This WACV 2022 paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

Learning Temporal Video Procedure Segmentation from an Automatically Collected Large Dataset

Lei Ji *[1,2,3], Chenfei Wu *[3], Daisy Zhou *[4], Kun Yan[5], Edward Cui[4], Xilin Chen[1,2], Nan Duan[3]
[1] Institute of Computing Technology, Chinese Academy of Science, Beijing, China
[2]University of Chinese Academy of Sciences, Beijing, China
[3]Microsoft Research Asia, Beijing, China, [4]Bing Multimedia Team, Microsoft, China
[5]SKLSDE Lab, Beihang University, Beijing, China
{leiji,chewu,daisyz,edwac,nanduan}@microsoft.com, kunyan@buaa.edu.cn, xlchen@ict.ac.cn

Abstract

Temporal Video Segmentation (TVS) is a fundamental video understanding task and has been widely researched in recent years. There are two subtasks of TVS: Video Action Segmentation (VAS) and Video Procedure Segmentation (VPS): VAS aims to recognize what actions happen inside the video while VPS aims to segment the video into a sequence of video clips as a procedure. The VAS task inevitably relies on pre-defined action labels and is thus hard to scale to various open-domain videos. To overcome this limitation, the VPS task tries to divide a video into several category-independent procedure segments. However, the existing dataset for the VPS task is small (2k videos) and lacks diversity (only cooking domain). To tackle these problems, we collect a large and diverse dataset called TIPS, specifically for the VPS task. TIPS contains 63k videos including more than 300k procedure segments from instructional videos on YouTube, which covers plenty of how-to areas such as cooking, health, beauty, parenting, gardening, etc. We then propose a multi-modal Transformer with Gaussian Boundary Detection (MT-GBD) model for VPS, with the backbone of the Transformer and Convolution. Furthermore, we propose a new EIOU metric for the VPS task, which helps better evaluate VPS quality in a more comprehensive way. Experimental results show the effectiveness of our proposed model and metric.

1. Introduction

Coupled with the significant increase of content-based video data on the Internet, video analysis remains an intensely studied field, e.g., temporal video segmentation [27, 3], dense video caption [18, 26], visual grounding [18, 26]. *Temporal video segmentation* (TVS), which is a fundamental step in content-based video analysis, plays a key role in video analysis. YouTube[1] already released the video segmentation feature on several videos, most of which rely on the video creator to manually input the timestamps[2]. Although automatic segmentation is of high-demand for web applications, it is still a quite challenging research problem.

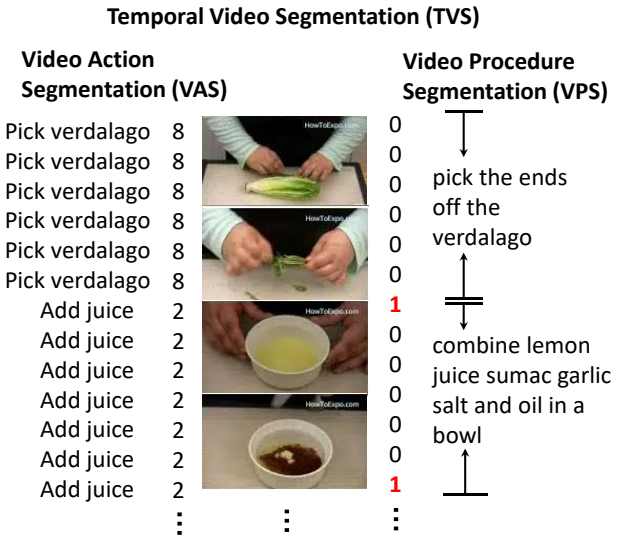

Figure 1. Difference between Video Action Segmentation (VAS) and Video Procedure Segmentation (VPS). VAS predicts action label for each frame while VPS predicts semantic boundaries for video procedure and can be viewed as pre-processing for video captioning.

