predicts future perceptual features. A mismatch between the predicted features and the actual, observed features generates a prediction error signal. The prediction error triggers a gating mechanism to update the working event model (hidden states in the LSTM) with a new model and mark an event boundary.

Extensive experiments demonstrate that the predictive learning approach can learn robust event representations from only a single-pass through an unlabeled streaming video. We show state-of-the-art performance in unsupervised temporal segmentation and spatial-temporal action localization while offering competitive performance with fully supervised baselines that require extensive amounts of annotation. Additionally, we show that our framework can process streaming video data of extremely long duration, close to real-time processing.

Keywords: Temporal event segmentation, self-supervised learning, representation learning, cognitive psychology, perception, action localization, streaming perceptual processing, wildlife monitoring, continual learning.

Contents

1 Introduction

How do we detect and segment events? How do we represent events? How do we perceive events? And, more importantly, what is an event? In computer vision research, the terms "actions", "activities", and "events" are often conflated. Most works lump these terms to represent "some activity happening" in the scene involving objects and actors labeled with a phrase, i.e., jumping, chopping an onion, changing tires, cooking a meal, and so on. There is no clear distinction among activity recognition, action recognition, or event recognition in existing computer vision literature. There is also not much clarity into the nature of events. On the contrary, event perception is a mature field of cognitive science research [66, 76, 70]. We will start by summarizing some of the cognitive science findings that we use as inspiration to build the solutions outlined in this chapter. Ideally, one should read the references cited and not just rely on our summary, which is just a brief introduction to a much richer body of work. We will highlight some ideas that we use to construct computer vision solutions to perform self-supervised temporal event segmentation.

Events are the central units of our experiences. The brain receives a continuous stream of sensory input, both from the outside world and inside the body, and segments them into discrete units or packet representations called events. Each event is indicative of key moments in the input sensory stream. Hence, an event is defined as *"a segment of time at a given location that is perceived by an observer to have a beginning and an end"* [106]. Note that this definition of events makes it distinct from activities. For example, cooking is an activity, but "cooking a meal" is an event. Just like a story, the latter has a beginning, a middle, and an end. Events can be of different types and duration, depending on the agent and acted upon environments. Some events are short, e.g., making a bed. Some events are extended, e.g., a cricket match. Events involving animate agents, such as humans and animals, are often goal-directed. While the goal of these events may not always be immediately visible to the observer, it exists as the purpose that guides the event. There are also events that do not involve animate agents nor have goals, such as natural events.

There is substantial support for the notion that the human cognition uses *"structured representations of events, called event models, to capture information about the Spatio-temporal framework, entities and objects, and other salient features of a situation"* [70]. These event models are a compositional representation of the event and its constituent elements and have partonomies, i.e., hierarchies formed by the "part of" relation. Figure 1 shows an example of a "make sandwich" event and its hierarchical structure. At the lowest level of the hierarchy are elementary actions, such as "carry knife" and "open lid". These actions are part of longer activity units, such as "cut bun", which are in turn part of the "make sandwich" event. This partonomy is analogous to that for objects, which can also be described as a composition of individual parts.

Just like for objects parts, that have boundaries, events have segmentation boundaries at multiple timescales. Event segmentation and the grouping of the segments into a hierarchy is a continuous activity that happens simultaneously at multiple timescales. Evidence from neuroscience experiments [103] indicates that the posterior temporal, parietal cortex, and lateral frontal cortex become active during event boundaries. Some experiments [46] show that event segmentation plays a significant role in the core functions of cognitive control and memory encoding. However, the process of event segmentation does not require conscious attention. Event segmentation can be processed purely with bottom-up, signal-based features extracted from motion and appearance. This conclusion was based on a movie experiment by Zacks *et al.* [103, 104]. A set of participants were shown movie clips of events and were asked to mark the boundaries of events using a clicker as they watched. A different set of participants were asked to do the same but were shown the movie in reverse! The boundaries marked by both groups of participants were remarkably sim-

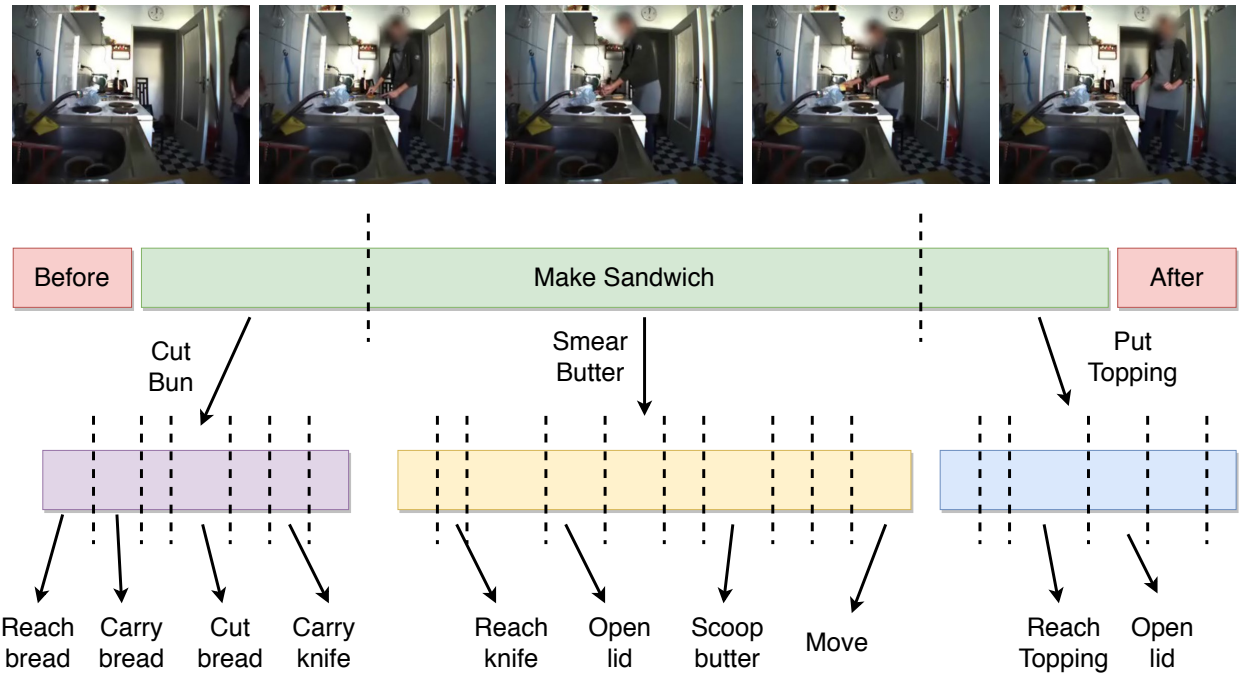


Figure 1: The events hierarchy consists of multiple levels of segmentation. The higher-level event "Make Sandwich" can be segmented into lower-level events "Cut Bun", "Smear Butter", and "Put Topping". Each lower-level event can be further segmented into its constituent events at a lower level. The event "Make Sandwich" can be part of an even higher level event "Have Breakfast". This compositional relationship between events forms "the events hierarchy". Images are from the Breakfast Actions Dataset [44]

ilar. Reversing movie direction prevents viewers from relying on familiar schemas to interpret events, making high-level information - such as goal attaintments - difficult to identify [26]. This is evidence for the claim that high-level information about the event is not necessary to segment events. It also challenges us to design computer vision systems that do not need prior training labels for event segmentation, i.e., to design unsupervised event segmentation algorithms in a continual online manner.

In this chapter, we consider the event segmentation problem, i.e., how do we mark the temporal boundaries (when one event ends, and another begins [107]) of the event *and* spatially locate them in the image. Building the event model and the partonomic hierarchy is a separate process that we do not consider here. First, we will introduce the Event Segmentation Theory (EST) model based on the perceptual prediction model to compute the event boundaries proposed by Zacks *et al.* [66]. Then we present our computer vision solution based on the perceptual prediction model from EST in three progressive versions: temporal segmentation using a perceptual prediction framework, temporal segmentation along with event working models based on attention maps, and finally spatial and temporal localization of events. We end with a discussion of other solutions for event segmentation in the literature and how they relate to ours.

2 The Event Segmentation Theory from Cognitive Science

Event Segmentation Theory (EST), developed by Zacks *et al.* [105] based on cognitive neuroscience experiments, posits that humans maintain a stable representation of "what is happening now" that is updated based on transient increases in the *perceptual prediction error*. This representation of

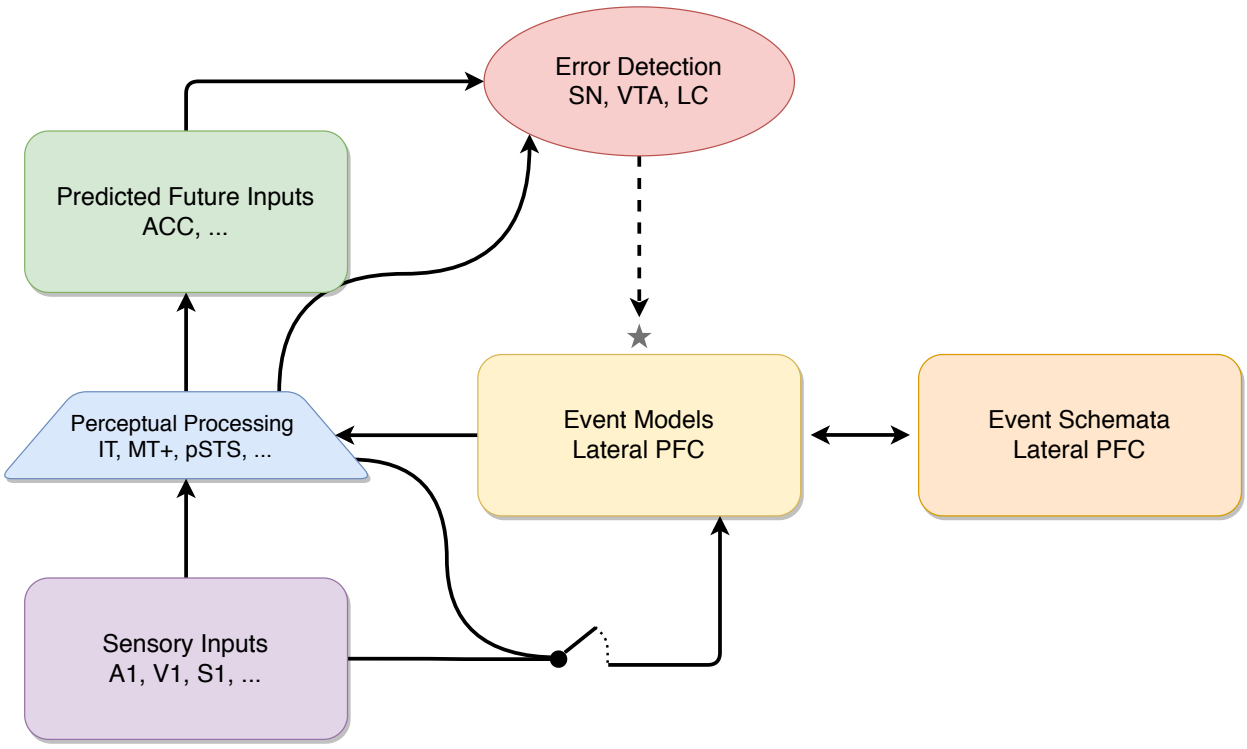


Figure 2: Information flow, according to the Event Segmentation Theory (EST), adapted from [105]. The letter acronyms refer to the different brain regions where these activities are happening, as discovered by cognitive neuroscience experiments. The "Perceptual Processing" unit uses "Sensory Inputs" to extract higher-level features relevant to the predictive task. The extracted features are used to predict future perceptual features and compared to the actual future perceptual features. The "Error Detection" module computes the prediction error, which generates a reset signal (denoted by ⋆) at high prediction errors. The reset signal updates the "Event Model" by accepting input from the "Sensory Inputs" and "Perceptual Processing" blocks.

the current event is called the working *event model* and is used to anticipate the next incoming sensory input. This process of making predictions to anticipate the next input is the key element of this theory; predictions play a central role in building event representations.

Our brains are constantly making predictions of what features our senses expect to see next. When we observe someone cooking, we continuously make predictions. These predictions are made at different temporal granularities. We predict the trajectory of motion at fine temporal scales to anticipate hand location in the next frame. And, at coarse scales to anticipate what utensils could be used next. Indeed, there is a definition of artificial intelligence that puts the ability to make predictions as the central feature [29]. The extent to which an agent can be considered intelligent is determined by the temporal window and spatial extent to which it can predict with precision. Thus, a small insect can make predictions of its immediate surrounding and its near future. Humans, armed with high-level scientific reasoning and logic, can predict events over much larger space and longer temporal windows. The content of the prediction depends on the task at hand. In our current case, it is about making predictions about aspects of a visual event.

The error in the prediction drives the segmentation process to group events into discrete time intervals. As depicted in Figure 2, EST is a continual perceptual process that segments a continuous stream of multimodal sensory input into a discrete and coherent set of events, which can be further segmented to form a hierarchy of events at multiple timescales. The process of event segmentation

does not require conscious attention. Instead, it emerges as a side effect of the ongoing perceptual prediction process.

As sensory inputs are perceived, a perceptual processing unit receives, filters, and encodes the incoming features to useful higher-level representations. Computationally, the perceptual processing unit could be represented by a deep learning stack to process and extract features. A key component of the perceptual processing in EST is that the encoded features are conditioned upon on the representations from the working event model. This conditioning makes the feature representations robust to small variations from one instant to another. The working event model guides the perceptual processing to extract relevant features for the event being observed. Working event models are very specific and limited in capacity; they get updated at event boundaries.

The perceptual processing unit sends the extracted features (conditioned on the working event model) to a prediction unit to anticipate future perceptual features. Any mismatch in the predicted and actual features for the next time instance is monitored continuously and is termed the *prediction error*. The prediction error is a measure of the quality of predictions and hence an indicator of the suitability of the working event model. A transient increase in the prediction error is an indicator of an event boundary, i.e., the current event could be changing. A new event model is needed to make future predictions. Thus, the prediction error works as a gating mechanism to update the working model with an updated representation of the new event. This update of the working event model involves integrating sensory input and prior expectations from long-term memory about the next event. This long-term event memory is referred to as a *event schemata*, which encodes a more stable representation of the event compared to the working event model. Event schemata store sequential information in terms of distinctive physical features about objects and actors, likely next events, and actors' goals, all of which are learned from past occurrences of the event. The changes in the event schemata happen at a slower learning rate than the working event memory.

The prediction quality depends on the accuracy of the working event model in representing what is being perceived. Typically, the working model is a good fit, and the prediction error is low. However, from time to time, the current observations can become less predictable, resulting in an increase in the prediction error and the need for an update in the working event model. In the EST framework, this update is mediated via a gating mechanism, a function of the error signal. When the error signal increases, the working event model is updated based on the sensory signal and the event schemata information until the error reduces. Thus, the overall system operates mostly at a stable state with low prediction errors and transient periods of high errors signaling the event boundaries.

3 Version 1: Single-pass Temporal Segmentation using Prediction

The event segmentation theory offers us a mechanism for temporal segmentation of events, i.e., breaking a video into parts, without the need for training labels and in a continual, online manner. In this section, we demonstrate the power of EST using a simple incarnation of the EST that outperforms more complex state-of-the-art supervised approaches with only a single pass through the video.

Computationally, we implement the EST framework using well-known deep learning components. The perceptual prediction block is implemented as an encoder module, based on convolutional neural networks (CNNs) [50]. A Long Short-Term Memory (LSTM) cell [32] is used to aggregate the features temporally and predict future perceptual features. The LSTM's inner cell structure offers

an ideal implementation alternative for perceptual feature prediction for two reasons. First, the LSTM's recurrent nature allows us to integrate over past frames to predict the future perceptual frames. Second, the hidden state of the LSTM acts as an internal event model that can be used to condition the extracted features on a stable event representation built from previous frames. An adaptive learning mechanism provides a simple and practical method for implementing the gating mechanism used to update the event model based on the perceptual prediction error.

In [1], we focused on building representations for the working event model and did not implement the event schemata block. We begin with a discussion on frame encoding and feature extraction, in Section 3.1, followed by an explanation of how we utilize a recurrent cell to compute a stable representation of previous frames (Section 3.2). We also briefly discuss the role of the reconstruction layer in converting the predicted representation to future perceptual features in section 3.3. We introduce the gating mechanism and adaptive learning functions for boundary detection in Sections 3.5 & 3.6. Algorithm 1 shows the pseudocode for the perceptual prediction framework.

At the core of the approach, illustrated in Figure 3, is a predictive processing platform that encodes a visual input I_t into a higher-level abstraction I'_t using an encoder network. The abstracted feature is used as a prior for predicting the feature I'_{t+1} at time $t+1$. The future prediction module, LSTM, combines the extracted features with a stable representation (hidden state) of the event based on previous frames to predict a future perceptual representation. The reconstruction or decoder network transforms the predicted representation into actual features (same dimensions as I_{t+1}), which are used to detect the event boundaries between successive activities in streaming, input video.

3.1 Feature Extraction and Encoding

We encode the input frame at each time step into abstracted, higher-level visual features and use the features as a basis for perceptual processing rather than the raw input at the pixel level (for reduced network complexity) or higher-level semantics (which require training data in the form of labels). An optimized encoder only extracts features relevant to the task being solved, in this case, prediction. The encoding process entails learning a function $g(I(t), \omega_e)$ that transforms an input frame I_t from pixel space into a higher dimensional feature space I'_t.

The transformation function $g(I(t), \omega_e)$ extracts relevant spatial features into a condensed vector representation using a set of learnable parameters ω_e. In practice, a pre-trained backbone architecture can be used here for encoding raw pixels to useful features. In this work, we use a VGG16 [77] CNN model pre-trained on ImageNet [72] as the transformation function $g(\cdot)$. The pre-trained weights are used to provide good initialization parameters but are further finetuned using the prediction error.

