Summarizing While Recording: Context-Based Highlight Detection for Egocentric Videos

Yen-Liang Lin[1], Vlad I. Morariu[2], and Winston Hsu[1]

[1]National Taiwan University, Taipei, Taiwan

[2]University of Maryland, College Park, MD, USA

yenliang@cmlab.csie.ntu.edu.tw,morariu@umiacs.umd.edu,whsu@ntu.edu.tw

Abstract

In conventional video summarization problems, contexts (e.g., scenes, activities) are often fixed or in a specific structure (e.g., movie, sport, surveillance videos). However, egocentric videos often include a variety of scene contexts as users can bring the cameras anywhere, which makes these conventional methods not directly applicable, especially because there is limited memory storage and computing power on the wearable devices. To resolve these difficulties, we propose a context-based highlight detection method that immediately generates summaries without watching the whole video sequences. In particular, our method automatically predicts the contexts of each video segment and uses a context-specific highlight model to generate the summaries. To further reduce computational and storage cost, we develop a joint approach that simultaneously optimizes the context and highlight models in an unified learning framework. We evaluate our method on a public Youtube dataset, demonstrating our method outperforms state-of-the-art approaches. In addition, we show the utility of our joint approach and early prediction for achieving competitive highlight detection results while requiring less computational and storage cost.

1. Introduction

There has been enormous growth in egocentric videos, which are extremely unstructured and can vary in length from a few minutes to a few hours. Thus, it is becoming increasingly important to derive an efficient algorithm that extracts a brief summary of these videos to enable further browsing or indexing of such large-scale data. Severe camera motion, varied illumination conditions, and cluttered background in egocentric videos make it difficult to use shot detection algorithms to find important key-frames,

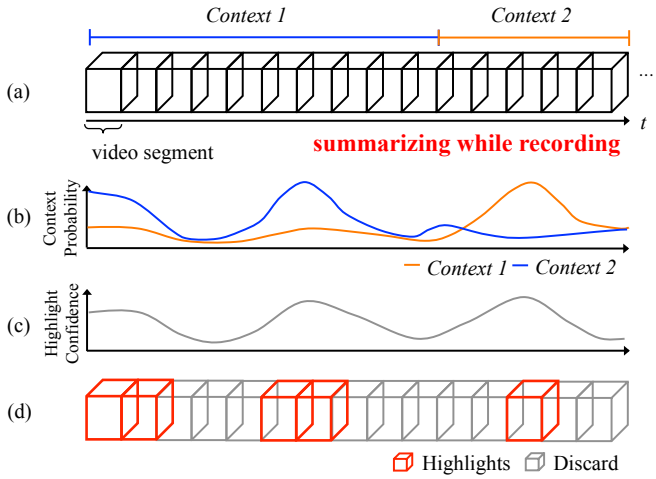

Figure 1. We propose a context-based highlight detection approach that detects highlight segments based on the predicted contextual information, and generates the summaries immediately without watching the whole video sequences. The highlight and contextual model are trained using structured SVM, where different learning strategies are proposed. Our method can handle continuous events with multiple contexts, which are ubiquitous in egocentric videos.

which are commonly used in previous video summarization approaches [23, 21, 4]. As a result, there is a growing interest [17, 16, 22, 24] in analyzing and summarizing egocentric videos captured by wearable cameras (e.g., Google Glass, GoPro) in unconstrained environments.

Summarizing egocentric videos is different from conventional video summarization problems at least in two ways: First, they often include a variety of scene types, activities, and environments, as users can take the cameras anywhere. Therefore, we cannot directly apply existing video summarization methods specifically designed for a certain context or structure (e.g., surveillance, news and sport videos) to egocentric domains. Also, unlike taking photos, the ego-

centric camera is always on, which makes it difficult for users to manually specify the context label of each video event. Second, mobile devices only have limited memory storage and computing power. For example, Google Glass can operate approximately 45 minutes during continuous use and only has 2 GB RAM and 12 GB storage. Therefore, it is impractical to store whole video sequences and perform summarization afterwards. A better strategy would be to only summarize and save the most important and informative content, discarding uninformative content while recording.

Specifically, we aim at developing a method that can summarize informative content in unconstrained videos and requires less power and memory storage to meet the constraints in the mobile devices. Recent work [24] proposed an online video summarization method that removes redundant video footage based on signal reconstruction errors and group sparse coding framework. However, these video summaries might not be semantically meaningful, e.g., unseen backgrounds can be selected as highlights based on low-level reconstruction error. Different from their approach, we propose a context-based highlight detection method that leverages contextual information to select informative segments. Our method sequentially scans the video stream, estimates the context label of each video segment and detects highlights (using the inferred context label to select corresponding highlight models) (ref. Fig. 1). Unlike [24], where a not-seen-yet background can have high construction errors and be mistakenly selected as highlights, our method will not highlight a background as it will have a low highlight confidence value under a certain context with our proposed constraints. We use structured SVM to learn highlight and context models, where different learning strategies are proposed, i.e., optimizing both tasks independently or jointly. The proposed approach is able to perform early prediction, inferring the most probable class label at an early stage, and using the class-specific model to summarize video highlights, further reducing both computing power and storage space.