"The goal of temporal video segmentation is to divide the video stream into a set of meaningful and manageable segments" [9]. To achieve this goal, two types of subtasks have been proposed in recent years, as shown in Fig. 1. The first subtask is *Video Action Segmentation* (VAS), which tem-

[1]https://www.youtube.com/watch?v=gCtLbNe78O0
[2]https://techcrunch.com/2020/05/28/youtube-introduces-video-chapters-to-make-it-easier-to-navigate-through-longer-videos/

porally divides a video into several segments and predicts an action for each segment. [21] proposed the 50Salads dataset, which provides a total of 52 actions for the cooking domain. Later, [24] proposed the COIN dataset, which provides a total of 180 actions for the multiple domains from YouTube. However, as the actions are pre-defined, the videos are inevitably annotated under the restriction of the action labels and it is hard to generalize to new actions. To remove this restriction and learn video temporal segments from pure visual evidence, [27] proposed the second subtask, *Video Procedure Segmentation* (VPS). Similar to the VAS task, the VPS task also temporally divides a video into several segments and each procedure segment is category-independent. The VPS task defines a natural procedure segmentation for real-world scenarios and has a high application value, such as automatically generating chapters for YouTube videos. However, there are few VPS datasets in the research field and the existing VPS datasets have limitations in diversity and scale. For example, [27] proposed the Youcook2 dataset, which only provides 2000 videos in the cooking domain.

To tackle these problems, we first introduce a new automatically collected dataset called TIPS for video procedure segmentation. Fig. 2 shows an example of this dataset. TIPS is a large and diverse dataset built from instructional videos on YouTube, which contains 63k videos and more than 300k procedure segments. It covers plenty of instructional videos from various domains, e.g. cooking, health, beauty, parenting, gardening. We then propose a novel Multi-modal Transformer Gaussian Boundary Detection (MT-GBD) model for the VPS task. The MT-GBD model uses a visual transformer to encode the video, a language transformer to encode the transcript, and a cross transformer to encode the interaction between visual and language followed by a temporal convolution network, and finally detects the segment key point with Gaussian boundary. Furthermore, to evaluate the procedure segmentation performance of the model in a human-sense manner, we introduce a new metric, called Experienced IOU (EIOU), specifically for video procedure segmentation.

In summary, our contributions are as follows:

(1)We introduce a large, open-domain, category-independent dataset for video procedure segmentation.

(2)We propose a novel Multimodal Transformer Gaussian Boundary Detection (MT-GBD) model with detailed ablation for video procedure segmentation.

(3)We design a new Experienced IOU metric for video procedure segmentation, which comprehensively considers procedure length, precision, and recall.

2. Related Work

2.1. Temporal Video Segmentation Datasets

Early datasets for temporal video segmentation focus on video action segmentation[6, 21, 11, 1, 24]. Most of them only involve a specific domain, such as cooking. Recently, [27] introduced video procedure segmentation task and proposed the Youcook2 dataset, which provides incontiguous video procedure segments. Tab. 1 gives a comparison of existing datasets that support temporal video segmentation.

0.12em0pt1pt **Dataset**	**Task**	**Samples**	**Domain**	**Actions**
GTEA	VAS	28	Cooking	7
50Salads	VAS	50	Cooking	17
Breakfast	VAS	1,989	Cooking	48
EPIC-KITCHENS	VAS	432	Cooking	149
COIN	VAS	11,827	Open	180
0em0.5pt0.5pt heightYoucook2	VPS	2,000	Cooking	-
TIPS(ours) 0.12em0pt1pt	VPS	**63,756**	Open	-

Table 1. Comparisons of existing datasets that support temporal video segmentation.

The differences between TIPS and existing datasets are four-folds. (1) *Scale*. TIPS is the largest temporal video segmentation dataset, and more specifically, the largest video procedure segmentation dataset. (2) *Diversity*. TIPS contains open-domain instructional videos on YouTube including cooking, health, beauty, parenting, gardening, etc. (3) *Contiguity*. Unlike in Youcook2 dataset, the segments are incontiguous while the segments in TIPS are contiguous. The contiguous segmentation better overviews the procedure of the whole video structure globally. (4) *Auto-generated*. We introduce a workflow to auto-generate the TIPS dataset from YouTube with guarantee of high quality. With the increase of videos uploaded to YouTube, it is feasible to collect more data automatically in the future.