3.2 Recurrent Prediction for Feature Forecasting

Given the perceptual features at time t (I'_t), the next step is the prediction of the perceptual features at time $t+1$. The predicted features are a function of the extracted features I'_t, and an internal, working model of the current event. The internal model modulates the input sensory signal at every frame, similar to the conditioning of the perceptual processing unit on the event model in Section 2. Formally, this can be defined by a generative model $P(I'_{t+1}|\omega_p, I_t)$, where ω_p is the set of hidden parameters characterizing the internal state of the current observed event.

To capture the temporal dependencies among *intra*-event frames and *inter*-event frames, we use a Long Short Term Memory Network (LSTM)[32]. The LSTM model is mathematically expressed in

Algorithm 1 Temporal Event Segmentation Model. The input is an untrimmed/streaming video $\mathbb{I}$, which is a set of frames $\{I_1, ..., I_t, I_{t+1}, ...I_T\}$. The output is a set of event boundaries $\{b_1, b_2, ...b_{T-1}\}$.

Input: Video frames $\{I_1, ..., I_t, I_{t+1}, ...I_T\} \in \mathbb{R}^{TxCxWxH}$
Output: Event boundary values $\mathbb{B} = \{b_1, b_2, ...b_{T-1}\}$

procedure LSTM(h_{t-1}, I'_t)
  $i_t \leftarrow \sigma(W_i I'_t + W_{hi} h_{t-1} + b_i)$ ▷ Input gate
  $f_t \leftarrow \sigma(W_f I'_t + W_{hf} h_{t-1} + b_f)$ ▷ Forget gate
  $o_t \leftarrow \sigma(W_o I'_t + W_{ho} h_{t-1} + b_o)$ ▷ Output gate
  $g_t \leftarrow \phi(W_g I'_t + W_{hg} h_{t-1} + b_g)$
  $m_t \leftarrow f_t \cdot m_{t-1} + i_t \cdot g_t$
  $h_t \leftarrow o_t \cdot \phi(m_t)$
  return h_t
end procedure

procedure GATE($E_P(t)$)
  $P_q(t) = P_q(t-1) + \frac{1}{n}(E_P(t) - P_q(t-1))$ ▷ Running average
  if $\frac{E_P(t)}{P_q(t-1)} > \psi_e$ **then**
    $P_q(t-1) \leftarrow P_q(t)$
    return $True$
  else
    $P_q(t-1) \leftarrow P_q(t)$
    return $False$
  end if
end procedure

procedure SEGMENT(I_t, I_{t+1}, h_{t-1}) ▷ Main segmentation layer
  $I'_t \leftarrow$ ENCODER(I'_t) ▷ Basic CNN encoder
  $I'_{t+1} \leftarrow$ ENCODER(I'_{t+1}) ▷ Basic CNN encoder
  $h_t \leftarrow$ LSTM(h_{t-1}, I'_t)
  $y_{t+1} \leftarrow$ DECODER(h_t) ▷ Single dense layer
  $E_P(t) \leftarrow \sum_{i=1}^{n} \|I'_{t+1} - y'_{t+1}\|^2_{\ell_1}$
  $b_t \leftarrow$ GATE($E_P(t)$)
  return h_t, b_t
end procedure

$h_t \leftarrow 0$
for $\{I_t, I_{t+1}\} \in \{I_1, I_2\}, \{I_2, I_3\}, ..., \{I_{T-1}, I_T\}$ **do**
  $h_t, b_t \leftarrow$ SEGMENT(I_t, I_{t+1}, h_t)
  $\mathbb{B}$.append(b_t)
end for

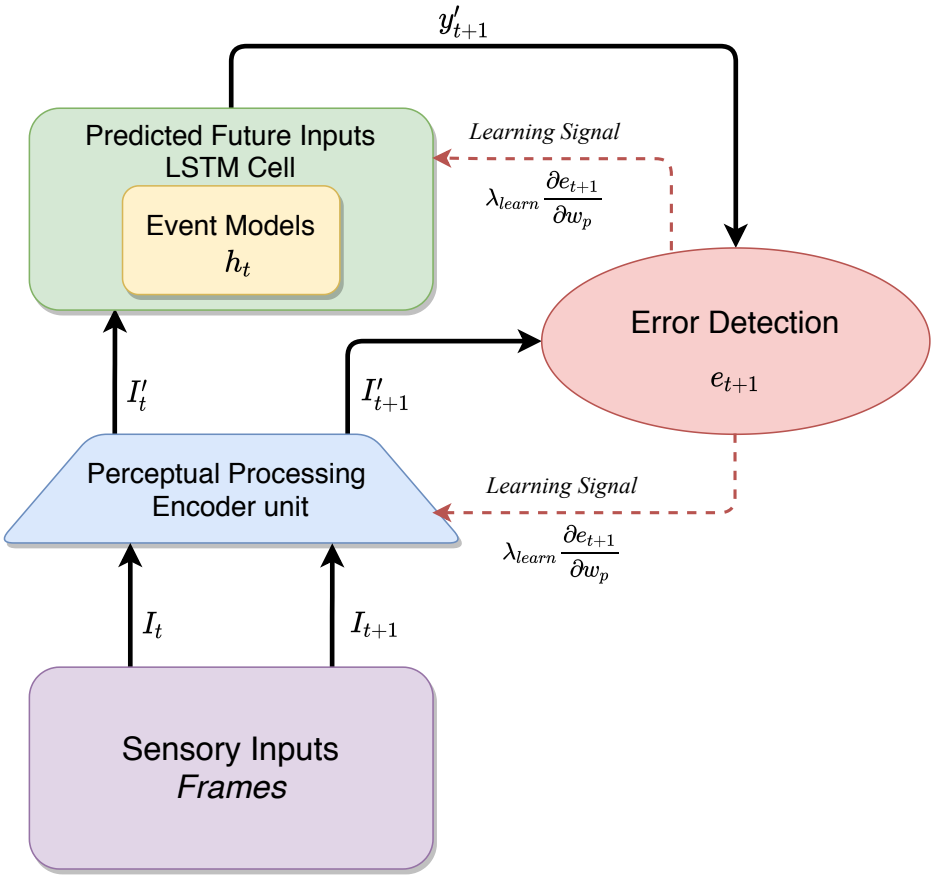


Figure 3: **Version 1 – Single-pass Temporal Segmentation using Perceptual Prediction:** The model architecture is a stripped down version of the blocks proposed for the EST. It consists of four essential components: an encoder network, a predicting unit, a decoding network, an error detection and boundary decision unit.

Algorithm 1, where σ is a non-linear activation function, the dot-operator $(\cdot)$ represents Hadamard (element-wise) multiplication, ϕ is the hyperbolic tangent function ($tanh$) and W_x and b_x represent the trained weights and biases for each of the gates. Collectively, $\{W_{hi}, W_{hf}, W_{ho}, W_{hg}\}$ and their respective biases constitute the learnable parameters ω_p.

The LSTM cell, formally defined in Algorithm 1, consists of three main gates; the input gate i_t, forget gate f_t and output gate o_t. The forget and input (combined with g_t memory layer) gates work together to update the internal event model as specified by their learnable parameters. The output gate processes the input signal by conditioning frame-wise perceptual input on the current event's internal memory. It is worth noting that other recurrent models, such as a Gated Recurrent Unit (GRU), could also be used.

The event state h_t is a representation of the event observed at time instant t and hence is more sensitive to the observed input $I'(t)$ than the event layer, which is more persistent across events. The event layer is a gated layer, which receives input from the encoder and the recurrent event model. However, the inputs to the event layer are modulated by a self-supervised gating signal (Section 3.5), which is indicative of the quality of predictions made by the recurrent model. The gating allows for updating the weights quickly but also maintains a coherent state within the event.

3.3 Feature Reconstruction

The goal of the perceptual processing unit (or rather the reconstruction network) is to reconstruct the predicted feature y'_{t+1} given a source prediction h_t, which maximizes the probability

$$p(y'_{t+1}|h_t) \propto p(h_t|y'_{t+1})\, p(y'_{t+1}) \tag{1}$$

where the first term is the likelihood and the second is the feature prior model. However, we model $\log p(y'_{t+1}|h_t)$ as a log-linear model $f(\cdot)$ conditioned upon the weights of the recurrent model ω_p and the observed feature I'_t and characterized by

$$\log p(y'_{t+1}|h_t) = \sum_{n=1}^{t} f(\omega_p, I'_t) + log\, Z(h_t) \tag{2}$$

where $Z(h_t)$ is a normalization constant that does not depend on the weights ω_p and is used to provide a more concrete estimate of the probability to account for uncertainty. In practice, it is ignored as the predictive learning provides necessary regularization. The reconstruction model completes the generative process for forecasting the feature at time $t+1$ and helps construct the self-supervised learning setting to identify event boundaries.

3.4 Self-Supervised Loss Function

One distinctive property of features within the same event is that they are predictable given previous perceptual inputs and a stable event model. We rely on this property to detect event boundaries. Therefore, we can define a prediction error function that receives, as input, the model's prediction y'_t, and the actual future sensory features I'_t as targets. The resulting error, termed as the perceptual prediction error, is calculated as shown in Equation 3.

$$E_P(t) = \sum_{i=1}^{n} \|I'_t - y'_t\|^2_{\ell_1} \tag{3}$$

The perceptual prediction error is reasonably indicative of the relevance of the recurrent model's internal state to the actual event being observed. When the event changes by crossing an event boundary, the internal model becomes unusable, leading to incorrect predictions. The prediction quality increases after updating the internal model with a newer representation capable of predicting future features. Figure 4 illustrates an instance of this effect. The minimization of the perceptual prediction error serves as the objective function for training the network.

3.5 Error Gating

According to the Event Segmentation Theory, the perceptual features can become highly unpredictable at event boundaries since the working event model will need to be updated to process the new event. For example, in Figure 4, we can see that the visual representation of the features learned by the encoder network for the activities *take bowl* and *crack eggs* are closer together than the features between the activities *take bowl* and *spoon flour*. At event boundaries, the diverging feature space becomes inexplicable by the recurrent internal model, causing a transient increase in the perceptual prediction error. We see that the perceptual prediction error (second plot from the bottom) matches well with the ground truth segmentation (second from the top) for

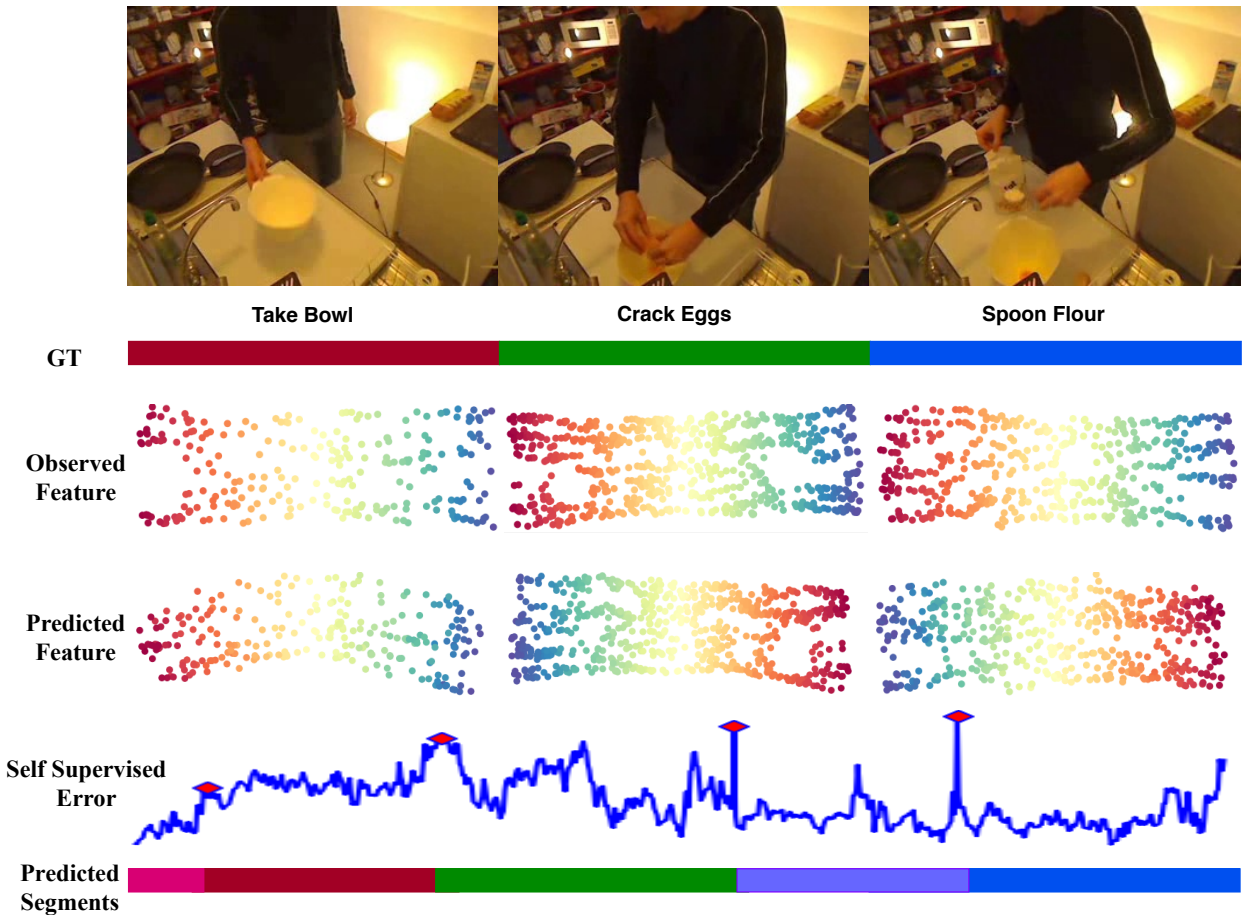


Figure 4: A visualization of the predictive learning process in the Event Segmentation Theory is shown here. The current sensory inputs are abstracted into features of lower variability that are conducive to prediction. A working event model is constructed and is used to continuously predict the features observed at the next time step. Predictions are compared continuously to observed features, and the resulting prediction error serves as an indicator of the suitability of the event model. A gating mechanism, a function of the prediction error, modulates the learning process, and provides cues for event segmentation. ♦ refers to predicted event boundaries. The features were visualized using T-SNE [61] for presentation. Reproduced with permission from [1].

the video "Make Pancake". As illustrated, the error rates are higher at the event boundaries and lower within the "in-event" frames.

To model the prediction error and identify event boundaries, we can use a low pass filter to maintain a running average of the perceptual prediction error made over the last n input time steps. An n value of 5 is used here based on the average human perception response time ($\approx 200ms$) [87]. We maintain a running average of the prediction error, called the prediction quality and given by

$$P_q(t) = P_q(t-1) + \frac{1}{n}(E_P(t) - P_q(t-1)) \tag{4}$$

where P_q is the prediction quality and $E_P(t)$ is the prediction error from Equation 3. A gating signal, $G(t)$, is triggered when the current prediction error exceeds the average prediction quality metric

by at least 50%. Formally, we can define the gating function as

$$G(t) = \begin{cases} 1, & \frac{E_P(t)}{P_q(t-1)} > \psi_e \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

where $E_P(t)$ is the perceptual prediction error at time t, $G(t)$ is the value of the gating signal at time t, $P_q(t-1)$ is the average prediction quality metric at time t and Ψ_e is the prediction error threshold for boundary detection. For optimal prediction, the perceptual prediction error would be very high at the event boundary frames and very low at all within-event frames. Ψ_e is a hyperparameter which can be used to set the timescale at which we will be detecting event boundaries.

3.6 Adaptive Learning for Plasticity

Learning a robust event representation is at the core of event segmentation approaches using predictive learning. An event representation is considered to be robust when the perceptual prediction error is low for *intra*-event frames and high at *inter*-event frames. If the learned event representations *overfit* to intra-event observations, then minor perturbations in the raw pixel space can increase the prediction error. This would negate the underlying assumption that transient errors indicate a change in the observed event. Additionally, the event representation must be stable across events with varying temporal duration to avoid catastrophic forgetting, i.e., the condition where the predictions do not capture intra-event variations in long event sequences. Hence, it is necessary to ensure that the model does not overfit to short-term perceptual features while maintaining a robust event representation representing the *entire* event.

To allow for some plasticity and avoid catastrophic forgetting in the network, we use adaptive learning. Adaptive learning is similar to the learning rate schedule, a commonly used technique for training deep neural networks. However, instead of using predetermined intervals for changing the learning rates, we use the prediction error to modulate the learning rate. The learning rate can be tuned to control the propagation of the error back to the learnable parameters.

For example, when the perceptual prediction rate is lower than the average prediction rate, the predictor model is considered a good, stable representation of the current event. Propagating the prediction error when there is a good representation of the event can lead to overfitting of the predictor model to that particular event and does not help generalize. Hence, lower learning rates are used for time steps when there are negligible prediction error and a relatively higher (by a magnitude of 100) for higher prediction error.

Intuitively, this adaptive learning rate allows the model to adapt much quicker to new events (at event boundaries where there are likely to be higher errors) and learn to maintain the internal representation for within-event frames. Formally, the learning rate is defined as the result of the adaptive learning rule described as a function of the perceptual prediction error defined in Section 3.4 and is defined as

$$\lambda_{learn} = \begin{cases} \Delta_t^- \lambda_{init}, & E_P(t) > \mu_e \\ \Delta_t^+ \lambda_{init}, & E_P(t) < \mu_e \\ \lambda_{init}, & otherwise \end{cases} \tag{6}$$

where Δ_t^-, Δ_t^+ and λ_{init} refer to the scaling of the learning rate in the negative direction, positive direction and the initial learning rate respectively and $\mu_e = \frac{1}{t_2 - t_1} \int_{t_1}^{t_2} E_P \, dE_P$. The learning rate is

adjusted based on the quality of the predictions characterized by the perceptual prediction error between a temporal sequence between times t_1 and t_2, typically defined by the gating signal.