Experiments demonstrate several key properties of our models. First, our approach successfully performs video highlight detection, despite having no advance knowledge of the context labels. In particular, we show superior performance on a public YouTube dataset [19] over other baseline methods, demonstrating the advantages of leveraging contextual information and using more powerful highlight models. Second, our joint approach can achieve competitive highlight detection results, while requiring less computational and storage cost. Third, our method predicts the context label accurately at the segment-level, which allows us to estimate the contextual information in an early stage and summarize only for the predicted class model to further reduce the computational cost.

The main contributions of this work include:

- We propose a context-based highlight detection method to summarize important video highlights, which achieves better performance compared to state-of-the-art approaches on a public YouTube highlight dataset [19].

- We summarize video highlights immediately without watching the whole video sequence, resolving the problem of limited memory and computing power on wearable cameras.

- We explore different learning strategies in a structured SVM framework for learning the highlight and context models; we show the utility of the joint and early prediction approach for achieving competitive highlight detection results while requiring less computational and storage cost.

Our approach can handle continuous events where multiple contexts occur within the one video (we predict the contexts and highlights at the segment-level). This is important because context commonly changes in egocentric videos and done fully automatically without any manual label information at testing time.

2. Related Work

Early video summarization methods mainly focus on edited and structured videos, e.g., movies, news, and sports [1, 12] and often extract key-frames by shot detection algorithms [23, 21, 4]. However, user-generated video data, such as egocentric or YouTube-style videos, typically do not contain clear shot boundaries and often include a variety of scene types, activities and environments, which makes prior methods not directly applicable.

Recently, [11, 14, 3] summarize a egocentric video by leveraging high-level object cues (e.g., people, hands, objects). These schemes perform well when the objects are clearly present. However, some important frames may only have small object or even no objects. Others use web-image priors learned from large image collections to select important frames as summaries [8, 22]. Adopting image priors may lead to generalization problems as mentioned in [8], e.g., large variety and inconsistency in appearance for food items. [2] uses the self-expressiveness property of summarization in a group sparse coding framework to select highlights. [24] proposes an online video highlighting approach removes redundant video footage based on a low-level signal reconstruction and group sparse coding framework. Namely, if an input video segment has high reconstruction error, this indicates that it includes unseen content from previous video data and will be selected as highlight.

Kitani et al. [9] adopt unsupervised learning for video segmentation. However, the drawback of these types of approaches is that generated video summaries may have no semantic meaning.

Some approaches exploit temporal segmentation to extract a video summary [13, 5, 16]. Kang Li et al. [13] decompose a video into sub-segments based on the different camera motion types and extract multiple features related to motion and scale at each segment to construct a videography dictionary. Gygli et al. [5] segment videos into a set of super frames and select an optimal subset to create an summaries. Poleg et al. [16] suggest temporal segmentation of an egocentric video by leveraging motion cues (e.g., wearer's head motion).

If the event category is already known, the summarization can be also category-specific. Potapov et al. [17] propose a category-specific video summarization approach for producing higher quality video summaries. The authors perform temporal segmentation to generate segments and adopt SVM for assigning an importance score to each segment. Sun et al. [19] rank domain-specific video highlights by analyzing online edited videos and formulate the problem into a latent ranking model. However, the assumption that the event category is given in advance is not appropriate for egocentric videos as it is hard for users to specify the class label for each time interval in always-on wearable cameras. Also, it is infeasible to save whole videos (e.g., [5, 17]) into the limited amounts of available memory on wearable cameras.

Different from prior methods, we propose a context-based highlight detection approach that summarizes the most important and interesting content for egocentric videos. Our method automatically predicts the contextual label for each video segment, and can therefore handle continuous changing background contexts. We use structured SVM to learn our highlight and context model, improving on prior online summarization methods [24] while immediately summarizing the content without watching the whole video sequence. We further show that joint modeling and early prediction can reduce computational and storage cost while achieving competitive results.

3. Method Overview

Our system consists of two main phases. The offline phase consists of learning our discriminative models for highlight detection and context prediction by a structured SVM framework with different learning strategies (see Sec. 4). The online phase consists of evenly partitioning the input videos into temporal segments and processing each segment sequentially to generate summaries while discarding uninformative content such as background clutter. Context labels are automatically inferred during inference to provide additional category knowledge without manually given in advance (see Sec.5).