2.2. Temporal Video Segmentation Methods

One formulation of this problem is frame-wise classification[5, 6, 4]. Previous methods[12, 23] relied on Hidden Markov Models (HMM). With the development of deep learning methods, RNNs and CNNs are widely used [19, 14, 15]. Recently, [3] proposed Multi-Stage TCN (MS-TCN), which uses dilated convolutions to capture long-range temporal dependencies. The other similar works are proposal based models including Deep Action Proposals[2, 10], Procnet[27]. Initially, the proposal-based approaches[2, 10] were proposed for overlapped and loosely-coupled event discovery instead of procedure segmentation. Later, Zhou etc.[27] refined the algorithm for video procedure segmentation. Besides, inspired by effective Transformer based method[22, 16] with pretraining on

Figure 2. An example of the proposed TIPS dataset.

video action recogniton, retrieval, captioning etc., we also adopt Transformer[25] for procedure segmentation.

In this work, we propose a novel MT-GBD model to predict whether the frame is a segmentation or not. The differences between our proposed MT-GBD and existing methods are two-fold: 1) Considering the *contiguity* of the procedure, MT-GBD adopts Transformer for closely-coupled context encoding and temporal convolution for frame-wise prediction; 2) MT-GBD uses *Gaussian Boundary Detection* (GBD) to enforce the model to predict more smooth results instead of hard boundaries used in [3, 14]. Experimental results show that MT-GBD outperforms the state-of-the-art methods to a large extent.

2.3. Temporal Video Segmentation Metrics

Existing temporal video segmentation models frequently employ the F1 score and MIOU as an evaluation metric [14, 3]. However, F1 reports the results based on a predefined threshold, while MIOU only measures the precision instead of Recall. The main differences between our proposed experienced IOU (EIOU) and those frequently-used metrics are two-folds: 1) EIOU is a better evaluation metric with consideration of both *precision and recall* without a pre-defined threshold; 2) EIOU calculates the evaluation score considering the *various importance* of each segment. According to our study, the novel EIOU is more like a human judge compared with F1 and MIOU.

3. The TIPS Dataset

Currently, existing temporal video segmentation datasets are not targeting procedure segmentation, or small dataset and focus on one specific domain. It is hard to evaluate the performance of deep models. However, manually labeling a large and diverse video dataset is quite challenging since it requires the annotators to watch different kinds of videos with various unfamiliar fields, foreign languages, or long duration. To overcome those issues, we collected a new TIPS dataset automatically from YouTube, composed of videos with contiguous segments from diverse categories.

3.1. Dataset Collect Methods

To generate a large-scale dataset, we built a 3-step workflow to automatically obtain labeled data from YouTube. Our key insight is that a large number of instructional videos contain explicit speech utterance of procedural steps, i.e, “step one”, “step two”. These features make it possible for the machine to automatically generate segment labels. In detail, the TIPS dataset is collected in the following 3 steps:

Collecting instructional videos. Our goal is to find high-quality instructional videos. To quickly find large-scale dataset, we first download videos from YouTube and then filter them according to a heuristic approach. We search the video title with keywords like ”How to” or ”ways to” etc. as instructional videos. In all, we processed 160M videos and traversed more than 20M instructional videos.

Select well-organized videos. To guarantee the data quality, based on the ASR speech text, we filter unorganized videos by the following rules. 1) Firstly, we match the speech text that explicitly mention the keywords like “step 1”, “step two”, and so on. 2) Secondly, we filter out invalid videos that contain less than 2 steps or more than 10 steps. Videos with 1 step are too short for segmentation and those with too long steps are complex which can be considered as future work. 3) Thirdly, we further filter out videos with missing steps through counting the number of each step, that means, all steps must be continuous. 4) Fourthly, we filter out the remaining videos with sub-steps which contains multiple steps between two continuous steps. These videos are complicated in structure and our proposed dataset focuses only on top-level procedure segmentation.