3.7 Results

3.7.1 Datasets

We evaluate and analyze the performance of the perceptual prediction framework on three large, publicly available datasets - Breakfast Actions [44], INRIA Instructional Videos dataset[6] and the 50 Salads dataset [85]. Each dataset offers a different challenge to the approach allowing us to evaluate its performance on various challenging conditions.

Breakfast Actions Dataset is a large collection of 1,712 videos of 10 breakfast activities performed by 52 actors. Each activity consists of multiple sub-activities that possess visual and temporal variations according to the subject's preferences and style. Varying qualities of visual data and complexities such as occlusions and viewpoints increase the complexity of the temporal segmentation task.

INRIA Instructional Videos Dataset contains 150 videos of 5 different activities collected from YouTube. Each of the videos is, on average, 2 minutes long and has around 47 sub-activities. A "background" class denotes a sequence where there does not exist a clear sub-activity that is visually discriminable. This offers a considerable challenge for approaches that are not explicitly trained for such visual features.

50 Salads Dataset is a multimodal dataset collected in the cooking domain. The dataset contains over four (4) hours of annotated data of 25 people preparing two mixed salads each. It provides data in different modalities such as RGB frames, depth maps, and accelerometer data for devices attached to different items such as knives, spoons, and bottles, to name a few. The annotations of activities are provided at different levels of granularities - high, low, and eval. We use the "eval" granularity following evaluation protocols in prior works [48, 67].

3.7.2 Evaluation Metrics

We use two commonly used evaluation metrics for analyzing the performance of this approach. We use the same evaluation protocol and code as in [6, 74]. We utilize the Hungarian matching algorithm to obtain the one-to-one mappings between the predicted segments and the ground truth to evaluate the performance due to the unsupervised nature of the predictive approach. We use the mean over frames (MoF) to evaluate the network's ability to temporally localize the sub-activities. We evaluate the divergence of the predicted segments from the ground truth segmentation using the Jaccard index (Intersection over Union or IoU). We also use the F1 score to evaluate the quality of the temporal segmentation. The evaluation protocol for the recognition task in Section 3.7.4.1 is the unit level accuracy for the 48 classes as seen in Table 3 from [44] and compared in [44, 3, 83, 35].

3.7.3 Ablative Studies

We evaluate different variations of our framework to compare the effectiveness of each component. We varied the prediction history n, and the prediction error threshold Ψ. Increasing frame window tends to merge frames and smaller clusters near the event boundaries to the prior activity class due to a transient increase in error. This results in a higher IoU and a lower MoF. Low error threshold results in over-segmentation as boundary detection becomes sensitive to small changes.

The number of predicted clusters decreases as the window size and threshold increases. We also trained four models (Figure 5), with different predictor units. We trained two recurrent neural networks (RNN) as the predictor units with and without adaptive learning (AL) described in Section 3.6 indicated as *RNN + No AL* and *RNN + AL*, respectively. We also trained LSTM without adaptive learning (*LSTM + No AL*) to compare against our main model (*LSTM + AL*). We use RNNs as a possible alternative due to the short-term future predictions (1 frame ahead) required.

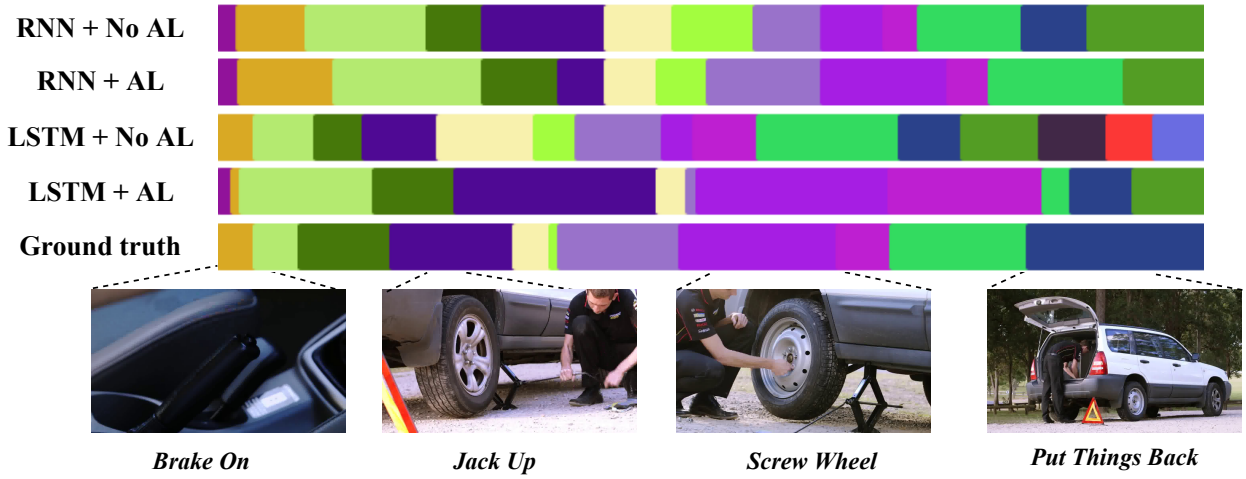


Figure 5: **Ablative Studies**: Illustrative comparison of variations in the model architecture, using RNNs and LSTMs with and without adaptive learning on the INRIA Instructional Videos Dataset on a video with ground truth *Change Tire*. It can be seen that complex visual scenes with activities of shorter duration pose a significant challenge to the framework and cause fragmentation and over-segmentation. However, the use of adaptive learning helps alleviate this to some extent. Note: Temporal segmentation timelines are shown without the background class for better visualization.

3.7.4 Quantitative Evaluation

Breakfast Actions Dataset We evaluate the performance of our full model *LSTM + AL* on the breakfast actions dataset and compare against fully supervised, weakly supervised, and unsupervised approaches. We show the performance of the SVM[44] approach to highlight the importance of temporal modeling. As shown in Table 1, the perceptual prediction approach outperformed all other unsupervised and all weakly supervised approaches and some fully supervised approaches.

It should be noted that the other unsupervised approach [74] requires the number of clusters (from ground truth) to achieve the reported performance. In contrast, our approach does not require such knowledge and is done in a streaming fashion. Additionally, the weakly supervised methods [35, 68, 13] require both the number of actions and an ordered list of sub-activities as input. ECTC [35] is based on discriminative clustering, while OCDC [9] and Fine2Coarse [68] are also RNN-based methods.

50 Salads Dataset We also evaluate our approach on the 50 Salads dataset, using only the visual features as input. We report the Mean of Frames (MoF) metric for a fair comparison. As can be seen from Table 2, the predictive approach significantly outperforms the other unsupervised approach, improving by 6.6% on the MoF metric. We also show the performance of the frame-based classification approaches VGG and IDT [48] to demonstrate the effectiveness of temporal modeling.

It should be noted that the fully supervised approaches require significantly more training data - both in the form of labels as well as training epochs. Additionally, the TCN approach [67] uses

Supervision	Approach	MoF	IoU
Full	SVM [44]	15.8	-
	HTK(64)[45]	56.3	-
	ED-TCN[67]	43.3	42.0
	TCFPN[13]	52.0	**54.9**
	GRU[69]	**60.6**	-
Weak	OCDC[9]	8.9	23.4
	ECTC[35]	27.7	-
	Fine2Coarse[68]	33.3	**47.3**
	TCFPN + ISBA[13]	**38.4**	40.6
None	KNN+GMM[74]	34.6	**47.1**
	Ours (LSTM + AL)	**42.9**	46.9

Table 1: Segmentation Results on the Breakfast Action dataset. MoF refers to the Mean over Frames metric and IoU is the Intersection over Union metric.

Supervision	Approach	MoF
Full	VGG**[48]	7.6%
	IDT**[48]	54.3%
	S-CNN + LSTM[48]	66.6%
	TDRN[51]	68.1%
	ST-CNN + Seg[48]	72.0%
	TCN[67]	**73.4**%
None	LSTM + KNN[8]	54.0%
	Ours (LSTM + AL)	**60.6%**

Table 2: Segmentation Results on the 50 Salads dataset, at granularity *'Eval'*. **Models were intentionally reported without temporal constraints for ablative studies.

accelerometer data to achieve the state-of-the-art performance of 73.4%

INRIA Instructional Videos Dataset: Finally, we evaluate our approach on the INRIA Instructional Videos dataset, which posed a significant challenge in the form of high amounts of background (noise) data. We report the F1 score for a fair comparison to the other state-of-the-art approaches. As can be seen from Table 3, the predictive model outperforms the other unsupervised approach [74] by 7.5%, the weakly supervised approach [9] by 7.9% and has a competitive performance to the fully supervised approaches[62, 6, 74].

We also evaluate the performance of the models with and without adaptive learning. The effectiveness of LSTMs in capturing long-term temporal dependencies is significant, as can be seen in Table 3, primarily due to the long duration of activities in the dataset. Additionally, adaptive learning has a significant improvement in the segmentation framework, improving the performance by 9% and 11% for the RNN-based model and the LSTM-based model, respectively, indicating reduced overfitting of the model to the visual data.

3.7.4.1 Improved Features for Action Recognition To evaluate the network's ability to learn highly discriminative features for recognition, we evaluated the performance of the predictive approach in a recognition task. We pre-train the model for temporal segmentation using the Breakfast Actions dataset and use the hidden layer of the LSTM as input to a fully connected layer to minimize a cross-entropy loss for training. We also trained another network with the same structure - VGG16 + LSTM - without pretraining to show the usefulness of the learned features

Supervision	Approach	F1
Full	HMM + Text [62]	22.9%
	Discriminative Clustering[6]	41.4%
	KNN+GMM[74] + GT	**69.2%**
Weak	OCDC + Text Features [9]	28.9%
	OCDC [9]	**31.8%**
None	KNN+GMM[74]	32.2%
	Ours (RNN + No AL)	25.9%
	Ours (RNN + AL)	29.4%
	Ours (LSTM + No AL)	36.4%
	Ours (LSTM + AL)	**39.7%**

Table 3: Segmentation Results on the INRIA Instructional Videos dataset. We report F1 score for fair comparison.

using self-supervision.

Approach	Precision
HCF + HMM [44]	14.90%
HCF + CFG + HMM [44]	31.8%
RNN + ECTC [35]	35.6%
RNN + ECTC (Cosine) [35]	36.7%
HCF + Pattern Theory [83]	38.6%
HCF + Pattern Theory + ConceptNet[3]	42.9%
VGG16 + LSTM	33.54%
VGG16 + LSTM + Predictive Features(AL)	37.87%

Table 4: Activity recognition results on Breakfast Actions dataset. HCF and AL refer to handcrafted features and Adaptive Learning, respectively.

As can be seen from Table 4, using self-supervision to pre-train the network before the recognition task improves the recognition performance of the network and has yielded comparable performance to the other state-of-the-art approaches. It improves the recognition accuracy by 4.3% over the network without predictive pretraining.

3.7.5 Qualitative Evaluation

Through the predictive, self-supervised framework, we can learn the sequence of visual features in streaming video. We visualize the model's segmentation performance on the Breakfast Actions Dataset in Figure 6. It can be seen that the predictive framework has high temporal coherence and does not suffer from over-segmentation, especially when the segments are long. Long activity sequences allow the model to learn from observation by providing more "*intra*-event" samples. Additionally, weakly supervised approaches like OCDC[9] and ECTC[35] suffer from over-segmentation and intra-class fragmentation. This could arguably be attributed to the fact that they tend to enforce semantics in the form of a weak ordering of activities in the video regardless of the changes in visual features. Fully supervised approaches - such as HTK[45] - perform better, especially due to their ability to assign semantics to visual features. However, they are also affected by unbalanced data and dataset shift, as can be seen in Figure 6 where the background class was segmented into other classes.

We also qualitatively evaluate the impact of adaptive learning and long term temporal memory in

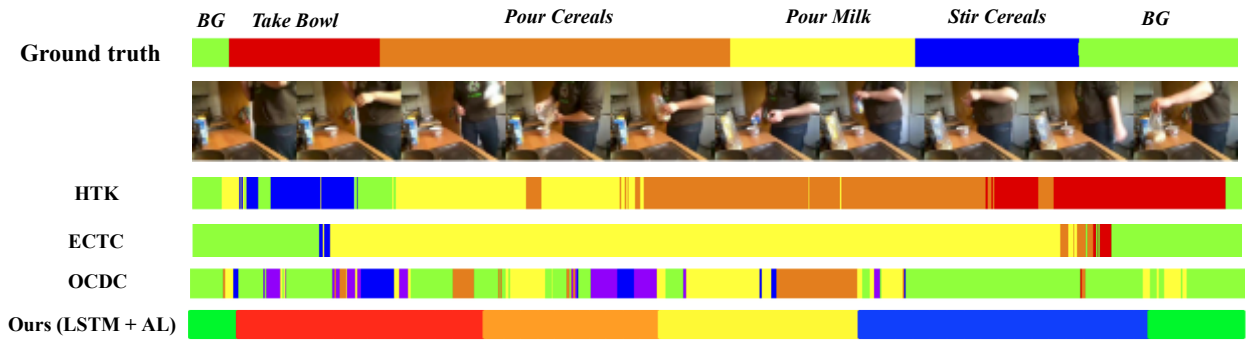


Figure 6: Illustration of the segmentation performance of main model on the Breakfast Actions Dataset on a video with ground truth *Make Cereals*. The predictive approach does not show the tendency to over-segment and provides coherent segmentation. The approach, however, shows a tendency to take longer to detect boundaries for visually similar activities.

Figure 5, where the performance of the alternative methods described in Section 3.7.4. It can be seen that the use of adaptive learning during training prevents the model from overfitting to any single class's intra-event frames and helps generalize to other classes regardless of the amount of training data. It is not to say that the problem of unbalanced data is alleviated, but adaptive learning *does* help to some extent.

4 Version 2: Segmentation using Attention-based Event Models

The simple perceptual prediction framework, introduced in Section 3, uses a global representation of input video frames for perceptual prediction. It does not *attend* or rather focus on relevant spatial areas for both perception and prediction and as such may not capture *fine-grained* event representations. This is essentially strengthening the pathway proposed in the EST (Figure 2) for the event models to influence the perceptual processing of the current sensory input. In our earlier model, this pathway was weakly implemented via the memory structure in the LSTM. In this section, we show that the framework can be enhanced with the idea of *spatial attention* to help focus on relevant spatial regions for a more fine-grained segmentation approach that can handle very long video sequences, *without significant training needs*. The full architecture is shown in detail in Figure 7 and formally expressed in Algorithm 2. The spatial attention mechanism helps the network to learn an attention function between the internal current event model in the recurrent memory and the input at each time step. The resulting attention map can be used to localize the event within each processed frame spatially.

4.1 Feature Extraction

The feature extraction process in this architecture differs from the previous perceptual framework. We need to maintain the spatial arrangement of feature vectors for the attention mechanism to work. In this architecture, the encoder outputs a grid of feature vectors with the same spatial resolution as the resulting attention map. In other words, the attention mechanism dictates the value of attention assigned to each of the encoded feature vectors. We only use convolution operations in the encoder since kernels (weights) maintain feature maps' spatial configuration.

The raw input images are transformed from pixel space into a higher-level feature space using an encoder (CNN) model. This encoded feature representation allows the network to extract features of higher importance to the task being learned. The encoder network transforms an input image

Algorithm 2 Temporal Event Segmentation Model with Attention-based Spatial Event Localization. The input is an untrimmed/streaming video $\mathbb{I}$, which is a set of frames $\{I_1,...,I_t,I_{t+1},...I_T\}$. The output is a set of event boundaries $\{b_1,b_2,...b_{T-1}\}$.

Input: Video frames $\{I_1,...,I_t,I_{t+1},...I_T\} \in \mathbb{R}^{TxCxWxH}$
Output: Event boundary values $\mathbb{B} = \{b_1,b_2,...b_{T-1}\}$

procedure ATTENTION(I'_t, h_{t-1}) ▷ Attention block
  $a_t \leftarrow linear(tanh(linear(h_{t-1}) + linear(I'_t)))$
  $A_t \leftarrow softmax(a_t)$
  $I''_t \leftarrow A_t \odot I'_t$
  return I''_t
end procedure

procedure SEGMENT($I_t, I_{t+1}, h_{t-1}, y_{t-1}$) ▷ Main segmentation layer
  $I'_t \leftarrow$ ENCODER(I_t) ▷ Basic CNN encoder
  $I'_{t+1} \leftarrow$ ENCODER(I_{t+1}) ▷ Basic CNN encoder
  $I''_t \leftarrow$ ATTENTION(I'_t, h_{t-1})
  $h_t \leftarrow$ LSTM($h_{t-1}, linear(concat(I''_t, y_{t-1}))$)
  $y_t \leftarrow$ DECODER(h_t) ▷ Single dense layer
  $e_t \leftarrow \|(I'_{t+1} - y_t)^{\odot 2} \odot (I'_{t+1} - I'_t)^{\odot 2}\|^2$ ▷ Motion weighted loss
  $b_t \leftarrow$ GATE(e_t)
  return h_t, b_t, y_t
end procedure

$h_t \leftarrow 0$
$y_{t-1} \leftarrow 0$
for $\{I_t, I_{t+1}\} \in \{I_1, I_2\}, \{I_2, I_3\}, ..., \{I_{T-1}, I_T\}$ **do**
  $h_t, b_t, y_t \leftarrow$ SEGMENT($I_t, I_{t+1}, h_t, y_{t-1}$)
  $\mathbb{B}.append(b_t)$
end for

with dimensions $W \times H \times D$ to output features with dimensions $N \times N \times M$, where $N \times N$ are the spatial dimensions and M is the feature vector length.