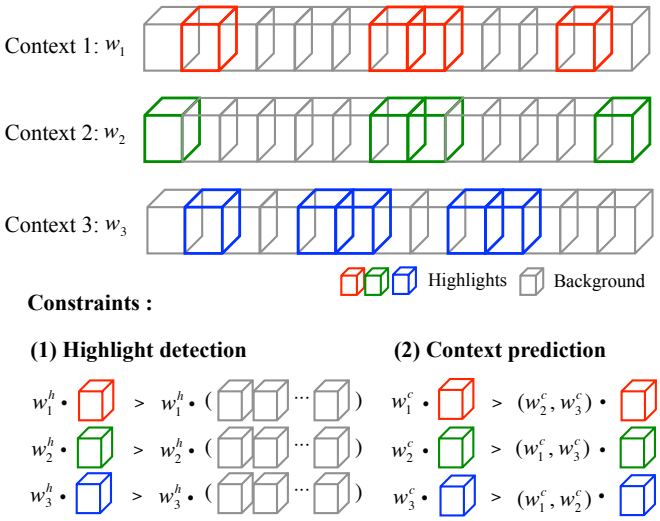

Figure 2. Proposed constraints for learning context and highlight models in a structured SVM framework. The highlight detection constraints require that the highlight segments have higher values than the other segments within the same video sequence; the context prediction constraints require that segments with the correct context label have higher value than other context labels.

4. Structured Learning for Highlight and Context Models

We adopt a structured SVM (SSVM) formulation using a margin rescaling loss for learning highlight and context models. Two learning strategies are investigated, sequential and joint SSVM learning. The first one learns two disjoint sets of weights independently, one for detecting video context and another one for determining highlights. The second one jointly optimizes both tasks and have a single set of weights being able to simultaneously detect highlights and estimate context labels.

Formally, given the training data in the form of $\{(X_i, y_i)\}_{1,...,N}$ where X_i denotes a video sequence and $y_i = (y^c, y^s, y^f) \in Y$ is a tuple of labels. $y^c \in \{1, ..., k\}$ is the context label, $y^s \in [0, 1]$ is the confidence score for the highlight, and $y^f \in \{1_i, ..., T_i\}$ is the temporal position for the highlight segment in the video sequence X_i [1].

Sequential SSVM. A structured SVM learns two disjoint sets of model weights with the constraints that highlight segments should have higher scores than other segments (second constraint in Eq. 1), and correct context (class) labels achieve higher scores than the incorrect ones (first constraint in Eq. 1) (see Fig. 2). We employ loss functions ($\Delta_{context}$ and $\Delta_{highlight}$) to penalize the outputs

[1]For a video with several highlights, it can be represented by multiple data sequences, e.g., $(X_1, y_1), (X_2, y_2), ..., (X_m, y_m)$, where $X_1, X_2, ..., X_m$ correspond to the same video sequence.

y (for context prediction) and y' (for highlight detection) that deviate from the correct outputs y_i. These constraints enable our models to select the correct contexts and detect highlights during testing, since they should higher values with our proposed constraints.

$$\begin{aligned}&\min_{w^h,w^c,\xi^h,\xi^c} \left\|w^h\right\|_2^2 + \|w^c\|_2^2 + \lambda_1 \sum_{i=1}^{N} \xi_i^h + \lambda_2 \sum_{i=1}^{N} \xi_i^c \\ &\text{sb.t.} \forall y \in \{y^f = y_i^f, y^c \neq y_i^c\}, \forall y' \in \{y'^f \neq y_i^f, y'^c = y_i^c\} : \\ &\langle w^c, \Psi(X_i, y_i)\rangle - \langle w^c, \Psi(X_i, y)\rangle + \xi_i^c \geq \Delta_{context}(y_i, y) \\ &\left\langle w^h, \Psi(X_i, y_i)\right\rangle - \left\langle w^h, \Psi(X_i, y')\right\rangle + \xi_i^h \geq \Delta_{highlight}(y_i, y')\end{aligned} \tag{1}$$

Here λ_1 and λ_2 are the parameters controlling the trade-off between a large margin and constraint violation. The slack variables ξ_i^h and ξ_i^c corresponding to X_i penalizes the constraint violation. For a more concise notation, the features are stacked for learning multi-class models, $\Psi(X_i, y_i) = \Psi(y^c) \otimes \Psi_X(X_i) = \big[0, ..., 0|...|\Psi_X(X_i, y^f)|...|0, ..., 0\big]$. Here $\Psi_X(X_i, y^f)$ is a feature map extracting the segment features at the temporal position y^f. The vector $\Psi(y^c)$ has a 1 in the place of current class y^c and 0 otherwise. Finally, $w^c = \{w_1^c, ..., w_k^c\}$ are model parameters for context prediction (*Context SSVM*) and $w^h = \{w_1^h, ..., w_k^h\}$ are model parameters for highlight detection (*Highlight SSVM*). $\left\langle w^{\{h,c\}}, \Psi(X_i, y_i)\right\rangle$ represents the inner product between the model parameters and the feature descriptors.