Construct segment labels. We finally collect 63k well-structured videos with high-quality procedure segmentation. Then we map each video segmentation to the corresponding solution step. Fig. 2 shows an example of ”how to set up GitHub”. Since the speech text explicitly mentioned “step number one”, “step number two”, the ground-truth segmentation is then labeled with the corresponding timestamps. For example, if the phrase “step number one” was pronounced at the timestamp 0:00:33, then the first corresponding segmentation label is annotated at the exact timestamp 0:00:33.

3.2. Dataset Statistics

In this section, we list the statistics of our proposed TIPS dataset. As shown in Tab. 1, the TIPS dataset contains a total of 63,756 samples with 358,247 procedure segments, among which 19,140 videos are taken as test dataset.

Fig. 3 presents the top 20 categories of TIPS. The categories are generated by aggregating video tags or titles.

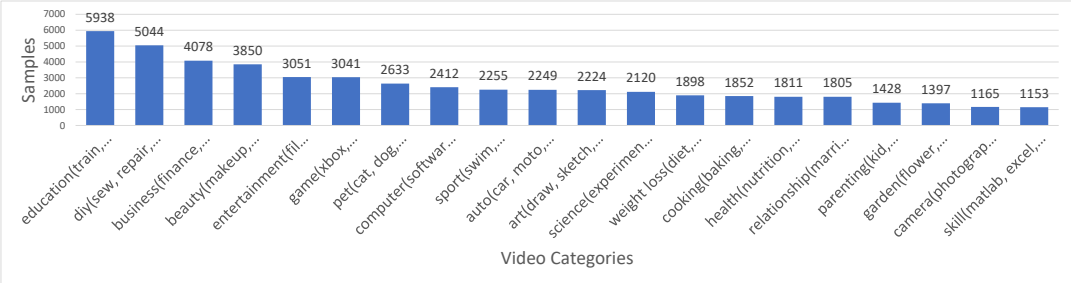

Figure 3. Top 20 Categories Distribution of this Dataset.

TIPS contains a variety of popular categories, such as education, DIY, business, and beauty. Each primary category contains 3-7 secondary categories. For example, the primary category "education" contains several secondary category such as "certificate", "exam". Besides, the TIPS dataset contains 20% videos longer than 10 minutes, which also brings challenges for deep-learning models.

3.3. Data Quality Study

The data quality is of crucial importance for semi-automatically generated data. We perform 3 types of quality check. 1) as mentioned in section 3.1, we select the well-organized video through well-designed heuristic rules; 2) we randomly select 100 cases and ask the annotators to manually check the data for accurate evaluation.

Specifically. we conduct data quality study of both precision and recall for these segmentations. The precision is 1.0 and the recall is 0.95 for the labeled 100 videos. In details, we recruited 4 annotators and asked each annotator to label 100 videos. To evaluate the precision, the labeling tool directly localized to the checkpoint of each segmentation with a 10-second window (5 seconds before and after the segmentation checkpoint), and the annotators were asked to watch this video clip to select whether there is a segmentation checkpoint or not during this time. All segmentations are valid. Then for the recall, we asked the annotators to watch the whole video and report whether there is any missing step. There are 5% videos missing the final step, such as "finally...". The labels automatically annotated are basically agreed by users except a small portion of a little time shift between the speech and the real action, usually less than 5 seconds. This motivates us to design the Gaussian function to blur the sharp segmentation. Furthermore, the annotator is asked to label whether the first sentence of each step can be exploited as captioning. There are over 30% invalid sentences like "step 1, please you , gotta" and "step 3 on behalf of ludmila.", which are hardly used as the descriptive sentence for the segmentation.

4. The MT-GBD Method

In this section, we introduce our proposed Multi-modal Transformer with Gaussian Boundary Detection (MT-GBD) with four parts: Feature Extraction, Multimodal Transformer, Temporal convolution and Gaussian Boundary Detection.