4.2 Attention Unit

Attention units have successfully been applied in supervised tasks such as image captioning [101] and for various natural language processing tasks such as translation and language modeling [94, 7, 58, 12, 102]. Attention, in auto-regressive language models, is used to expose different temporal - or spatial - segments of the input to the decoding recurrent cell at every time step using fully supervised architectures. We use attention in a slightly different form, where the LSTM is decoded only once (per input frame) to predict future features and uses attention-weighted input to do so. Unlike [101, 94, 7, 58, 12, 102], the attention weights and biases are trained using unsupervised loss functions.

In this framework, we utilize Bahdanau attention [7] to spatially localize the event in each processed frame. The attention unit receives as an input the encoded features and outputs a set of attention weights (A_t) with dimensions $N \times N \times 1$. The hidden feature vectors (h_{t-1}) from the prediction layer of the previous time step are used to calculate the output attention weights (expressed visually in

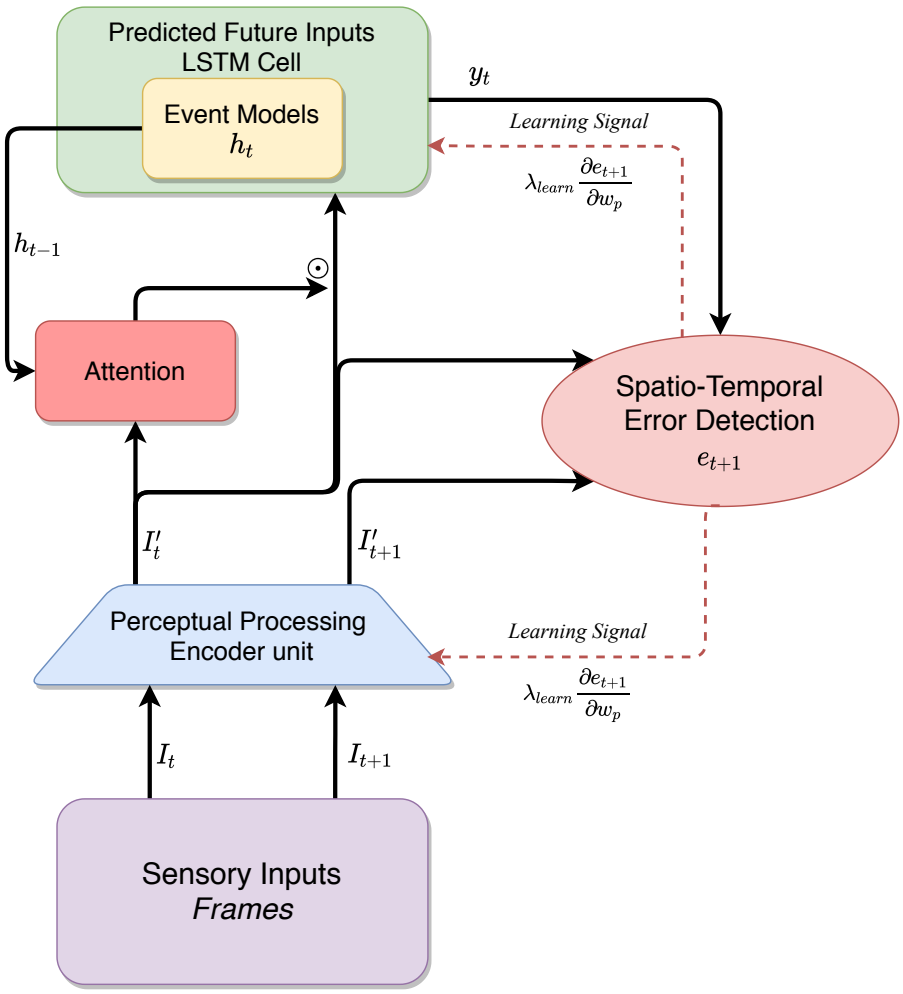


Figure 7: **Version 2: Temporal Segmentation using Attention-based Event Models:** The self-learning perceptual prediction algorithm's architecture enhanced with an attention mechanism that offers a stronger way for the event model to influence the current sensory input. Input frames from each time instant are encoded into high-level features using a deep-learning stack, followed by an attention overlay based on inputs from previous time instants, which is input to an LSTM. The training loss is composed based on the predicted and computed features from the current and next frames.

Figure 7), as

$$A_t = \gamma(FC(tanh(FC(h_{t-1}) + FC(I'_t)))) \tag{7}$$

where FC represents a single fully connected neural network layer and γ represents a softmax function. The weights (A_t) are then multiplied by the encoded input feature vectors (I'_t) to generate the masked feature vectors (I''_t). Some visualization of the resulting spatial attention masks are shown in Figure 8. The attention mask is extracted from A_t, linearly scaled and resized, then overlayed on the raw input image (I_t).

4.3 Motion Weighted Loss Function

The prediction loss we discussed in Section 3.4 applies an L2 loss function over the prediction of the whole frame. In this section, we introduce a motion weighted loss function which extracts motion related features from the feature vectors. The motion weighted loss acts upon the encoded frames in feature space rather than raw pixel space and is calculated using a continuous mask

applied to the prediction loss. This modification aims to increase the prediction loss at moving features while reducing the loss of static/background features. Figure 8 shows a comparison between prediction and motion weighted loss for an event with the bird moving in and out of a nest. The motion weighted loss is defined formally as

$$e_t = \|(I'_{t+1} - y_t)^{\odot 2} \odot (I'_{t+1} - I'_t)^{\odot 2}\|^2 \tag{8}$$

where $\odot$ denotes Hadamard (element-wise) operation. Note that the motion-weighted vector (the second term) is computed at the *feature level* and not at the pixel level and hence is more robust to minor changes due to sensor noise.

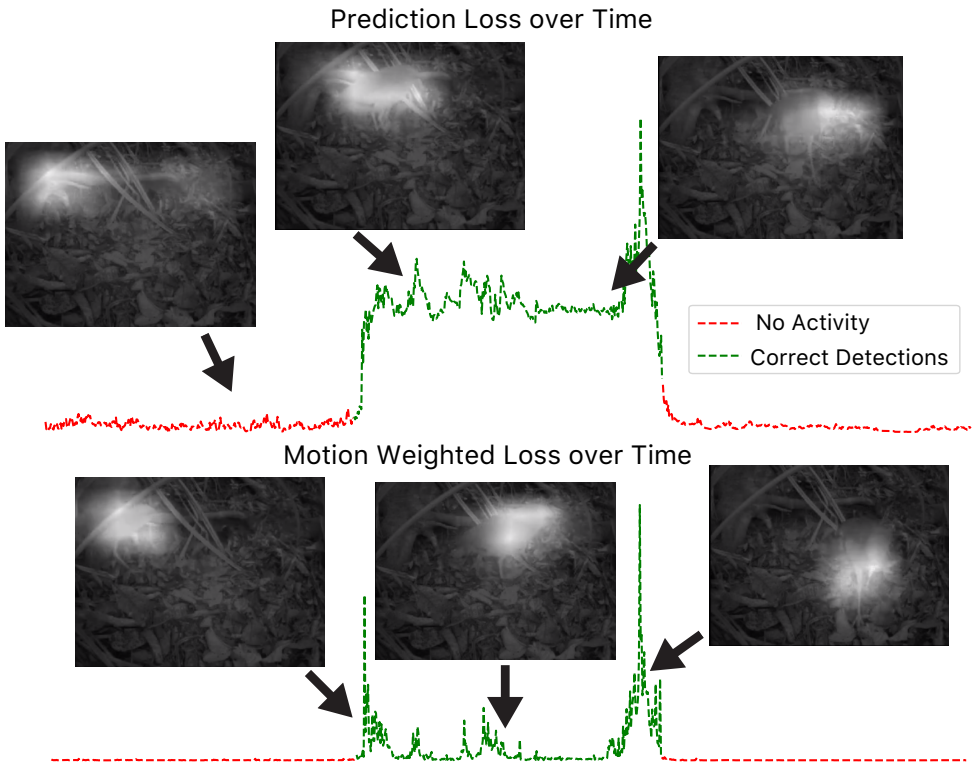


Figure 8: Plots of the prediction and motion-weighted losses before, during, and after an activity: (top) feature prediction loss over the frames, (bottom) motion weighted feature prediction loss over the frames. Errors for some selected frames are shown for both plots, overlaid with the corresponding attention map.

4.4 Results

In this section, we will look into the results of processing videos spanning over several days to flag spatial locations and temporal boundaries of events. Unlike other datasets, events in such extended videos only take place over a few frames, while the rest of the video contains no significant events or motion (background activities). We modify the perceptual prediction framework to include an attention mechanism for spatial localization, and we also mask the prediction loss function to extract motion related features. We begin by defining the wildlife extended video dataset used for testing the modified approach, followed by an explanation of the

evaluation metrics used to quantify performance. We discuss the model variations evaluated and conclude by presenting quantitative and qualitative results.

4.4.1 Dataset

We analyze the performance of our model on a wildlife monitoring dataset. The dataset consists of 10 days (254 hours) of continuous monitoring of a nest of the Kagu, a flightless bird of New Caledonia. The labels include four unique bird activities, {feeding the chick, incubation/brooding, nest building while sitting on the nest, nest building around the nest}. Start and end times for each instance of these activities are provided with the annotations. We modified the annotations to include the "walk-in" and "walk-out" events representing the transitioning events from an empty nest to incubation and vice versa. Our approach can flag the nest building (on and around the nest), feeding the chick, walk in and out events. Other events based on climate, time of day, lighting conditions are ignored by our segmentation network. Figure 9 shows a selection of images from the dataset.

Figure 9: Samples of images from the Kagu bird wildlife monitoring dataset

4.4.2 Evaluation Metrics

We provide quantitative results, in the form of receiver operating characteristic (ROC) charts, for both frame level (Figure 10) and activity level (Figures 11 & 12) event segmentation. Frame window size (ϕ) is defined as the maximum joining window size between events; a high ϕ value can cause separate detected events to merge, which decreases the overall performance.

4.4.2.1 Frame Level The recall value in frame-level ROC is calculated as the ratio of true positive frames (event present) to the number of positive frames in the annotations dataset, while the false positive rate is expressed as the ratio of the false positive frames to the total number of negative frames (event not present) in the annotation dataset. The threshold value (ψ) is varied to obtain a single ROC line while varying the frame window size (ϕ) results in a different ROC line.

4.4.2.2 Activity Level The Hungarian matching (Munkres assignment) algorithm is utilized to achieve a one-to-one mapping between the ground truth labeled events and the detected events. Recall is defined as the ratio of the number of correctly detected events (overlapping frames) to the total number of ground-truth events. For the activity level ROC chart, the recall values are plotted against the false positive rate per minute, defined as the ratio of the total number of false-positive detected events to the total duration of the dataset in minutes. The false-positive rate per minute evaluation metric is also used in the ActEV TRECVID challenge [4] to quantify activity detection systems' performance. Frame window size value (ϕ) is varied to obtain a single ROC line while varying the threshold value (ψ) results in a different ROC line.

4.4.3 Ablative Studies

Different variations of our framework have been evaluated to quantify the effect of individual components on the overall performance. In our experiments, we tested the base model, which trains the perceptual prediction framework - including an attention unit - using the prediction loss function for backpropagation of the error signal. We refer to the base model as *LSTM+ATTN*. We also experimented with the effect of removing the attention unit from the model architecture, on the overall segmentation performance; results of this variation are reported under the model name *LSTM*. Further testing includes using the motion weighted loss for backpropagation of the error signal. We refer to the motion weighted model as *LSTM+ATTN+MW*. Each of the models has been tested extensively; results are reported in Sections 4.4.4 & 4.4.5, as well as visually expressed in Figures 10, 11, 12, 13 & 14.

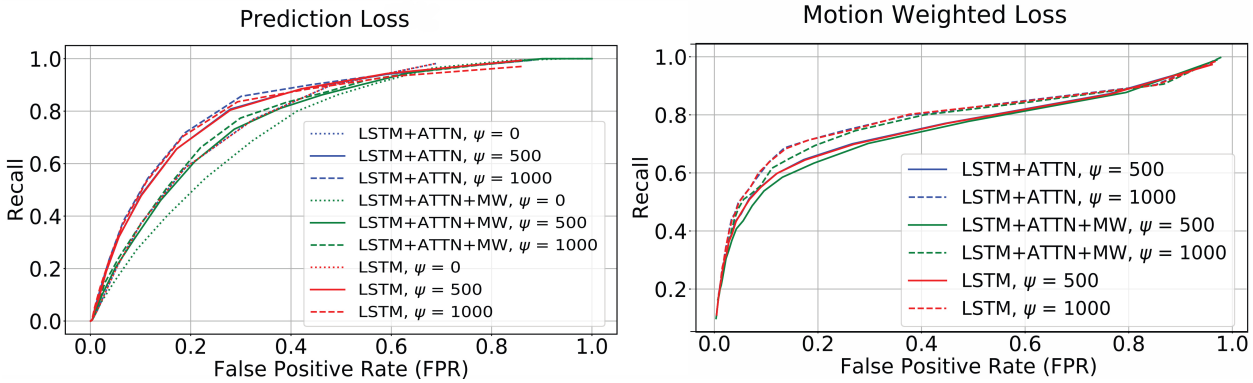


Figure 10: Frame-level event segmentation ROCs when activities are detected based on simple thresholding of the prediction and motion weighted loss signals. Plots are shown for different ablation studies.

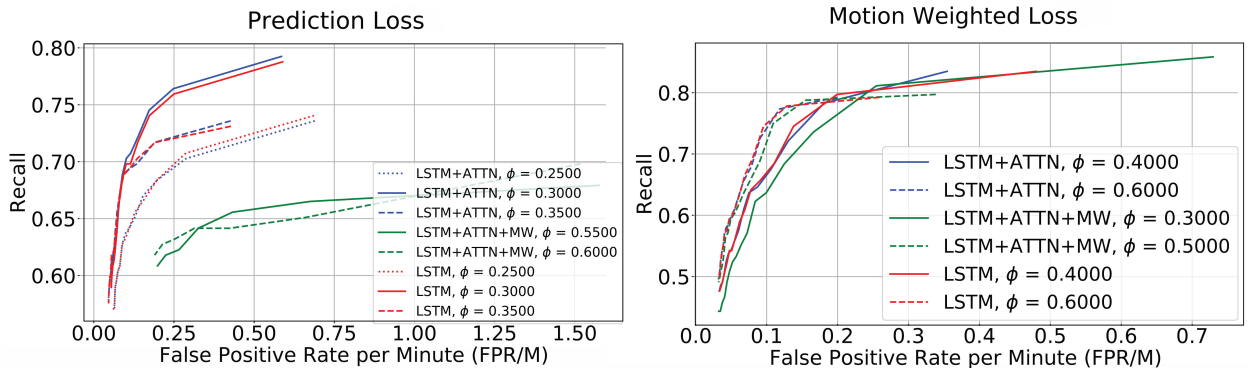


Figure 11: Activity-level event segmentation ROCs when activities are detected based on simple thresholding of the prediction and motion weighted loss signals. Plots are shown for different ablation studies.

4.4.4 Quantitative Evaluation

We trained three different models, *LSTM, LSTM+ATTN,* and *LSTM+ATTN+MW,* for frame level and activity level event segmentation. Simple and adaptive gating functions were applied to prediction and motion weighted loss signals (Section 4.3) for frame level and activity level experiments. We present ROC curves for each model in Figures 10, 11 & 12 by varying parameters such as the threshold value ψ and the frame window size ϕ.

It is to be noted that thresholding a loss signal does not necessarily imply that the model was

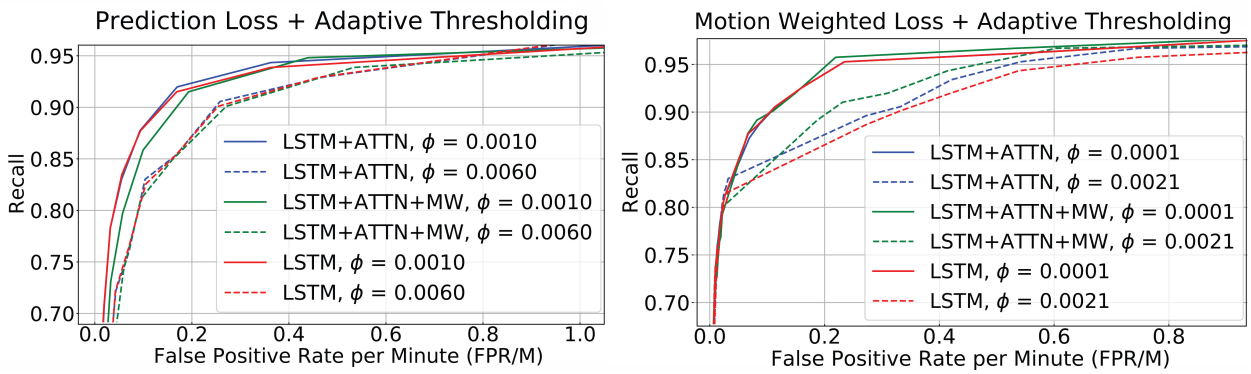


Figure 12: Activity-level event segmentation ROCs when activities are detected based on adaptive thresholding of the prediction and motion weighted loss signals. Plots are shown for different ablation studies.

trained to minimize this particular signal. In other words, the loss functions used for backpropagating the error to the models' learnable parameters are identified only in the model name (Section 4.4.3); however, thresholding experiments have been conducted on different types of loss signals, regardless of the backpropagating loss function used for training.