The inequality constraints of Eq. 1 require the highlight segments to have higher values than other segments; segments with correct class labels should be relatively higher than any other class by a large margin defined by the loss functions:

$$\begin{aligned}\Delta_{context}(y_i, y) &= \alpha \cdot [y_i^c \neq y^c] \\ \Delta_{highlight}(y_i, y') &= \beta \cdot (y_i^s - y'^s)[y_i^f \neq y'^f]\end{aligned} \tag{2}$$

where $[.]$ is the Iverson bracket notation. The variables α and $\beta \cdot (y_i^s - y'^s)$ are the losses for context prediction and highlight detection respectively. λ_1 and λ_2 are automatically selected (both are set to 10) to maximize the highlight detection and context prediction performance for all classes in the training set respectively.

Joint SSVM. As motivated before, we attempt to use more compact models to further reduce the computational and storage cost while maintaining the highlight detection performance. The intuition is that both tasks are correlated and can be jointly optimized. For example, a taking off segment in surfing is considered informative (more salient than other segments) and discriminative (different from other contexts). In other words, informative in a certain degree implies that it is also discriminative, thus both constraints are related and can be jointly trained. For this purpose, we jointly optimize both tasks using a unary set of weights in a SSVM learning framework, where the objective function is re-formulated as:

$$\begin{aligned}&\min_{w,\xi} \|w\|_2^2 + \lambda \sum_{i=1}^{N} \xi_i, \\ &\text{sb.t.} \forall y \in \{y^f \neq y_i^f \vee y^c \neq y_i^c\}, \\ &\langle w, \Psi(X_i, y_i)\rangle - \langle w, \Psi(X_i, y)\rangle + \xi_i \geq \Delta(y_i, y)\end{aligned} \tag{3}$$

The notations are similar to the definitions in Eq. 1. We combine highlight w^h and context w^c models into a single set of models w and define a new loss function as:

$$\Delta(y_i, y) = \begin{cases} \alpha & \text{if} \quad y_i^c \neq y^c \\ \beta \cdot (y_i^s - y^s) & \text{if} \quad y_i^c = y^c, y_i^f \neq y^f \end{cases} \tag{4}$$

The terms α and β are selected by maximizing the highlight detection accuracy on the training set.

The optimization problem in Eq. 3 contains a number of constraints, namely $|Y|$ inequalities per training sample. Therefore, we adopt a cutting plane method [7] for constraint generation. This approach searches for the best weight vector and the set of active (most violated) constraints simultaneously in an iterative manner. While searching for the most violated constraints during optimization, we avoid using video segments from two different videos, since scores might not be directly comparable across videos. The label $\bar{y}_i$ is selected by maximizing the following equation:

$$\bar{y}_i = \arg\max_{y \in Y} \Delta(y_i, y) + \langle \Psi(X_i, y), w\rangle - \langle \Psi(X_i, y_i), w\rangle \,, \tag{5}$$

In our preliminary experiments, we observed that even if the context prediction constraints for non-highlights are not included, the learned models still generalize to non-highlights. The reason might be that both highlights and non-highlights share a common structure, so that what helps discriminate between highlight segment categories also helps discriminates between non-highlight segment categories. Therefore, we only keep the context prediction constraints for highlights, which is more efficient (due to fewer constraints) for parameter optimization in the structured SVM framework.

5. Online Context Prediction and Highlight Detection

Given the learned models, our method sequentially scans the input video $X = \{x_1, ..., x_T\}$, predicts the context labels for each video segment at time t, and generates the highlight confidence scores based on contextual labels. The final summary is generated by thresholding these confidence scores (here, we give a way to generate the summary for real situation. For comparing with the baseline methods (e.g., [19], [24]), we follow the same evaluation criteria as in [19]. Given the predicted highlight confidence scores and ground truth, we compute the AP scores, which capture performance for all possible thresholds.)

We only show the method for joint SSVM approach, while sequential SSVM follows a similar procedure, except using different sets of highlight and context models (i.e., sequentially applying context and highlight models). For each video segment at time t, the context prediction score for each class is defined as

$$score(c,t) = \langle w_c, \Psi(X,t) \rangle, c = \{1, ..., k\} \quad (6)$$

Then, the highlight confidence scores are given by

$$v(t) = \max_c score(c,t) \quad (7)$$

Since our model jointly optimizes both context prediction and highlight detection, we can compute the highlight confidences by selecting the max scores by different context classes. The computational and storage costs are less than sequential SVM method, as we do not need to use different sets of weights to predict video context and detect highlights.

Early prediction. To further reduce the computational and storage cost, we investigate early prediction in preliminary experiments. Our method accurately predicts the context label at segment-level, which enables us to use past prediction results to infer the most probable context label at an early stage and only use the context-specific model to summarize video highlights. To do this, we compute cumulative scores that integrate all past class prediction scores and use the difference between the max and second max of the cumulative scores to infer the target class label early:

$$\begin{aligned} d(t) &= \Gamma(\hat{c},t) - \Gamma(\tilde{c},t) > \varepsilon \\ \hat{c} &= \max_c \Gamma(c,t), \tilde{c} = \max_{c \neq \hat{c}} \Gamma(c,t) \end{aligned} \quad (8)$$

where $\Gamma(c,t) = \Gamma(c,t-1) + score(c,t)$ accumulates the past prediction results. If the difference exceeds a pre-set threshold ε (the effect of different threshold values is investigated in Sec. 6.4), we use the context-specific model (i.e., $\hat{c}$) to generate highlights, significantly reducing the computational cost by avoiding the application of all context models to compute the prediction scores (see Eq. 6)