4.1. Feature Extraction

Firstly, SlowFast ResNeXt-101 proposed by [7] is used to extract video features, as shown in Eq. (1).

$$V^f = SlowFast(video), \tag{1}$$

where $video \in R^{M\times w\times h\times c}$ is a sampled video with a maximum of M frames. Each frame has a region size of $w \times h$ and c channels. $V^f \in \mathbb{R}^{M\times d^{'}}$ is the output feature. To enhance each frame with positional information, an additional positional embedding matrix is used, as denoted in Eq. (2).

$$V = Linear(V^f) + E^{(p)}[p_{idx}], \tag{2}$$

where $Linear(V^f) \in \mathbb{R}^d$ is the video semantic embedding. $E^{(p)} \in \mathbb{R}^{M\times d}$ is the video positional embedding matrix, $p_{idx} \in [1, 2, \cdots, M]$ is the positional indexes. Therefore, $E^{(p)}[p_{idx}] \in \mathbb{R}^{M\times d}$ is the video positional embedding and thus $V \in \mathbb{R}^{M\times d}$ is the final video embedding. Then, a Bert Tokenizer is used to tokenize video transcripts, as shown in Eq. (3).

$$word_{ids} = BertTokenizer(transcript), \tag{3}$$

where $word_{ids} \in \mathbb{R}^N$ is the tokenized word index, N is the max length of the input transcripts. Note that we remove instruction flags such as "step 1", "step two" in the transcript, since we expect the model to predict procedure segmentation with semantic information, instead of focusing on these special flags. Similar to the visual part, Eq. (4) is used to calculate the transcript feature with the consideration of position information.

$$L = E^{(t)}[word_{ids}] + E^{(q)}[q_{idx}], \tag{4}$$

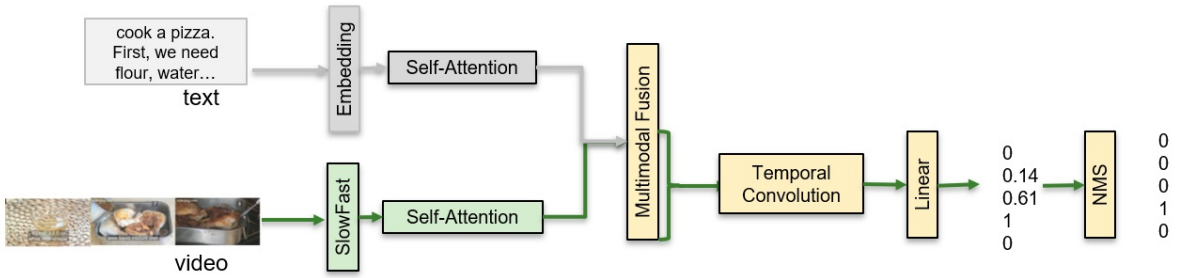

Figure 4. The MT-GBD model architecture. This model includes three major modules: feature extractor, multimodal fusion and temporal network. First there are two extractors to extract video and text features, and then self-attention model for encoding each modality information. Then the MHA(multi-head attention) module is adopted to fuse the multimodal fusion. Finally, a temporal convolution network followed by linear layer is used to predict the final segmentation labels.

Where $E^{(t)} \in \mathbb{R}^{S\times d}$ is the transcript word embedding matrix. S is the vocabulary size. $q_{idx} = [1, 2, \cdots, N]$ is transcript word positions. $E^{(q)} \in \mathbb{R}^{N\times d}$ is the transcript position embedding matrix. $L \in \mathbb{R}^{N\times d}$ is the final transcript embedding.