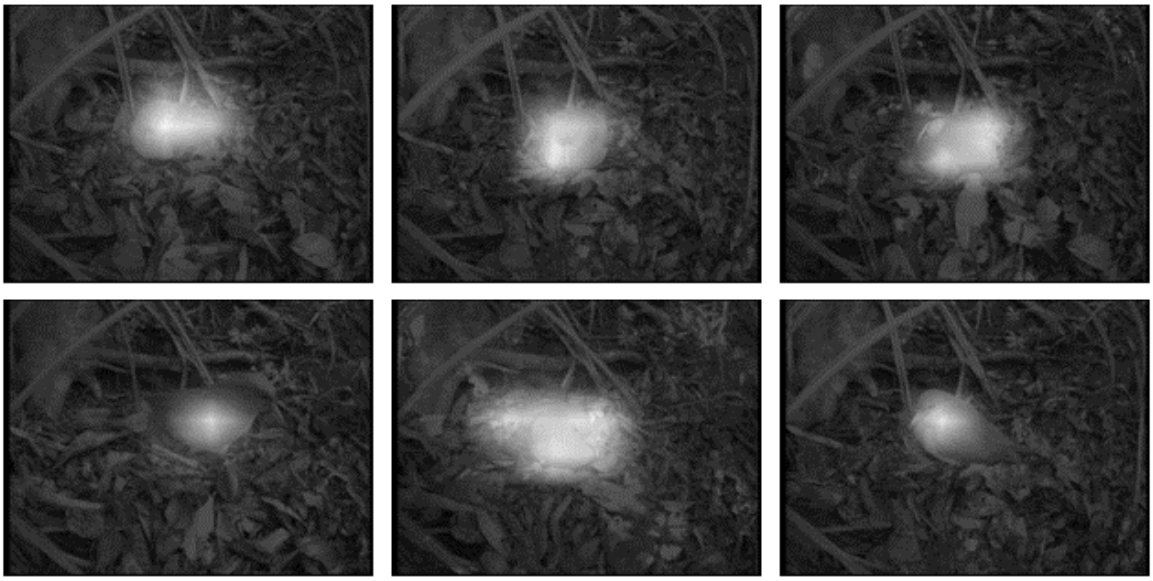
The best performing model, for frame level segmentation, (*LSTM+ATTN*, $\psi = 1000$) is capable of achieving {40%, 60%, 80%} frame recall value at {5%, 10%, 20%} frame false positive rate respectively. Activity level segmentation can recall {80%, 90%, 95%} of the activities at {0.02, 0.1, 0.2} activity false positive rate per minute, respectively, for the model (*LSTM+ATTN*, $\phi = 0.0021$) as presented in Figure 12. A 0.02 false positive activity rate per minute can also be interpreted as one false activity detection every 50 minutes of training (for a recall of 80% of the groundtruth activities).

Comparing the results shown in Figures 11 & 12 indicates a significant increase in the overall performance when using an adaptive threshold for loss signal gating. The efficacy of adaptive thresholding is evident when applied to activity level event segmentation. Results have also shown that the model can effectively generate attention maps (Section 4.4.5) without impacting the segmentation performance.

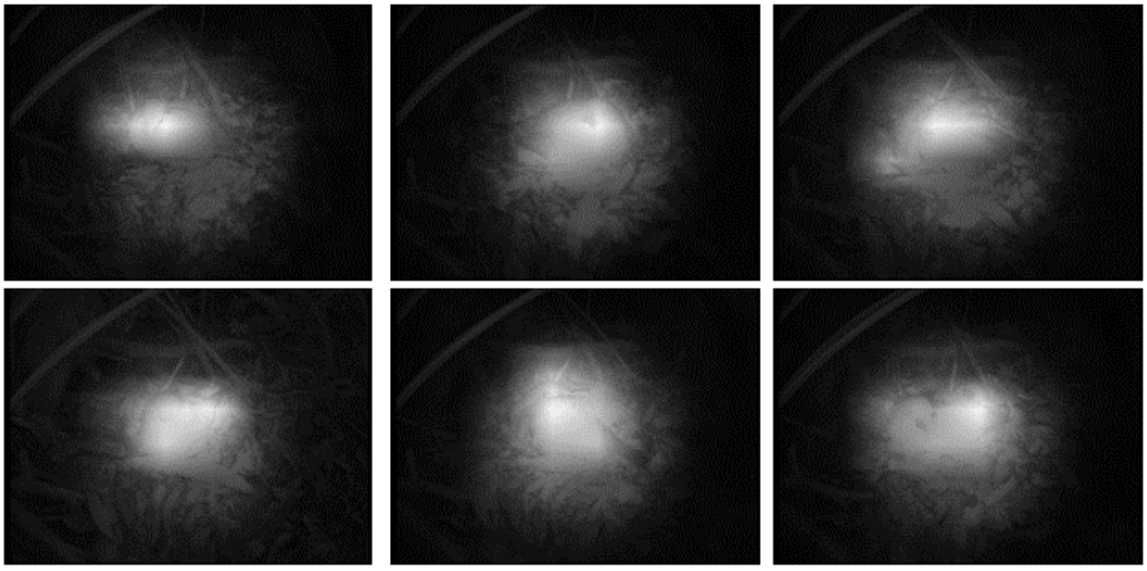
4.4.5 Qualitative Evaluation

Samples of the qualitative attention results are presented in Figures 13 & 14. The attention mask, extracted from the model, has been trained to track the event, in all processed frames, without supervision. Our results show that the events are tracked and localized in various lighting (shadows, day/night) and occlusion conditions. Attention has also learned to indefinitely focus on the bird regardless of its motion state (stationary/Non-stationary), which indicates that the model has acquired a high-level temporal understanding of the events in the scene and learned the underlying structure of the bird in an unsupervised manner. Supplementary results [1] display a timelapse of attention weighted frames during illumination changes and moving shadows. Figure 8 summarizes the visualization of the prediction loss signal, motion weighted loss signal, and attention mask during a walk in and out event.

[1] Available at https://ramyamounir.github.io/projects/EventSegmentation

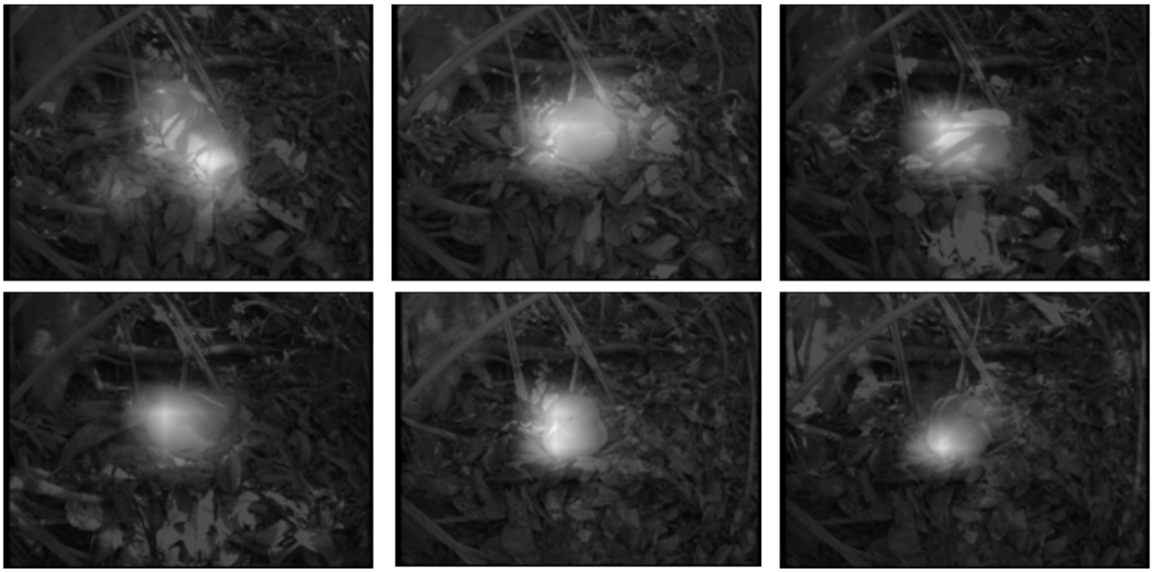


(a) Samples at daytime while the bird is stationary

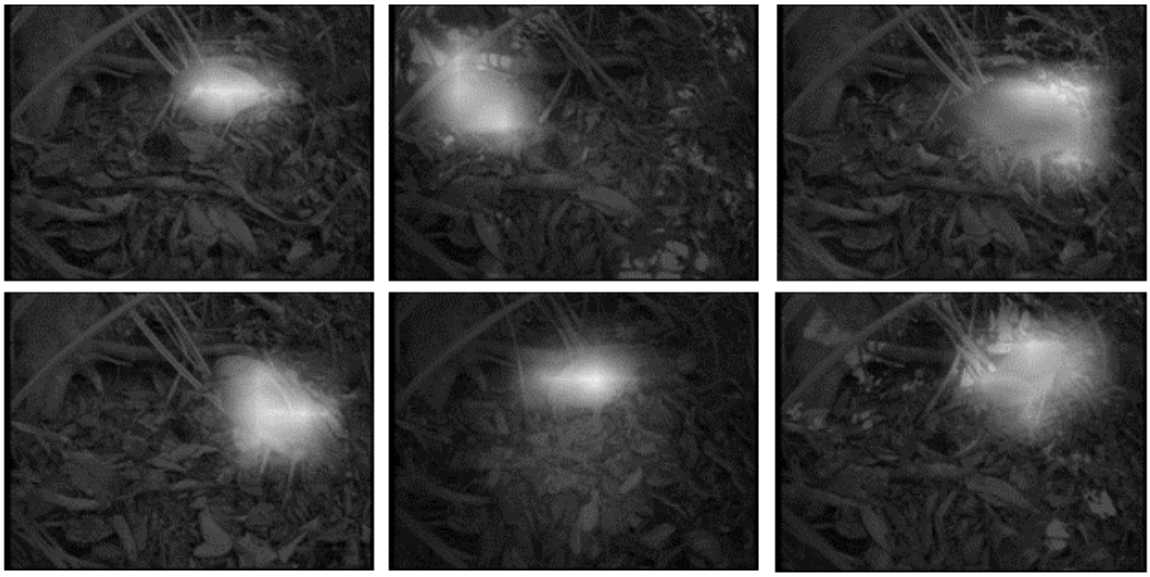


(b) Samples at night while the bird is stationary

Figure 13: Samples of Bahdanau attention weights visualized on input images.



(a) Samples at daytime with moving shadows



(b) Samples at daytime while the bird is moving

Figure 14: Samples of Bahdanau attention weights visualized on input images.

5 Version 3: Spatio-Temporal Localization using Prediction Loss Map

Finally, we show that the framework from the previous section can be further enhanced to localize the event in the images, i.e., mark using bounding boxes where the event is happening. The attention-maps results in Section 4 only generate a heat map over the grid size defined by the encoder's output spatial resolution. In other words, the attention map acts as a pointer as to where the action is happening; however, it does not generate an accurate mask nor a bounding box. To generate a bounding box, we use a region proposal network and prediction loss map to filter these proposals using a spatio-temporal energy minimization function. The following description is from [2].

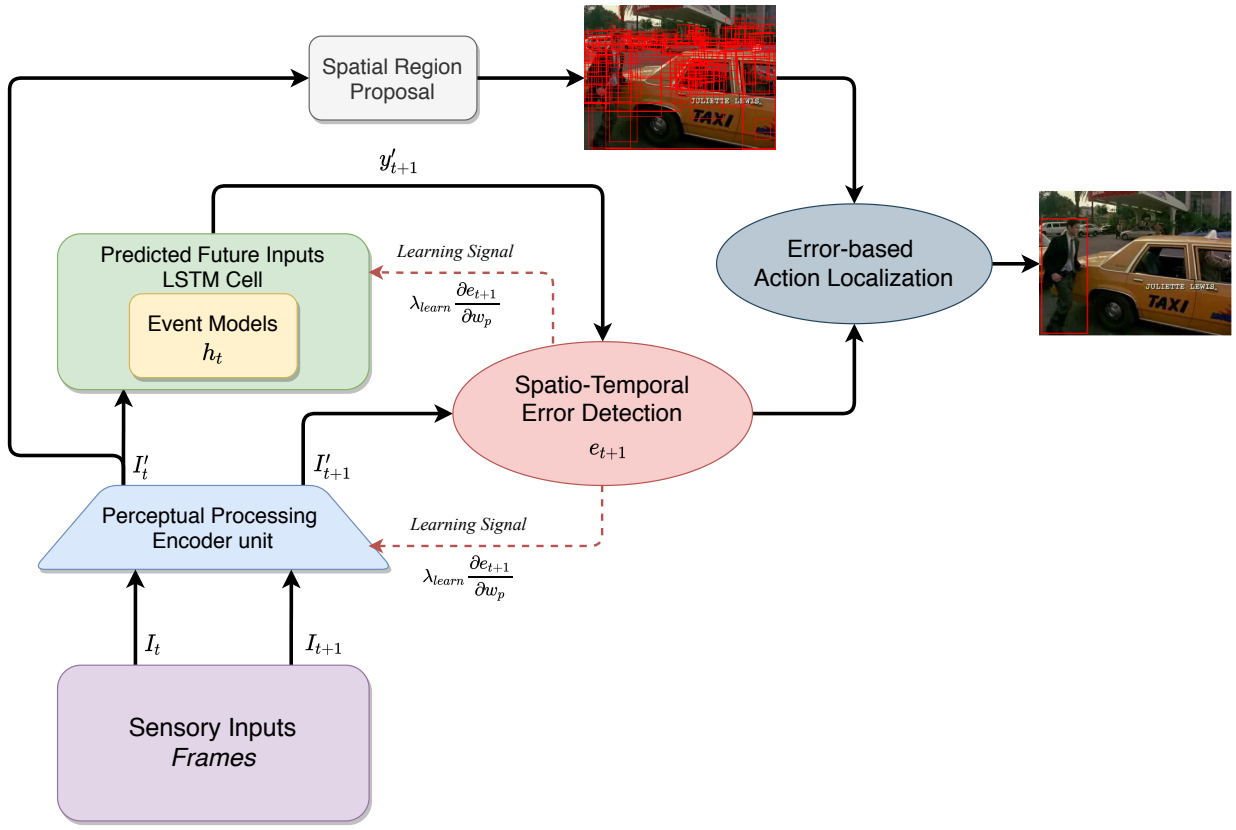


Figure 15: **Version 3: Spatio-Temporal Localization using Prediction Loss Map:** This approach envelopes Version 2 with a component that localizes the dominant action based on prediction error. It has four core components: (i) feature extraction and spatial region proposal, (ii) a future prediction framework, (iii) a spatial-temporal error detection module, and (iv) the error-based action localization process.

We start by extracting relevant features (Section 5.1) from the raw frames in pixel space. The extracted features will be used as an input to the region proposal network and the prediction modules (Section 5.2). An energy function combines the prediction loss and the proposed regions (bounding boxes) to extract action tubes (Section 5.4) that are consistent spatially and temporally. Figure 15 shows the four components of the architecture: (i) feature extraction and spatial region proposal, (ii) a self-supervised future prediction framework, (iii) a spatial-temporal error detection module, and (iv) the error-based action localization process.

5.1 Feature Extraction

Similar to previous approaches, a CNN encoder is used to extract relevant features. However, the extracted features in this approach are used as an input for the spatial region proposal network and the prediction stack. The region proposal is essentially a set of bounding boxes identifying possible areas of action and associated objects for each frame. A pre-trained CNN is used for extracting the features and generating proposals.

We use class-agnostic proposals (i.e., the object category is ignored, and only feature-based localizations are taken into account) for two primary reasons. First, we do not want to make any assumptions on the actor's characteristics, such as label, role, and affordance. Second, despite significant progress in object detection, there can be many missed detections, especially when the object (or actor) performs actions that can transform their physical appearance. It is to be noted that these considerations can result in a large number of region proposals that require careful and robust selection but can yield higher chances of correct localization.

5.2 Hierarchical Prediction Stack

Similar to the approaches mentioned above, this framework learns a predictive function on the high-level features extracted by the encoder. A Long Short-Term Memory (LSTM) network is used to predict the next set of feature vectors, in a temporal sequence, given the encoded input and an internal model of the current event. The internal model effectively captures the spatial-temporal dynamics of the observed event. Similar to the previous attention model, the LSTM processes a set of feature vectors each time step, not a single feature vector. The spatial resolution of the feature maps is determined by the last convolution layer of the encoder.

The choice of an LSTM network is not arbitrary; while other approaches such as convolutional decoders [40] and mixture-of-network models [95] are viable alternatives for future prediction, we propose the use of recurrent networks for the following reasons. First, we want to model the temporal dynamics across *all* frames of the observed action (or event). Second, LSTMs can allow for multiple possible futures and hence will not average the outcomes of these possible futures, as can be the case with other prediction models. Imagine a sequence of frames $I_a = (I_a^1, I_a^2, \ldots I_a^n)$ corresponding to the activity a. Given the complex nature of videos such as those in instructional or sports domains, the next set of frames can be followed by frames of activity b or c with equal probability, given by $I_b = (I_b^1, I_b^2, \ldots I_b^m)$ and $I_c = (I_c^1, I_c^2, \ldots I_c^k)$ respectively. Using a fully connected or convolutional prediction unit is likely to result in the prediction of features that tend to be the average of the two activities a and b, i.e., $I_{avg}^k = \frac{1}{2}(I_b^k + I_c^k)$ for the time k. This is not a desirable outcome because the predicted features can either be an unlikely outcome or, more probably, be outside the plausible manifold of representations. The use of recurrent networks such as RNNs and LSTMs allows for multiple futures that can be possible at time $t+1$, conditioned upon the observation of frames until time t. Third, since we work with error-based localization, using LSTMs can ensure that the learning process aggregates the spatial-temporal error across time and can yield progressively better predictions, especially for actions of longer duration.