Here, we assume the transition points for different context are given for this pilot study, so we know where to initialize the cumulative scores. We can utilize existing methods to find the transition points, e.g., the method [14] categorized each frame into different classes, e.g., static, in transit or moving the head and finds sub-events with these classes or re-initialize the cumulative score automatically if the highlight confidence is low for a period, i.e., the camera wearer probably transits to other context, and current model can not detect highlights. Then we reinitialize the cumulative score and use all context models to find the correct context by Eq. 8, but leave this to future work. For handling the unseen contexts, we can use our method when the context is known and adopt existing context-independent methods otherwise. Thus, our method will not reduce the quality for this case.

6. Experiments

6.1. Dataset and Evaluation Criteria

We evaluate our method on a public YouTube Highlight dataset [19] (totally 1430 minutes) [2], which includes six event categories: "skating", "gymnastics", "dog", "parkour", "surfing", and "skiing". As our method predicts the contexts and selects highlights at segment-level, the evaluation results on separate categories (contexts) are as same as on the concatenation of multiple categories. Thus, we report our results on each event category. The ground truth highlights are annotated by Amazon Mechanical Turkers. Segments annotated more than two times as highlights are considered ground truth highlights (three times for parkour and skiing). We use the training and testing set, and evaluation criteria as in [19]. We calculate the average precision (AP) for highlight detection, and compute mean average precision (mAP) to summarize the performance of all videos. The AP is computed by using the ground truth labels (1 for highlights; 0 for non-highlights) and estimated highlight confidence scores.

6.2. Feature Representation

We define a video segment as 100 frames evenly sampled across each raw video as in [19]. We extract STIP features based on an off-the-shelf tool [10]. Experiments show that STIP works well on this dataset, and requires less storage and computational cost compared to more sophisticated features, e.g., dense trajectory [20] and CNN features [6, 18]. We reduce the feature dimension to 64 by principal component analysis (PCA). Video features with the same class are used to learn a Gaussian mixture model (GMM) with 120 components to generate Fisher vectors (FV) [15] with a final feature dimension of 15360. Power and L2 normalization schemes are also applied.

6.3. Highlight Detection Results

We compare our methods with several state-of-the-art methods: Sun et al. [19], Zhao et al. [24] and HOG/HOF+FV+Binary SSVM on Youtube Highlight dataset [19], where Zhao et al's method has been shown to outperforms the other baseline approaches, e.g., evenly spaced segments, K-means clustering and DSVS algorithm proposed in [2]. For Zhao et al's method, we fix the dictionary size to 200 following the same settings in the original paper and use the reconstruction errors from ℓ_1/ℓ_2 group

[2] To evaluate and analyze whether context benefits highlight detections requires both highlight and context annotations. To our best knowledge, this is the largest publicly available one.

Category	Method	skating	gymnastics	surfing	dog	parkour	skiing	Ave. mAP
Given context label	Turk-rank [19]	63%	41%	61%	49%	50%	50%	∼52%
	Latent-rank [19]	62%	40%	61%	60%	61%	36%	∼53%
	Binary SSVM	61%	33%	59%	70%	40%	42%	51%
	Highlight SSVM (ours)	**57%**	**41%**	**58%**	**70%**	**52%**	**66%**	**57%**
No context label	Zhao et al. [24]	51%	30%	37%	27%	55%	51%	42%
	Random Context	22%	24%	41%	37%	47%	39%	35%
	Sequential SSVM (ours)	**52%**	**33%**	**58%**	**71%**	**44%**	**54%**	**52%**
	Joint SSVM (ours)	**54%**	**44%**	**56%**	**69%**	**47%**	**59%**	**55%**
	Joint SSVM (GT Class) (oracle)	52%	45%	56%	69%	57%	59%	56%

Table 1. Video highlight detection comparison. We report the results of our method and baseline methods on a public Youtube highlight dataset [19]. We categorize these methods into two groups: *given context label* and *no context label*, depending on whether the class label is provided at test time. Our *Sequential SSVM* and *Joint SSVM* detect video highlights based on the estimated contextual information, and significantly outperforms Zhao et al.'s method [24], which detects the highlights based on the low-level reconstruction errors from group sparse coding framework. Our highlight model (*Highlight SSVM*) shows better results to [19] and Binary SSVM, demonstrating the effectiveness of using margin rescaling loss in SSVM learning. We also report the results with random context labels (*Random Context*) and ground truth context label for our joint approach (*Joint SSVM (GT class)*), verifying the importance of using contextual information for video summarization and showing that our approach doest not cause the performance degradation compared to the ground truth (oracle) case.

sparse coding system to compute the average precision for each video. The threshold (denoted by ε_0 in [24]) for controlling the summary length is selected to maximize the performance on the dataset. For Sun et al's method, we compare two variants of their approach: Turk-rank and Latent-rank. Turk-rank is trained with annotations from turkers while Latent-rank is trained by using noisy data harvested online. We directly use the numbers reported in their paper. For Binary SSVM, we use highlight segments of each event category as positives and other segments as negatives and feed them into a SSVM framework using 0/1 loss. We categorize these methods into two groups: *given context label* and *no context label*, based on whether a context label is provided at test time or not.