4.2. Multimodal Transformer

Current state-of-the-art models for multimodal fusion are transformer based models. Recently, as a basic module in Transformer proposed by [25], Self-Attention has been widely used in many NLP tasks. Typically, Self-Attention can be written in the following formula in Eq. (5):

$$\begin{aligned} Q &= Linear(X), \\ K &= Linear(X), \\ V &= Linear(X), \\ \tilde{X} &= softmax(QK^\top/\sqrt{d_k} + M)V. \end{aligned} \tag{5}$$

If input feature $X \in \mathbb{R}^{N_x\times d_x}$, then $Q, K, V \in R^{N_x\times d_k}$, $M \in \mathbb{R}^{N_x}$ is the attention mask, $\tilde{X} \in \mathbb{R}^{N_x\times d_x^{'}}$ is the attended result. For simplification, we rewrite Eq. (5) as Eq. (6):

$$\tilde{X} = Att(X, M) \tag{6}$$

Then, the key attention operations in our proposed multimodal transformer can be simply represented in Eq. (7):

$$\begin{aligned} \tilde{V} &= Att^{(a)}(V, M^V), \\ \tilde{L} &= Att^{(b)}(L, M^L), \\ \tilde{H} &= Att^{(c)}([\tilde{V}; \tilde{L}], [M^V; M^L]), \end{aligned} \tag{7}$$

where $M^V \in R^M$ is video attention mask, $M^L \in \mathbb{R}^N$ is the transcript attention mask, $a, b, c \in \mathbb{R}^+$ denotes the number of stacked attention layers. $[;]$ denotes the concatenation operator. $\tilde{V} \in \mathbb{R}^{M\times h}$ is the video attended result, $\tilde{L} \in \mathbb{R}^{N\times h}$ is the transcript attended result. $H \in \mathbb{R}^{(M+N)\times h}$ is the cross attended result.

4.3. Temporal Convolution Network

Although Transformer models multi-modal fusion, we still resort to the convolutional layers to perform the temporal segmentation to further improve the performance similar to [14]. In detail, we perform a 1 × 3 convolution layer with an ReLU activation followed by a 1 × 1 convolutional layer with residual connections for combination. The action is denoted in Eq. (8)

$$\hat{H} = ReLU(W1 * \tilde{H}[1:M] + b1) \tag{8}$$

$$H = \tilde{H}[1:M] + W2 * \hat{H} + b2 \tag{9}$$

where $\tilde{H}[1:M]$ is the input with only video attended results, $H \in \mathbb{R}^{M\times h}$ is the output of the layers, $*$ denotes the convolution operator, $W1 \in \mathbb{R}^{3\times h\times h}$ are the weights of the 1 × 3 convolution, $W2 \in \mathbb{R}^{1\times h\times h}$ are the weights of the 1 × 1 convolution, $b1, b2 \in \mathbb{R}^h$ are bias vectors.

Finally, a linear layer is used to map the channel size of H into 2, as denoted in Eq. (10).

$$\tilde{S} = Linear(H) \tag{10}$$

where $\tilde{S} \in \mathbb{R}^{M\times 2}$ is the final output logits of the multimodal transformer.

4.4. Gaussian Boundary Detection

Different from the VAS task, the VPS task has only two types of frame labels: boundary or non-boundary. This brings two challenges. Firstly, a large number of negative frames overwhelm the training. Secondly, sometimes there is a temporal shift between transcript and video, which leads to inconsistency between the strict ground-truth signals and the real scene since the boundaries of some steps are not absolute in many instructional videos. Gaussian heat map is widely applied in 2-d keypoint-based detection [13, 17] to define a keypoint to maintain semantic continuity as well as to balance positives and negatives in the training. To this end, we propose Gaussian Boundary Detection (GBD) and

add additional processing in both the training and test phase. Concretely, given a radius r and a center x, the value of x_i can be defined by following function.

$$Gaussian\_blur(x_i) = \exp \frac{-(x - x_i)^2}{2r^2} \tag{11}$$

During the training phase, we use 1d Gaussian to blur the ground-truth boundary, as denoted in Eq. (12):

$$\overline{S} = Gaussian\_blur(S) \tag{12}$$

where S is the original ground-truth, $\overline{S} \in \mathbb{R}^n$ is the blurred ground-truth.