In contrast to the previous approaches, the prediction unit here consists of a stack of LSTMs. The output of one LSTM is used as the input to another LSTM. Each LSTM in the stack has its own parameters, defining a different internal model of the event based on its position in the hierarchy. This modification allows for modeling both spatial and temporal dependencies since each higher-level LSTM acts as a progressive decoder framework that captures the temporal dependencies captured by the lower-level LSTMs. The first LSTM captures the spatial dependency

that is propagated up the prediction stack.

The updated hidden state of the first (bottom) LSTM layer (h_t^1) depends on the current observation f_t^S, the previous hidden state (h_{t-1}^1) and memory state (m_{t-1}^1). Each of the higher-level LSTMs at level l takes the output of the bottom LSTM's output h_t^{l-1} and memory state m_t^{l-1} and can be defined as $(h_t^l, m_t^l) = LSTM(h_{t-1}^l, h_t^{l-1}, m_t^{l-1})$. Note this is different from a typical hierarchical LSTM model [78] in that the higher LSTMs are impacted by the output of the lower level LSTMs at *current time step*, as opposed to that from the previous time step. Collectively, the event model W_e is described by the learnable parameters and their respective biases from the hierarchical LSTM stack.

Hence, the top layer of the prediction stack acts as the decoder whose goal is to predict the next feature f_{t+1}^S given all previous predictions $\hat{f}_1^S, \hat{f}_2^S, \ldots \hat{f}_t^S$, an event model W_e and the current observation f_t^S. We model this prediction function as a log-linear model characterized by

$$\log p(\hat{f}_{t+1}^s | h_t^l) = \sum_{n=1}^{t} f(W_e, f_t^S) + log\, Z(h_t) \tag{9}$$

where h_t^l is the hidden state of the l^{th} level LSTM at time t and $Z(h_t)$ is a normalization constant. The LSTM prediction stack acts as a generative process for anticipating future features.

5.3 Prediction Loss

The attention map for this framework is extracted directly from the actual prediction loss (See Figure 7) at the top of the prediction stack. The prediction loss is a factor of the quality of the predictions made and the relative spatial alignment of the prediction errors. The motion weighted loss from Equation 8 is used to compute a weight α_{ij} associated with each spatial location (i, j) in the predicted feature $\hat{f}_{t+1}^S$ as

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{m=1}^{w_k} \sum_{n=1}^{h_k} \exp(e_{mn})} \tag{10}$$

where e_{ij} represents the weighted prediction error at location (i, j) (Equation 8). It can be considered to be a function $a(f_t^S, h_{t-1}^l)$ of the state of the top-most LSTM and the input feature f_t^S at time t. The resulting matrix is an error-based attention map that allows us to localize the prediction error at a specific spatial location. And the average spatial error over time, $E(t)$, is used for temporal localization.

5.4 Action Tubes Extraction

The actions are extracted as tubes using an energy objective function that is minimized to ensure spatial and temporal consistency. The action localization module receives, as an input, a stream of bounding box proposals (several proposals per frame) and a stream of prediction loss maps (one per frame). An energy function is defined to extract coherent action tubes by filtering the proposals using the prediction loss map and returning the collection of proposals with a higher probability of action localization. This is achieved by assigning an energy term to each of the bounding box proposals ($\mathcal{B}_{it}$) at time t and choosing the top k bounding boxes with the least energy as our final proposals. The energy of a bounding box $\mathcal{B}_{it}$ is defined as

$$E(\mathcal{B}_{it}) = w_\alpha \ \phi(\alpha_{ij}, \mathcal{B}_{it}) + w_t \delta(\mathcal{B}_{it}, \{\mathcal{B}_{j,t-1}\}) \tag{11}$$

where $\phi(\cdot)$ is a function that returns a value characteristic of the distance between the bounding box center and location of the maximum error, $\delta(\cdot)$ is a function that returns the minimum spatial distance between the current bounding box and the closest bounding box from the previous time step. The constants w_α and w_t are scaling factors. Figure 18 shows an example of extracted action tubes for a single activity in a stream of frames.

5.5 Results

5.5.1 Data

We use three publicly available datasets to evaluate our approach on the action localization task.

UCF Sports [71] is an action localization dataset consisting of 10 classes of sports actions, such as skating and lifting, collected from sports broadcasts. It is an interesting dataset since it has a high concentration of distinct scenes and motions that make it challenging for localization and recognition. We use the splits (103 training and 47 testing videos) as defined in [47] for evaluation.

JHMDB [38] is composed of 21 action classes and 928 trimmed videos. All videos are annotated with human-joints for every frame. The ground truth bounding box for the action localization task is chosen such that the box encompasses all the joints. This dataset offers several challenges, such as increasing amounts of background clutter, high inter-class similarity, complex motion (including camera motion), and occluded objects of interest. We present all results as the average across all three splits.

THUMOS'13 [41] is a subset of the UCF-101 [82] dataset, consisting of 24 classes and 3,207 videos. Ground truth bounding boxes are provided for each of the classes for the action localization task. It is also known as the **UCF-101-24** dataset. Following prior works [53, 81], we perform the experiments and report results on the first split.

We also analyze the approach's generalization ability on egocentric videos by evaluating it on the *unsupervised gaze prediction task*. There is ample evidence from cognitive psychology for a strong correlation between gaze points and action localization [89]. Hence, the gaze prediction task would be a reasonable measure of the generalization to action localization in egocentric videos. We evaluate the performance on the **GTEA Gaze** [16] dataset, which consists of 17 sequences of tasks performed by 14 subjects, with each sequence lasting about 4 minutes. We use the official splits for the GTEA datasets as defined in prior works [16].

5.5.2 Metrics and Baselines

For the **action localization** task, we follow prior works [53, 81] and report the mean average precision (mAP) at various overlap thresholds, obtained by computing the Intersection Over Union (IoU) of the predicted and ground truth bounding boxes. We also evaluate the quality of bounding box proposals by measuring the average, per-frame IoU, and the bounding box *recall* at varying overlap ratios.

Since ours is an unsupervised approach, we obtain class labels by clustering the learned representations using the *k-means* algorithm. While more complicated clustering may yield better recognition results [81], the k-means approach allows us to evaluate the robustness of learned features. We evaluate our approach in two settings K_{gt} and K_{opt}, where the number of clusters is

Supervision	Approach	Approach
Full	STPD[91]	44.6
	Max Path Search [90]	**54.3**
Weak	Ma et al. [60]	44.6
	GBVS [24]	42.1
	Soomro et al. [81]	**47.7**
None	IME Tublets [37]	51.5
	APT [93]	**63.7**
	Predictive Approach	55.7

Table 5: Comparison with fully supervised and weakly supervised baselines on class-agnostic action localization on UCF Sports dataset. We report the average localization accuracy of each approach i.e. average IoU.

set to the number of ground truth action classes, and an optimal number obtained through the elbow method [43], respectively. From our experiments, we observe that K_{opt} is three times the number of ground truth classes, which is not unreasonable and has been a working assumption in other deep learning-based clustering approaches [31]. Clusters are mapped to the ground truth clusters for evaluation using the Hungarian method, as done in prior unsupervised approaches [39, 100]. We also compare against other LSTM and attention-based approaches (Section 5.5.3.3) to the action localization problem for evaluating the effectiveness of our training protocol.

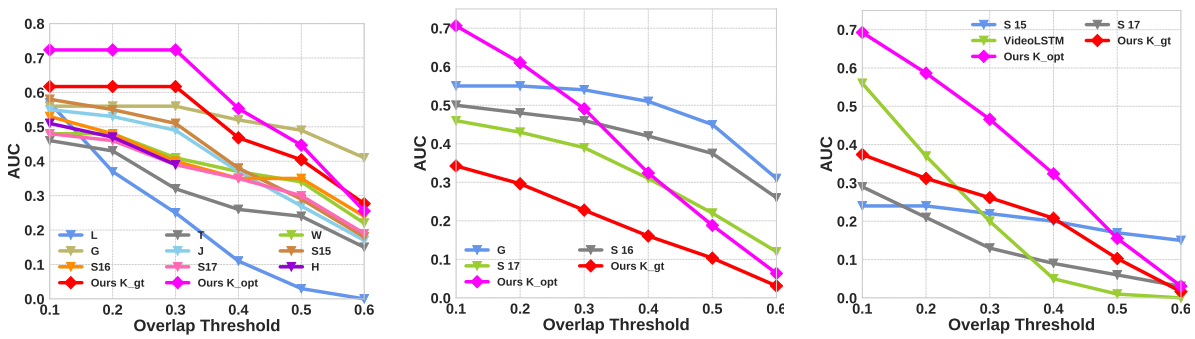
For the **gaze prediction** task, we evaluate the approaches using **Area Under the Curve** (AUC), which measures the area under the curve on saliency maps for true positive versus false-positive rates under various threshold values. We also report the **Average Angular Error** (AAE), which measures the angular distance between the predicted and ground truth gaze positions. Since our model's output is a saliency map, AUC is a more appropriate metric than the average angular error (AAE), which requires specific locations.

5.5.3 Quantitative Evaluation

In this section, we present the quantitative evaluation of our approach on two different tasks, namely action localization and egocentric gaze prediction. For the action localization task, we evaluate our approach on two aspects - the quality of proposals and spatial-temporal localization.

5.5.3.1 Quality of Localization Proposals We first evaluate the quality of our localization proposals by assuming perfect class prediction. This allows us to independently assess the quality of localization performed in a self-supervised manner. We present the evaluation results in Table 5 and compare them against fully supervised, weakly supervised, and unsupervised baselines. As can be seen, the predictive approach outperforms many supervised and weakly supervised baselines. APT [93] achieves a higher localization score. However, it produces, on average, $1,500$ proposals per video, whereas our approach returns approximately 10 proposals. A large number of localization proposals per video can lead to higher recall and IoU but makes the localization task more challenging, i.e., action labeling per video harder, and can affect the ability to generalize across domains.

Also, it should be noted that our approach produces proposals in *streaming* fashion, as opposed to many of the other approaches, which produce action tubes based on motion computed across the entire video. This can make real-time action localization in streaming videos harder.



(a) (b) (c)

Figure 16: AUC for the action localization tasks are shown for (a) UCF Sports, (b) JHMDB and (c) THUMOS13 datasets. We compare against baselines with varying levels of supervision such as Lan *et al.* [47], Tian *et al.* [88], Wang *et al.* [99], Gkioxari and Malik [22], Jain *et al.* [37], Soomro *et al.* [79, 80, 81], Hou *et al.* [34], and VideoLSTM [53].

5.5.3.2 Spatial-temporal Action Localization We also evaluate our approach on the spatial-temporal localization task. This evaluation allows us to analyze the robustness of the self-supervised features learned through prediction. We generate video-level class labels through clustering and use the standard evaluation metrics (Section 5.5.2) to quantify the performance. The AUC curves with respect to varying overlap thresholds are presented in Figure 16. We compare against a mix of supervised, weakly-supervised, and unsupervised baselines on all three datasets.

On the **UCF Sports** dataset (Figure 16(a)), we outperform all baselines including several supervised baselines, except for Gkioxari and Malik [22] at higher overlap thresholds ($\sigma > 0.4$) when we set number of clusters k to the number of ground truth classes. When we allow for some over-segmentation and use the *optimal* number of clusters, we outperform all baselines till $\sigma > 0.5$.

On the **JHMDB** dataset (Figure 16(b)), we find that our approach, while having high recall even at more stringent thresholds (77.8% @ $\sigma = 0.5$), the large camera motion and intra-class variations have a significant impact on the classification accuracy. Hence, the mAP suffers when we set k to be the number of ground truth classes. When we set the number of clusters to the optimal number of clusters, we outperform other baselines at lower thresholds (mAP @ σ<0.5). It should be noted that the other unsupervised baseline (Soomro *et al.* [81]) uses object detection proposals from a Faster R-CNN backbone to score the *"humanness"* of a proposal. This assumption tends to make the approach biased towards human-centered action localization and affects its ability to generalize towards actions with non-human actors. On the other hand, we do not make any assumptions on the characteristics of the actor, scene, or motion dynamics.

On the **THUMOS'13** dataset (Figure 16(c)), we achieve consistent improvements over unsupervised and weakly supervised baselines, at $k = k_{gt}$ and achieve state-of-the-art mAP scores when $k = k_{opt}$. It is interesting to note that we perform competitively (when $k = k_{gt}$) with the weakly-supervised attention-based VideoLSTM [53], which uses a convLSTM for temporal modeling along with a CNN-based spatial attention mechanism. It should be noted that we have a higher recall rate (0.47 @ $\sigma = 0.4$ and 0.33 @ $\sigma = 0.5$) at higher thresholds than other state-of-the-art approaches on THUMOS'13 and shows the robustness of the error-based localization approach to intra-class variation and occlusion.

Clustering quality. Since there is a significant difference in the mAP score when we set a different number of clusters in k-means, we measured the homogeneity (or purity) of the clustering. The homogeneity score measures the "quality" of the cluster by measuring how well a cluster models

Approach	Annotations		# Proposals	Average Recall					mAP
	Labels	Boxes		0.1	0.2	0.3	0.4	0.5	@0.2
ALSTM [75]	✓	✗	1	0.46	0.28	0.05	0.02	-	0.06
VideoLSTM [53]	✓	✗	1	0.71	0.52	0.32	0.11	-	0.37
Actor Supervision [15]	✓	✗	~ 1000	**0.89**	-	-	-	**0.44**	0.46
Our Approach	✗	✗	~ 10	0.84	**0.72**	**0.58**	**0.47**	0.33	**0.59**

Table 6: Comparison with other LSTM-based and attention-based approaches on the THUMOS'13 dataset. We report average recall at various overlap thresholds, mAP at 0.2 overlap threshold and the average number of proposals per frame.

a given ground-truth class. Since we allow the over-segmentation of clusters when we set k to the optimal number of clusters, this is an essential measure of feature robustness. Higher homogeneity indicates that intra-class variations are captured since all data points in a given cluster belong to the same ground truth class. We observe an average homogeneity score of 74.56% when k is set to the number of ground truth classes and 78.97% when we use the optimal number of clusters. As can be seen, although we over-segment, each of the clusters typically models a single action class to a high degree of integrity.

5.5.3.3 Comparison with other LSTM-based approaches We also compare our approach with other LSTM-based and attention-based models to highlight the importance of the self-supervised learning paradigm. Since LSTM-based frameworks can have highly similar architectures, we consider different requirements and characteristics, such as the level of annotation required for training and the number of localization proposals returned per video. We compare with three approaches similar in spirit to our approach - ALSTM [75], VideoLSTM [53] and Actor Supervision [15] and summarize the results in Table 6. It can be seen that we significantly outperform VideoLSTM and ALSTM on the THUMOS'13 dataset in both recall and $mAP\ @\ \sigma = 0.2$.

Actor Supervision [15] outperforms our approach on recall, but it is to be noted that the region proposals are dependent on two factors - (i) object detection-based actor proposals and (ii) a filtering mechanism that limits proposals based on ground truth action classes, which can increase the training requirements and limit generalizability. Also, note that returning a higher number of localization proposals can increase recall at the cost of generalization.

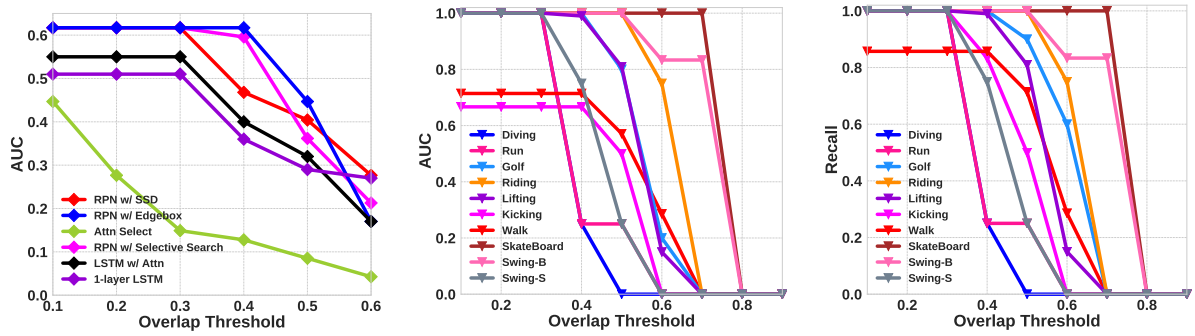
5.5.3.4 Ablative Studies This predictive framework has three major components that affect its performance the most - (i) the region proposal module, (ii) the future prediction module, and (iii) the error-based action localization module. We consider and evaluate several alternatives to all three modules. We choose selective search [92] and EdgeBox [110] as alternative region proposal methods to SSD.

We use an attention-based localization method for action localization as an approximation of the ALSTM [75] to evaluate the effectiveness of using the error-based localization. We also evaluate a 1-layer LSTM predictor with a fully connected decoder network to approximate [1] on the localization task. We evaluate the effect of attention-based prediction by introducing a Bahdanau [7] attention layer before prediction as an alternative to the error-based action localization module.

These ablative studies are conducted on the UCF Sports dataset. The results are plotted in Figure 17(a). It can be seen that the use of the prediction error-based localization has a significant improvement over a trained attention-based localization approach. We can also see that the

	Itti *et al.* [36]	GBVS [27]	AWS-D [49]	Center Bias	OBDL [33]	Ours
AUC	0.747	0.769	0.770	0.789	0.801	**0.861**
AAE	18.4	15.3	18.2	**10.2**	15.6	13.6

Table 7: Comparison with state-of-the-art on the unsupervised egocentric gaze prediction task on the GTEA dataset.