Table 1 summarizes the overall highlight detection results for different methods. Our approach (*Sequential SSVM* and *Joint SSVM*) significantly outperforms Zhao et al.'s method [24], as their video summaries might not have semantic meaning, e.g., unseen backgrounds can also be selected as highlights based on low-level reconstruction error. Instead, our approach predicts the most likely context label at each video segment and uses it to generate the highlight confidence score. By taking into account the context of the video being observed, our summaries capture more informative and interesting contents than [24]. In addition, our highlight model (*Highlight SSVM*) also outperforms [19] and Binary SSVM, demonstrating the effectiveness of using margin rescaling loss for learning more powerful highlight models. Note that our method uses less powerful feature descriptors than [19] (STIP [10] vs. dense trajectory [20]) and with fewer feature dimensions (15360 vs. 26000).

We evaluate two variants of our method, *Sequential SSVM* and *Joint SSVM*, where we optimize the highlight and context models independently and jointly respectively. Both achieve superior performance compared to the state-of-the-art method [24]. Interestingly, our joint modeling approach even shows slightly better performance than sequential approach and requires fewer model parameters. The reason might be that highlight detection and context prediction are correlated, therefore additional constraints from other tasks benefit to the original models and improve the results.

We also report results with random context label *Random Context*, where the context label of each segment is randomly selected and ground truth context label *Joint SSVM (GT class)*, where we use the ground truth context label for each video segment as the oracle. The results indicate that a random context label does not perform very well, demonstrating the importance of using context label information for video summarization. Moreover, our joint approach doest not lead to performance degradation compared to the ground truth case, despite predicting video categories automatically. The reason is that misclassified examples often appear in the non-highlight segments. Therefore, even if we use the wrong context model for non-highlight segments, it does not affect the final highlight detection results notably, i.e., still have low confidence values in other highlight models.

We also evaluate the classification accuracy (i.e., context prediction accuracy) of our joint modeling approach. The ground truth context label of each video segment is obtained from the event category for the whole video sequence. We compute classification accuracy by measuring the label differences (including both highlight and non-highlight segments) between our prediction results and ground truth labels. The results show that joint model *JointSSVM* (average accuracy = 81%) achieves slightly better results compared to *ContextSSVM* (average accuracy = 78%), showing the feasibility of our joint modeling approach.

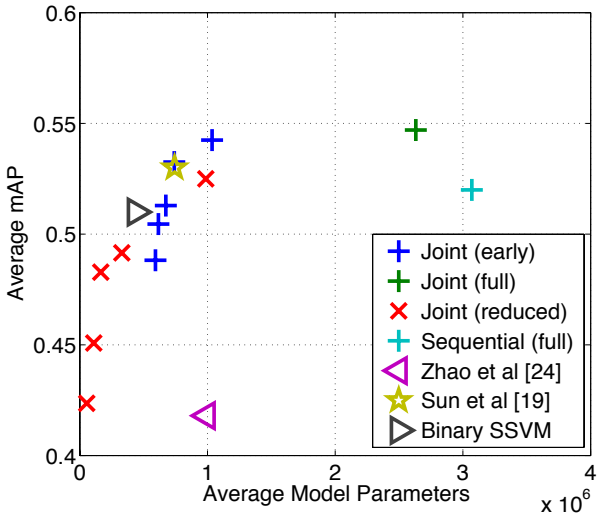

Figure 3. Average model parameters versus average mAP. The points close to the left-corner are more cost-effective, having fewer model parameters while achieving higher average mAP. Our method achieves better highlight detection performance while using less model cost compared to the state-of-the art method ([24]). Note that [19] and Binary SSVM requires additional context label information during testing, while ours are not. See Sec. 6.4 for more details.

6.4. Computational Cost

Having discussed the power of our joint approach for highlight detection and class prediction, we also investigate computational cost of our method against baseline approaches. We measure computational cost by the average number of model parameters used per video [3]. Fig. 3 shows the computational cost with different methods. For Sun et al.'s method [19] and HOG/HOF+FV+Binary SSVM, highlight confidence scores are computed by the inner product of the model and feature descriptors for every video segment. Therefore, the computing cost for each video can be computed as $\#parameters \times \#segments$. For Zhao et al.'s method [24], it is not straightforward to compute exact number of model parameters used for group sparse coding and online dictionary learning framework. Therefore, we approximate the computational cost of their method by $\mathbf{D} \in R^{m\times k} \times \#segments + (\mathbf{P}_t \in R^{k\times k} + \mathbf{Q}_t \in R^{m\times k}) \times \#updates$, where $\mathbf{D}$ is the dictionary, $\mathbf{P}_t$ and $\mathbf{Q}_t$ are matrices used for online dictionary learning. The parameter m denotes feature dimension (i.e., 162 for HOG/HOF) and k denotes codebook size (i.e., 200). See [24] for more details.

For our sequential approach *(Sequential (full))*, we apply both context and highlight models to compute the highlight confidence scores. Therefore, the computing cost is $(\#classes{+}1) \times \#parameters \times \#segments$, while the cost for joint approach *(Joint (full))* is $\#classes \times \#parameters \times \#segments$ (the storage cost of joint approach is half to sequential approach). We also experiment our method with fewer feature dimensions *(Joint (reduced))*. For early prediction approach *(Joint (early))*, we predict the class label early and only use the context-specific model to compute highlight confidence scores. Therefore, the model cost can be computed as

$$\begin{aligned} &\#classes \times \#parameters \times \eta + \\ &1 \times \#parameters \times (\#segments - \eta), \end{aligned} \tag{9}$$

where η denotes the decision point for early class prediction controlled by the threshold (ε) on the difference values in Eq. 8. Different thresholds show different result points in Fig. 3. The results show that our method is very cost-effective, achieving better highlight detection performance (average mAP) and requiring less model usage compared to state-of-the-art method method [24]. Note that for [19] and binary SSVM, their context label is given during testing time while ours are not. Moreover, our method uses linear models to detect video highlights, which is much more efficient than solving a ℓ_1/ℓ_2 signal decomposition problem as in [24].

6.5. Conclusions

In this work, we propose a context-based highlight detection approach that generates video highlights immediately without watching the whole videos. Experiments show that our method outperforms baseline approaches, showing the utility of leveraging contextual information. To learn our models, we utilize a structured prediction framework, where different learning strategies are investigated. We demonstrate that our joint modeling and early approach can achieve competitive results while requiring less computing power and storage.

[3] The source codes for baseline methods are not publicly available, thus we approximate the complexity by average model parameters.

References

[1] N. Babaguchi. Towards abstracting sports video by highlights. In *ICME*, 2000. 2

[2] Y. Cong, J. Yuan, and J. Luo. Towards scalable summarization of consumer videos via sparse dictionary selection. *TMM*, 2012. 2, 5

[3] A. Fathi, A. Farhadi, and J. M. Rehg. Understanding egocentric activities. In *ICCV*, 2011. 2

[4] D. B. Goldman, B. Curless, D. Salesin, and S. M. Seitz. Schematic storyboarding for video visualization and editing. In *SIGGRAPH*, 2006. 1, 2

[5] M. Gygli, H. Grabner, H. Riemenschneider, and L. V. Gool. Creating summaries from user videos. In *ECCV*, 2014. 3

[6] S. Ji, W. Xu, M. Yang, and K. Yu. 3d convolutional neural networks for human action recognition. *TPAMI*, 2013. 5