(a) (b) (c)

Figure 17: Qualitative analysis of our approach on UCF Sports dataset (a) ablative variations on AUC. (a) class-wise AUC, and (c) class-wise bounding box recall at different overlap thresholds.

choice of region proposal methods does have some effect on the performance of the approach, with selective search and EdgeBox proposals doing slightly better at higher thresholds ($\sigma \in (0.4, 0.5)$) at the cost of inference time and additional bounding box proposals (50 compared to the 10 from SSD-based region proposal). Using SSD for generating proposals allows us to share weights across the frame encoder and region proposal tasks and reduce the memory and computational footprint of the approach. We also find that using attention as part of the prediction module significantly impacts the architecture's performance. It could, arguably, be attributed to the objective function, which aims to minimize the prediction error. Note that we use the *error-based attention* to localize events in this approach as opposed to a learned attention vector from Section 4. We find that using Bahdanau attention to encode the features could impact the prediction function for this approach.

5.5.3.5 Unsupervised Egocentric Gaze Prediction Finally, we evaluate the ability to generalize to egocentric videos by quantifying the model's performance on the unsupervised gaze prediction task. Given that we do not need any annotations or other auxiliary data, we employ the same architecture and training strategy for this task. We evaluate on the GTEA gaze dataset and compare it with other unsupervised models in Table 7. As can be seen, we obtain competitive results on the gaze prediction task, outperforming all baselines on both the AUC and AAE scores. It is to be noted that we outperform the center bias method on the AUC metric. Center bias exploits the spatial bias in egocentric images and always predicts the center of the video frame as the predicted gaze position. The AUC metric's significant improvement indicates that our approach predicts gaze fixations that are more closely aligned with the ground truth than the center bias approach. Given that the model was not designed explicitly for this task, it is a remarkable performance, especially given the performance of fully supervised baselines such as DFG [108], which achieves 10.6 and 88.3 for AUC and AAE.

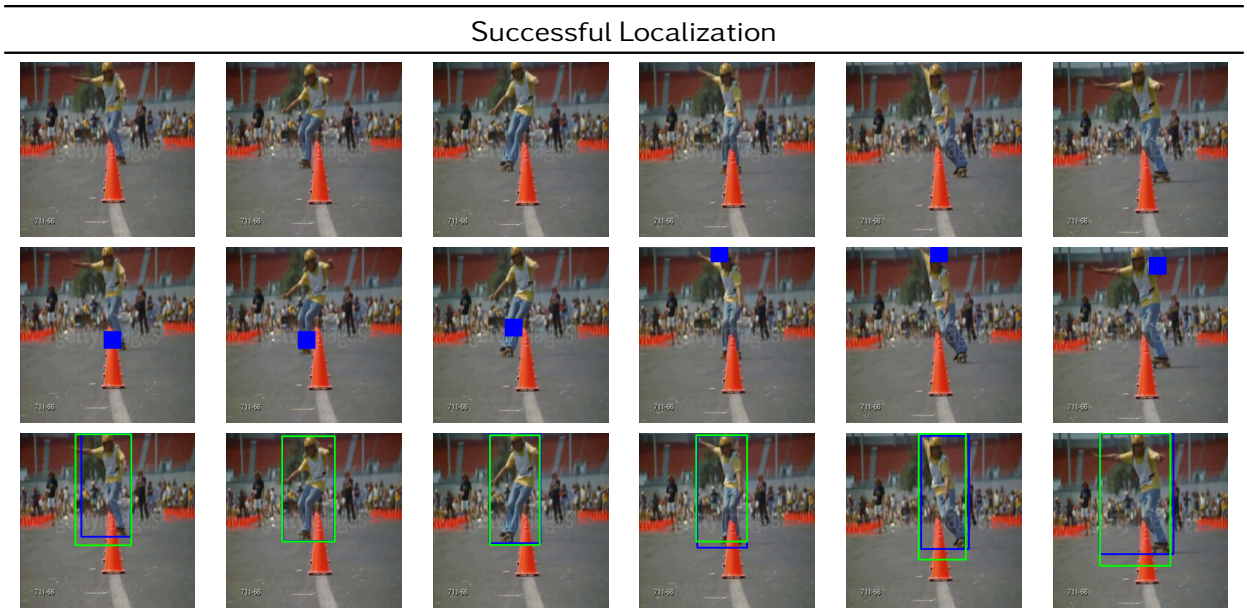


Figure 18: **Qualitative Examples**: Error-based attention location and the final prediction. Green BB: Prediction, Blue BB: Ground truth

5.5.4 Qualitative Evaluation

We find that our approach has a consistently high recall for the localization task across datasets and domains. We consider that an action is correctly localized if the average IoU across all frames is higher than 0.5, which indicates that most, if not all, frames in a video are correctly localized. We illustrate the recall scores and subsequent AUC scores for each class in the UCF sports dataset in Figures 17(b) and (c). For many classes ($7/10$ to be specific), we have more than 80% recall at an overlap threshold of 0.5. Through visual inspection, we find that the spatial-temporal error is often correlated with the actor but is usually not at the center of the region of interest and thus reduces the quality of the chosen proposals. We illustrate this effect in Figure 18. The first row shows the input frame, the second shows the error-based attention, and the last row shows the final localization proposals. If more proposals are returned (as is the case with selective search and EdgeBox), we can obtain a higher recall (Figure 17(b)) and higher mAP.

6 Other Event Segmentation Approaches in Computer Vision

Typically, there are three different classes of approaches for temporal event segmentation: fully supervised, weakly supervised, and fully unsupervised. While fully supervised approaches have more precise localization and better labeling accuracy, the required number of annotations is rather large. The labeled training data demands do not scale well with an increase in label classes. While not requiring frame-level annotations, weakly supervised approaches have the underlying assumption that there exists a large, annotated training set that allows for effective detection of all possible actors (both human and non-human) in the set of action classes. Unsupervised approaches, such as ours, do not make any such assumptions but can result in poorer localization performance. We alleviate this to an extent by leveraging advances in region proposal mechanisms and robust self-learning representations. One particular class of unsupervised approach, called self-supervision, uses the data itself, without annotations, for supervision.

6.1 Supervised Approaches

Fully supervised methods have been the dominant approach to temporal event segmentation. They consider the problem to be fully supervised and use the groundtruth annotations to take a *segment by classification* approach i.e., they assign the labels to semantically coherent *"chunks"*, with contiguous frames sharing the same label, to segment the video into its constituent segments. The common pipeline in these approaches has been to extract features (either hand-crafted or automated using deep learning) to perform frame-based labeling support vector machines [44] or model temporal dynamics using Hidden Markov Models [44], temporal convolutional neural networks (TCN) [67], spatiotemporal convolutional neural networks (CNN) [48] and recurrent networks [69] to name a few. While fully supervised approaches are appealing due to their strong quantitative performance, obtaining large-scale annotated datasets, especially with frame-level annotations, can become quite expensive and may not always be available. This problem becomes more pronounced as the granularity of the events becomes finer.

Similarly, some approaches tackle action localization through the simultaneous generation of bounding box proposals and labeling each bounding box with the predicted action class. Both bounding box generation and labeling are fully supervised, i.e., they require ground truth annotations of both bounding boxes and labels. Typical approaches leverage advances in object detection to include temporal information [22, 34, 37, 79, 80, 88, 90, 99] for proposal generation. The final step typically involves the use of the Viterbi algorithm [22] to link the generated bounding boxes across time.

6.2 Weakly-supervised Approaches

The underlying concept behind weak supervision is to alleviate the need for direct labeling by leveraging accompanying text scripts or instructions as indirect supervision for learning highly discriminant features. There have been two common approaches to weakly supervised learning for temporal segmentation of videos - (1) using script or instructions for weak annotation [9, 13, 6, 62], and (2) following an incomplete temporal localization of actions for learning and inference [35, 69]. While such approaches model the temporal transitions using RNNs, they still rely on enforcing semantics for segmenting actions and hence require some supervision for learning and inference.

These approaches have been explored for action localization to reduce the need for extensive annotations [15, 47, 53, 75]. They typically only require video-level labels and rely on object detection-based approaches to generate bounding box proposals. It is to be noted that weakly supervised approaches also use object-level labels and characteristics to guide the bounding box selection process. Some approaches [15] use a similarity-based tracker to connect bounding boxes across time to incorporate temporal consistency.

6.3 Unsupervised Approaches

Unsupervised approaches do not need external supervision in terms of annotated training data that can be used to determine when the output is right and wrong. These approaches have not been explored to the same extent as supervised and weakly-supervised approaches. The primary approach is to use clustering as the unsupervised approach using discriminant features[8, 74]. The models incorporate a temporal consistency into the segmentation approach by using either LSTMs [8] or generalized mallows model [74]. Garcia *et al.* [20] explore the use of a generative LSTM network to segment sequences like we do. However, they handle only coarse temporal resolution in life-log images sampled as far apart as 30 seconds. Consecutive images when events

change have more variability making for easier discrimination. Besides, they require an iterative training process, which we do not.

Some approaches do not require any supervision for either labels or bounding boxes. The two more common approaches are to generate action proposals using (i) super-voxels [37, 81] and (ii) clustering motion trajectories [93]. It should be noted that [81] also uses object characteristics to evaluate the "humanness" of each super-voxel to select bounding box proposals.

Our approach falls into the class of unsupervised action localization approaches. The most closely related approaches (with respect to architecture and theme) to ours are VideoLSTM [53] and Actor Supervision [15], which use attention in the selection process for generating bounding box proposals, but require video-level labels. We, on the other hand, do not require any labels or bounding box annotations for training.

6.4 Self-Supervised Approaches

One class of unsupervised approach that has gained attention is self-supervised approaches that use training data, but unannotated. There are two main types of self-supervised approaches: (1) hiding a subset of the data and trying to predict it, i.e., predict the occluded from the visible or predict color from gray-level [109, 96]; (2) altering the input and predicting the function (parameters) used for altering it [21, 14, 64, 17]. The prediction of part of the input, or the unaltered version of the input, forces the network to learn good semantic features from the dataset. Contrastive learning approaches [11, 23, 10] fall under the second category where the input is altered and the network is forced to learn a representation that maximizes the similarity between the original and the altered input. These approaches have been found to learn fairly robust representations that can then be coupled with different tasks, i.e., labeling, motion estimation, and so on.

In the context of videos, self-supervision has been mostly prediction of the next couple of image frames [84, 63, 65, 56, 18]. Feature-level predictions have been proposed to discover co-occurring visual feature spaces [98] and learn representations for better recognition [55, 52]. Such approaches propose the discovery of context through co-occurrences of features across temporal sequences to learn either multi-feature correlation such as motion and appearance modeling for future frame generation [52] or for unsupervised learning of co-occurring concepts like the presence of noses and eyes in faces [98].

There is also an increasing body of work on the prediction of future activities [86, 57, 59, 54, 42, 19, 97, 5, 73, 25]. However, these approaches all adopt the standard annotated training-based deep-learning framework.

Our approach is similar in spirit to the predictive learning framework from videos espoused by Vondrick et al. [95], but for higher-level concepts, instead of just action labels. There are strong indications from brain and cognitive studies, apart from Zack's research group, that predictions play a significant role in how the brain circuits learn new concepts. Based on studies of many neuroscience experiments, Hawkins *et al.* [28] have also suggested a repeated architecture of layers of feed-forward, predictive memory, and temporal pooling, which they demonstrated for anomaly detection in 1D signals. Heeger [30] has also has shown the effectiveness of a layered architecture with bottom-up and top-down connections, but driven by prior predictions for temporal signal processing. Lotter *et al.* [56] have implemented a predictive coding stack, but for video frame-level prediction.

7 Conclusions

This chapter drew from cognitive science research to define the problem of event segmentation and to design highly effective computer vision algorithms for spatio-temporal segmentation of events in videos. These approaches do not require any annotated data, nor do they require multiple passes through the data. They can process *streaming* video data while learning robust representations for segmenting events.

The main idea is to use predictive learning, as posited in the Event Segmentation Theory in cognitive science, to detect event boundaries. As in the EST, a perceptual processing unit (a CNN stack) sends the extracted features from the current frame (conditioned on the working event model) to a prediction unit (LSTM, a stack of LSTMs, bounding box prediction head), which predicts future perceptual features. A mismatch in the predicted features and the actual features computed for the next time instance generates a prediction error signal. A high error signals an event boundary. A new event model is now needed to make future predictions. The prediction error triggers a gating mechanism to update the working event model (hidden states in the LSTM) with a new model.

Extensive experiments on a variety of domains demonstrate that the predictive learning approach can learn robust event representations from unlabeled *streaming* video sequences, *with only one epoch of training (single pass through video)*. The event representations were able to obtain state-of-the-art results in unsupervised temporal segmentation (Section 3) and spatial-temporal action localization (Section 5) while offering competitive performance with fully supervised baselines that require extensive amounts of annotated training data. Additionally, we show that the predictive learning framework can process and segment streaming video data of extremely long duration (Section 4), at close to real-time processing. Predictive learning can help break the increasing dependency on training data and move towards open-world visual understanding. We hope that these results encourage research in this promising direction and unlatch computer vision research from the increasing dependence on more and more annotated data.

8 Acknowledgement

This research was supported in part by the US National Science Foundation grants CNS 1513126, IIS 1956050, and IIS 1955230.

References

[1] Sathyanarayanan N. Aakur and Sudeep Sarkar. "A Perceptual Prediction Framework for Self Supervised Event Segmentation". In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2019.

[2] Sathyanarayanan N Aakur and Sudeep Sarkar. "Action Localization through Continual Predictive Learning". In: *arXiv preprint arXiv:2003.12185* (2020).

[3] Sathyanarayanan Aakur, Fillipe DM de Souza, and Sudeep Sarkar. "Going Deeper with Semantics: Exploiting Semantic Contextualization for Interpretation of Human Activity in Videos". In: *IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE. 2019.

[4] *ActEV: Activities in Extended Video*. URL: https://actev.nist.gov/.

[5] Alexandre Alahi, Kratarth Goel, Vignesh Ramanathan, Alexandre Robicquet, Li Fei-Fei, and Silvio Savarese. "Social lstm: Human trajectory prediction in crowded spaces". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 961–971.

[6] Jean-Baptiste Alayrac, Piotr Bojanowski, Nishant Agrawal, Josef Sivic, Ivan Laptev, and Simon Lacoste-Julien. "Unsupervised learning from narrated instruction videos". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 4575–4583.

[7] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. "Neural machine translation by jointly learning to align and translate". In: *arXiv preprint arXiv:1409.0473* (2014).

[8] Bharat Lal Bhatnagar, Suriya Singh, Chetan Arora, CV Jawahar, and KCIS CVIT. "Unsupervised learning of deep feature representation for clustering egocentric actions". In: *International Joint Conference on Artificial Intelligence (IJCAI)*. AAAI Press. 2017, pp. 1447–1453.

[9] Piotr Bojanowski, Rémi Lajugie, Francis Bach, Ivan Laptev, Jean Ponce, Cordelia Schmid, and Josef Sivic. "Weakly supervised action labeling in videos under ordering constraints". In: *European Conference on Computer Vision (ECCV)*. Springer. 2014, pp. 628–643.

[10] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. "Unsupervised learning of visual features by contrasting cluster assignments". In: *Advances in Neural Information Processing Systems* 33 (2020).

[11] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. "A simple framework for contrastive learning of visual representations". In: *arXiv preprint arXiv:2002.05709* (2020).

[12] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. "Bert: Pre-training of deep bidirectional transformers for language understanding". In: *arXiv preprint arXiv:1810.04805* (2018).

[13] Li Ding and Chenliang Xu. "Weakly-Supervised Action Segmentation with Iterative Soft Boundary Assignment". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018.

[14] Carl Doersch, Abhinav Gupta, and Alexei A Efros. "Unsupervised visual representation learning by context prediction". In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 1422–1430.

[15] Victor Escorcia, Cuong D Dao, Mihir Jain, Bernard Ghanem, and Cees Snoek. "Guess where? actor-supervision for spatiotemporal action localization". In: *Computer Vision and Image Understanding* 192 (2020), p. 102886.

[16] Alireza Fathi, Yin Li, and James M Rehg. "Learning to recognize daily actions using gaze". In: *European Conference on Computer Vision*. Springer. 2012, pp. 314–327.

[17] Basura Fernando, Hakan Bilen, Efstratios Gavves, and Stephen Gould. "Self-supervised video representation learning with odd-one-out networks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 3636–3645.

[18] Chelsea Finn, Ian J. Goodfellow, and Sergey Levine. "Unsupervised Learning for Physical Interaction through Video Prediction". In: *CoRR* abs/1605.07157 (2016). arXiv: 1605.07157. URL: http://arxiv.org/abs/1605.07157.

[19] Katerina Fragkiadaki, Sergey Levine, Panna Felsen, and Jitendra Malik. "Recurrent network models for human dynamics". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2015, pp. 4346–4354.

[20] Ana Garcia del Molino, Joo-Hwee Lim, and Ah-Hwee Tan. "Predicting Visual Context for Unsupervised Event Segmentation in Continuous Photo-streams". In: *ACM Conference on Multimedia (ACM MM)*. ACM. 2018, pp. 10–17.