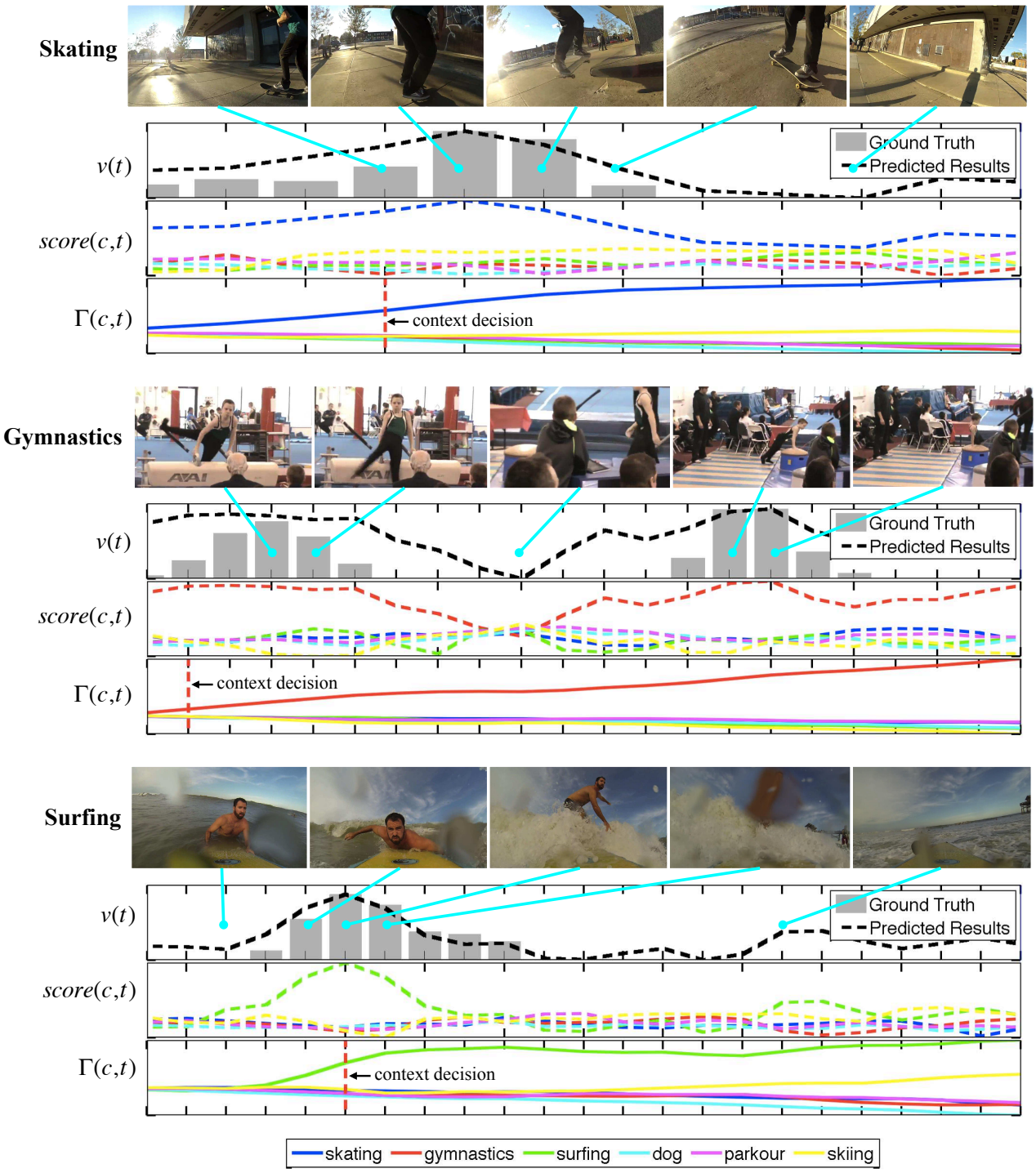

Figure 4. Example results of our method on three different categories. For each category, the first row shows example video segments for each input video sequence. The second row shows the highlight confidence scores $v(t)$, computed by the inferred context model (Eq. 7). The third row shows the scores for each context class, $score(c,t)$ (Eq. 6) and the last row shows the cumulative scores $\Gamma(c,t)$ (not normalized) for each context class. The results show that the proposed method can jointly detect the highlights and estimate the context labels accurately. Moreover, our context label prediction results demonstrate the ability for early context prediction to further reduce the computational cost.

[7] T. Joachims, T. Finley, and C.-N. J. Yu. Cutting-plane training of structural svms. *Machine Learning*, 2009. 4

[8] A. Khoslay, R. Hamidz, C.-J. Lin, and N. Sundaresanz. Large-scale video summarization using web-image priors. In *CVPR*, 2013. 2

[9] K. M. Kitani, T. Okabe, Y. Sato, and A. Sugimoto. Fast unsupervised ego-action learning for first-person sports videos. In *CVPR*, 2011. 3

[10] I. Laptev, M. Marszałek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *CVPR*, 2008. 5, 6

[11] Y. J. Lee, J. Ghosh, and K. Grauman. Discovering important people and objects for egocentric video summarization. In *CVPR*, 2012. 2

[12] B. Li and I. Sezan. Event detection and summarization in american football broadcast video. In *SPIE Storage ad Retrieval for Media Databases*, 2002. 2

[13] K. Li, S. Oh, A. G. Perera, and Y. Fu. A videography analysis framework for video retrieval and summarization. In *BMVC*, 2012. 3

[14] Z. Lu and K. Grauman. Story-driven summarization for egocentric video. In *CVPR*, 2013. 2, 5

[15] F. Perronnin, J. Sánchez, and T. Mensink. Improving the fisher kernel for large-scale image classification. In *ECCV*, 2010. 5

[16] Y. Poleg, C. Arora, and S. Peleg. Temporal segmentation of egocentric videos. In *CVPR*, 2014. 1, 3

[17] D. Potapov, M. Douze, Z. Harchaoui, and C. Schmid. Category-specific video summarization. In *ECCV*, 2014. 1, 3

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

[19] M. Sun, A. Farhadi, and S. Seitz. Ranking domain-specific highlights by analyzing edited videos. In *ECCV*, 2014. 2, 3, 4, 5, 6, 7

[20] H. Wang and C. Schmid. Action recognition with improved trajectories. In *ICCV*, 2013. 5, 6

[21] W.Wolf. Keyframe selection by motion analysis. In *ICASSP*, 1996. 1, 2

[22] B. Xiong and K. Grauman. Detecting snap points in egocentric video with a web photo prior. In *ECCV*, 2014. 1, 2

[23] H. Zhang. An integrated system for content-based video retrieval and browsing. *Pattern Recognition*, 1997. 1, 2

[24] B. Zhao and E. P. Xing. Quasi real-time summarization for consumer videos. In *CVPR*, 2014. 1, 2, 3, 4, 5, 6, 7