[21] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. *Unsupervised Representation Learning by Predicting Image Rotations*. 2018. arXiv: 1803.07728 [cs.CV].
[22] Georgia Gkioxari and Jitendra Malik. "Finding action tubes". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2015, pp. 759–768.
[23] Jean-Bastien Grill et al. "Bootstrap your own latent-a new approach to self-supervised learning". In: *Advances in Neural Information Processing Systems* 33 (2020).
[24] Matthias Grundmann, Vivek Kwatra, Mei Han, and Irfan Essa. "Efficient hierarchical graph-based video segmentation". In: *2010 ieee computer society conference on computer vision and pattern recognition*. IEEE. 2010, pp. 2141–2148.
[25] William L Hamilton, Rex Ying, and Jure Leskovec. "Representation learning on graphs: Methods and applications". In: *arXiv preprint arXiv:1709.05584* (2017).
[26] Bridgette M Hard, Barbara Tversky, and David S Lang. "Making sense of abstract events: Building event schemas". In: *Memory & cognition* 34.6 (2006), pp. 1221–1235.
[27] Jonathan Harel, Christof Koch, and Pietro Perona. "Graph-based visual saliency". In: *Advances in Neural Information Processing Systems*. 2007, pp. 545–552.
[28] Jeff Hawkins and Subutai Ahmad. "Why neurons have thousands of synapses, a theory of sequence memory in neocortex". In: *Frontiers in neural circuits* 10 (2016), p. 23.
[29] Jeff Hawkins and Sandra Blakeslee. *On intelligence*. Macmillan, 2004.
[30] David J Heeger. "Theory of cortical function". In: *Proceedings of the National Academy of Sciences* 114.8 (2017), pp. 1773–1782.
[31] John R Hershey, Zhuo Chen, Jonathan Le Roux, and Shinji Watanabe. "Deep clustering: Discriminative embeddings for segmentation and separation". In: *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2016, pp. 31–35.
[32] Sepp Hochreiter and Jürgen Schmidhuber. "Long short-term memory". In: *Neural computation* 9.8 (1997), pp. 1735–1780.
[33] Sayed Hossein Khatoonabadi, Nuno Vasconcelos, Ivan V Bajic, and Yufeng Shan. "How many bits does it take for a stimulus to be salient?" In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2015, pp. 5501–5510.
[34] Rui Hou, Chen Chen, and Mubarak Shah. "Tube convolutional neural network (T-CNN) for action detection in videos". In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. 2017, pp. 5822–5831.
[35] De-An Huang, Li Fei-Fei, and Juan Carlos Niebles. "Connectionist temporal modeling for weakly supervised action labeling". In: *European Conference on Computer Vision (ECCV)*. Springer. 2016, pp. 137–153.
[36] Laurent Itti and Christof Koch. "A saliency-based search mechanism for overt and covert shifts of visual attention". In: *Vision Research* 40.10-12 (2000), pp. 1489–1506.
[37] Mihir Jain, Jan Van Gemert, Hervé Jégou, Patrick Bouthemy, and Cees GM Snoek. "Action localization with tubelets from motion". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2014, pp. 740–747.
[38] Hueihan Jhuang, Juergen Gall, Silvia Zuffi, Cordelia Schmid, and Michael J Black. "Towards understanding action recognition". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2013, pp. 3192–3199.
[39] Xu Ji, João F Henriques, and Andrea Vedaldi. "Invariant information clustering for unsupervised image classification and segmentation". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2019, pp. 9865–9874.
[40] Xu Jia, Bert De Brabandere, Tinne Tuytelaars, and Luc V Gool. "Dynamic filter networks". In: *Neural Information Processing Systems*. 2016, pp. 667–675.

[41] Yu-Gang Jiang, Jingen Liu, A Roshan Zamir, George Toderici, Ivan Laptev, Mubarak Shah, and Rahul Sukthankar. *THUMOS challenge: Action recognition with a large number of classes*. 2014.

[42] Kris M Kitani, Brian D Ziebart, James Andrew Bagnell, and Martial Hebert. "Activity forecasting". In: *European Conference on Computer Vision*. Springer. 2012, pp. 201–214.

[43] Trupti M Kodinariya and Prashant R Makwana. "Review on determining number of Cluster in K-Means Clustering". In: *International Journal* 1.6 (2013), pp. 90–95.

[44] Hilde Kuehne, Ali Arslan, and Thomas Serre. "The language of actions: Recovering the syntax and semantics of goal-directed human activities". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2014, pp. 780–787.

[45] Hilde Kuehne, Juergen Gall, and Thomas Serre. "An end-to-end generative framework for video segmentation and recognition". In: *IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE. 2016, pp. 1–8.

[46] Christopher A Kurby and Jeffrey M Zacks. "Segmentation in the perception and memory of events". In: *Trends in cognitive sciences* 12.2 (2008), pp. 72–79.

[47] Tian Lan, Yang Wang, and Greg Mori. "Discriminative figure-centric models for joint action localization and recognition". In: *2011International Conference on Computer Vision*. IEEE. 2011, pp. 2003–2010.

[48] Colin Lea, Austin Reiter, René Vidal, and Gregory D Hager. "Segmental spatiotemporal cnns for fine-grained action segmentation". In: *European Conference on Computer Vision (ECCV)*. Springer. 2016, pp. 36–52.

[49] Victor Leboran, Anton Garcia-Diaz, Xose R Fdez-Vidal, and Xose M Pardo. "Dynamic whitening saliency". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39.5 (2016), pp. 893–907.

[50] Yann LeCun, Yoshua Bengio, et al. "Convolutional networks for images, speech, and time series". In: ().

[51] Peng Lei and Sinisa Todorovic. "Temporal Deformable Residual Networks for Action Segmentation in Videos". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018, pp. 6742–6751.

[52] Ruiyu Li, Makarand Tapaswi, Renjie Liao, Jiaya Jia, Raquel Urtasun, and Sanja Fidler. "Situation recognition with graph neural networks". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2017, pp. 4173–4182.

[53] Zhenyang Li, Kirill Gavrilyuk, Efstratios Gavves, Mihir Jain, and Cees GM Snoek. "Videolstm convolves, attends and flows for action recognition". In: *Computer Vision and Image Understanding* 166 (2018), pp. 41–50.

[54] Junwei Liang, Lu Jiang, Juan Carlos Niebles, Alexander G Hauptmann, and Li Fei-Fei. "Peeking into the future: Predicting future person activities and locations in videos". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 5725–5734.

[55] Wenqian Liu, Abhishek Sharma, Octavia Camps, and Mario Sznaier. "DYAN: A Dynamical Atoms-Based Network For Video Prediction". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 170–185.

[56] William Lotter, Gabriel Kreiman, and David Cox. "Deep predictive coding networks for video prediction and unsupervised learning". In: *arXiv preprint arXiv:1605.08104* (2016).

[57] Pauline Luc, Natalia Neverova, Camille Couprie, Jakob Verbeek, and Yann LeCun. "Predicting deeper into the future of semantic segmentation". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2017, pp. 648–657.

[58] Minh-Thang Luong, Hieu Pham, and Christopher D Manning. "Effective approaches to attention-based neural machine translation". In: *arXiv preprint arXiv:1508.04025* (2015).
[59] Shugao Ma, Leonid Sigal, and Stan Sclaroff. "Learning activity progression in lstms for activity detection and early detection". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 1942–1950.
[60] Shugao Ma, Jianming Zhang, Nazli Ikizler-Cinbis, and Stan Sclaroff. "Action recognition and localization by hierarchical space-time segments". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2013, pp. 2744–2751.
[61] Laurens van der Maaten and Geoffrey Hinton. "Visualizing data using t-SNE". In: *Journal of Machine Learning Research* 9.Nov (2008), pp. 2579–2605.
[62] Jonathan Malmaud, Jonathan Huang, Vivek Rathod, Nick Johnston, Andrew Rabinovich, and Kevin Murphy. "What's cookin'? interpreting cooking videos using text, speech and vision". In: *arXiv preprint arXiv:1503.01558* (2015).
[63] Michaël Mathieu, Camille Couprie, and Yann LeCun. "Deep multi-scale video prediction beyond mean square error". In: *CoRR* abs/1511.05440 (2015). arXiv: 1511.05440. URL: http://arxiv.org/abs/1511.05440.
[64] Ishan Misra, C Lawrence Zitnick, and Martial Hebert. "Shuffle and learn: unsupervised learning using temporal order verification". In: *European Conference on Computer Vision*. Springer. 2016, pp. 527–544.
[65] Natalia Neverova, Pauline Luc, Camille Couprie, Jakob J. Verbeek, and Yann LeCun. "Predicting Deeper into the Future of Semantic Segmentation". In: *CoRR* abs/1703.07684 (2017). arXiv: 1703.07684. URL: http://arxiv.org/abs/1703.07684.
[66] Gabriel A Radvansky and Jeffrey M Zacks. *Event cognition*. Oxford University Press, 2014.
[67] Colin Lea Michael D Flynn René and Vidal Austin Reiter Gregory D Hager. "Temporal convolutional networks for action segmentation and detection". In: *IEEE International Conference on Computer Vision (ICCV)*. 2017.
[68] Alexander Richard and Juergen Gall. "Temporal action detection using a statistical language model". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 3131–3140.
[69] Alexander Richard, Hilde Kuehne, and Juergen Gall. "Weakly supervised action learning with RNN based fine-to-coarse modeling". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Vol. 1. 2. 2017, p. 3.
[70] Lauren L Richmond and Jeffrey M Zacks. "Constructing experience: Event models from perception to action". In: *Trends in cognitive sciences* 21.12 (2017), pp. 962–980.
[71] Mikel D Rodriguez, Javed Ahmed, and Mubarak Shah. "Action mach a spatio-temporal maximum average correlation height filter for action recognition". In: *2008 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE. 2008, pp. 1–8.
[72] Olga Russakovsky et al. "Imagenet large scale visual recognition challenge". In: *International Journal of Computer Vision (IJCV)* 115.3 (2015), pp. 211–252.
[73] Adam Santoro, David Raposo, David G Barrett, Mateusz Malinowski, Razvan Pascanu, Peter Battaglia, and Timothy Lillicrap. "A simple neural network module for relational reasoning". In: *Advances in Neural Information Processing Systems*. 2017, pp. 4967–4976.
[74] Fadime Sener and Angela Yao. "Unsupervised Learning and Segmentation of Complex Activities from Video". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018.
[75] Shikhar Sharma, Ryan Kiros, and Ruslan Salakhutdinov. "Action Recognition using Visual Attention". In: *Neural Information Processing Systems: Time Series Workshop*. 2015.

[76] Thomas F Shipley and Jeffrey M Zacks. *Understanding events: From perception to action*. Vol. 4. Oxford University Press, 2008.

[77] Karen Simonyan and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition". In: *arXiv preprint arXiv:1409.1556* (2014).

[78] Jingkuan Song, Lianli Gao, Zhao Guo, Wu Liu, Dongxiang Zhang, and Heng Tao Shen. "Hierarchical LSTM with adjusted temporal attention for video captioning". In: *Proceedings of the 26th International Joint Conference on Artificial Intelligence*. AAAI Press. 2017, pp. 2737–2743.

[79] Khurram Soomro, Haroon Idrees, and Mubarak Shah. "Action localization in videos through context walk". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2015, pp. 3280–3288.

[80] Khurram Soomro, Haroon Idrees, and Mubarak Shah. "Predicting the where and what of actors and actions through online action localization". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 2648–2657.

[81] Khurram Soomro and Mubarak Shah. "Unsupervised action discovery and localization in videos". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2017, pp. 696–705.

[82] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. "UCF101: A dataset of 101 human actions classes from videos in the wild". In: *arXiv preprint arXiv:1212.0402* (2012).

[83] Fillipe DM de Souza, Sudeep Sarkar, Anuj Srivastava, and Jingyong Su. "Spatially Coherent Interpretations of Videos Using Pattern Theory". In: *International Journal on Computer Vision (IJCV)* (2016), pp. 1–21.

[84] Nitish Srivastava, Elman Mansimov, and Ruslan Salakhutdinov. "Unsupervised Learning of Video Representations using LSTMs". In: *CoRR* abs/1502.04681 (2015). arXiv: 1502.04681. URL: http://arxiv.org/abs/1502.04681.

[85] Sebastian Stein and Stephen J McKenna. "Combining embedded accelerometers with computer vision for recognizing food preparation activities". In: *ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM. 2013, pp. 729–738.

[86] Chen Sun, Abhinav Shrivastava, Carl Vondrick, Rahul Sukthankar, Kevin Murphy, and Cordelia Schmid. "Relational Action Forecasting". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 273–283.

[87] Simon Thorpe, Denis Fize, and Catherine Marlot. "Speed of processing in the human visual system". In: *Nature* 381.6582 (1996), p. 520.

[88] Yicong Tian, Rahul Sukthankar, and Mubarak Shah. "Spatiotemporal deformable part models for action detection". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2013, pp. 2642–2649.

[89] Steven P Tipper, Cathy Lortie, and Gordon C Baylis. "Selective reaching: Evidence for action-centered attention." In: *Journal of Experimental Psychology: Human Perception and Performance* 18.4 (1992), p. 891.

[90] Du Tran and Junsong Yuan. "Max-margin structured output regression for spatio-temporal action localization". In: *Advances in Neural Information Processing Systems*. 2012, pp. 350–358.

[91] Du Tran and Junsong Yuan. "Optimal spatio-temporal path discovery for video event detection". In: *CVPR 2011*. IEEE. 2011, pp. 3321–3328.

[92] Jasper RR Uijlings, Koen EA Van De Sande, Theo Gevers, and Arnold WM Smeulders. "Selective search for object recognition". In: *International Journal of Computer Vision (IJCV)* 104.2 (2013), pp. 154–171.

[93] Jan C Van Gemert, Mihir Jain, Ella Gati, Cees GM Snoek, et al. "APT: Action localization proposals from dense trajectories." In: *BMVC*. Vol. 2. 2015, p. 4.

[94] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. "Attention is all you need". In: *Advances in Neural Information Processing Systems*. 2017, pp. 5998–6008.

[95] Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. "Anticipating visual representations from unlabeled video". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 98–106.

[96] Carl Vondrick, Abhinav Shrivastava, Alireza Fathi, Sergio Guadarrama, and Kevin Murphy. "Tracking emerges by colorizing videos". In: *Proceedings of the European conference on computer vision (ECCV)*. 2018, pp. 391–408.

[97] Jacob Walker, Abhinav Gupta, and Martial Hebert. "Patch to the future: Unsupervised visual prediction". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2014, pp. 3302–3309.

[98] Hongxing Wang, Junsong Yuan, and Ying Wu. "Context-aware discovery of visual co-occurrence patterns". In: *IEEE Transactions on Image Processing* 23.4 (2014), pp. 1805–1819.

[99] Limin Wang, Yu Qiao, and Xiaoou Tang. "Video action detection with relational dynamic-poselets". In: *European conference on computer vision*. Springer. 2014, pp. 565–580.

[100] Junyuan Xie, Ross Girshick, and Ali Farhadi. "Unsupervised deep embedding for clustering analysis". In: *International Conference on Machine Learning (ICML)*. 2016, pp. 478–487.

[101] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. "Show, attend and tell: Neural image caption generation with visual attention". In: *International Conference on Machine Learning*. 2015, pp. 2048–2057.

[102] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. "Xlnet: Generalized autoregressive pretraining for language understanding". In: *Advances in Neural Information Processing Systems*. 2019, pp. 5754–5764.

[103] Jeffrey M Zacks, Todd S Braver, Margaret A Sheridan, David I Donaldson, Abraham Z Snyder, John M Ollinger, Randy L Buckner, and Marcus E Raichle. "Human brain activity time-locked to perceptual event boundaries". In: *Nature neuroscience* 4.6 (2001), pp. 651–655.

[104] Jeffrey M Zacks and Joseph P Magliano. "Film, narrative, and cognitive neuroscience". In: *Art and the senses* 435 (2011), p. 454.

[105] Jeffrey M Zacks, Nicole K Speer, Khena M Swallow, Todd S Braver, and Jeremy R Reynolds. "Event perception: a mind-brain perspective." In: *Psychological Bulletin* 133.2 (2007), p. 273.

[106] Jeffrey M Zacks and Barbara Tversky. "Event structure in perception and conception." In: *Psychological Bulletin* 127.1 (2001), p. 3.

[107] Jeffrey M Zacks, Barbara Tversky, and Gowri Iyer. "Perceiving, remembering, and communicating structure in events." In: *Journal of experimental psychology: General* 130.1 (2001), p. 29.

[108] Mengmi Zhang, Keng Teck Ma, Joo Hwee Lim, Qi Zhao, and Jiashi Feng. "Deep future gaze: Gaze anticipation on egocentric videos using adversarial networks". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 4372–4381.

[109] Richard Zhang, Phillip Isola, and Alexei A. Efros. *Colorful Image Colorization*. 2016. arXiv: 1603.08511 [cs.CV].

[110] Gao Zhu, Fatih Porikli, and Hongdong Li. "Tracking randomly moving objects on edge box proposals". In: *arXiv preprint arXiv:1507.08085* (2015).

Biographies

Ramy Mounir is a Ph.D. candidate in the Computer Science and Engineering department at the University of South Florida (USF) in Tampa. He received his B.Eng and M.S. degrees in Mechanical Engineering from USF in 2015 and 2018, respectively. He graduated Summa Cum Laude and received the Outstanding Graduate Award in 2015. He also received his Robotics Graduate Certificate from USF in 2018. He is the recipient of the Early Innovation Award from Intel Corporation. His research interests include self-supervised learning of hierarchical representations of objects and events, implementing perceptual and cognitive theories using computational deep learning methods and predictive models.

Sathyanarayanan N. Aakur is an Assistant Professor of Computer Science at Oklahoma State University. He received the B.Eng. degree in Electronics and Communication Engineering from Anna University, Chennai in 2013. He received the M.S. degree in Management Information Systems and the Ph.D. degree in Computer Science from the University of South Florida, Tampa, in 2015 and 2019, respectively. His research interests include self-supervised learning, commonsense reasoning for visual understanding, and deep learning applications for genomics.

Sudeep Sarkar is a Professor and Chair of Computer Science and Engineering and the Associate Vice President for Special Programs at the University of South Florida in Tampa. He received his B.Tech. degree from the Indian Institute of Technology, Kanpur, M.S. and Ph.D. degrees in Electrical Engineering from The Ohio State University, Columbus. He has more than 25-year expertise in computer vision and pattern recognition algorithms and systems, holds ten U.S. patents, licensed technologies, and has published high-impact journal and conference papers. He is a Fellow of the AAAS, IEEE, IAPR, AIMBE, and NAI. He has served on many journal boards and is currently the Editor-in-Chief for Pattern Recognition Letters.