During the test phase, instead of simply predicting each frame by a threshold, we use Non-maximum suppression(NMS) for boundary detection to filter out these noisy responses that helps model to precisely identify the correct boundaries. In particular, we use a temporal window with a length k to slide on the prediction and filter out the non-maximal values in the window.

5. The EIOU Metric

Many segmentation tasks use MIOU as the evaluation metric. However, the MIOU metric lacks of consideration of the following two factors. One one hand, different steps usually have different importance. Most of the time, the importance of a procedure is very closely related to its procedure length. For example, a video of "how to make a cocktail" has three segments, one minute of "ingredients", five minutes of "making cocktail" and one minute of "decoration". The key procedure of this video is the "making cocktail" procedure. That means if a longer procedure is wrongly predicted, the segment experience will be much worse. On the other hand, the MIOU metric focuses on whether each prediction precisely overlaps ground-truths without consideration of the recall. Another popular segmentation metric is the F1 score which depends on a predefined threshold. In this section, we first define EIOU and then compare EIOU with existing metrics.

5.1. Definition of EIOU

Let $g = \{g_1, g_2, \cdots, g_n\}$ be the ground-truth video segments, $p = \{p_1, p_2, \cdots, p_m\}$ be the predicted video segments. The MIOU metric used in [27] can be defined in Eq. (13).

$$MIOU = \sum_{i=1}^{m} \frac{1}{m} \max_j IOU(p_i, g_j), \tag{13}$$

where p_i is the ith predicted segment, g_j is the jth ground-truth segment. IOU is the Intersection over Union operation. For each predicted segment, the MIOU metric finds the best ground-truth match and average IOUs of these matches.

The MIOU metric doesn't differentiate segmentations with various lengths. Besides, MIOU focuses on precision instead of recall with only one-directional match, which only care about whether the predicted segment is correct but not how many ground truth segments are found. An example case is shown in Fig. 5, where the horizon axis is the video's time line. If the predicted segment p_1 and p_2 matches the ground truth segment g_1 at the same time, g_1 will choose the p_2, which has the maximum IOU with itself. As a result, p_1 is a Superfluous Prediction (SP) and p_2 is an Effected Prediction (EP). Since g_3 has no corresponding predictions, we call g_3 Missed Prediction (MP). To sum up, superfluous Prediction is the segment predicted but not the best match with ground truth. Missed prediction is the segment in ground truth but not the best match with predicted segment. All SP and MP are incorrect cases to be penalized.

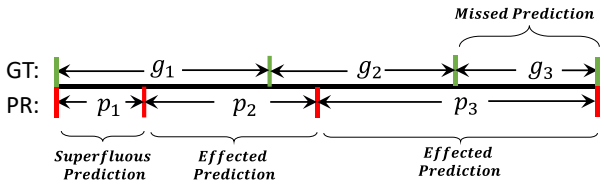

Figure 5. Illustration of SP, EP, and MP.

Motivated by this, we add length normalization and penalty for both superfluous and missed predictions. As a result, we propose Experienced IOU (EIOU), as denoted in Eq. (14).

$$EIOU = \frac{n}{n + n_s + n_m} \sum_{i=1}^{n_e} \frac{L(p_i)}{L} \max_j IOU(p_i, g_j), \tag{14}$$

where $L(p_i)$ is the length of the segment p_i and L is the overall length, n is the number of ground-truth segments, n_s is the number of SPs, n_m is the number of MPs, and n_e is the number of EPs. EIOU takes bi-direction alignment between the predicted and ground truth segments.

5.2. Comparison between metrics.

We compare different metrics through typical cases, as shown in Fig. 6.

- Comparison of Case A and Case B: Case A fails to predict a key point on a long procedure while Case B fails on a short procedure. Both MIOU and F1@0.5 cannot distinguish these two cases. However, EIOU significantly distinguishes them by segmentation length.
- Comparison of Case A and Case C: Case C fails to predict multiple ground-truth segments and is worse than

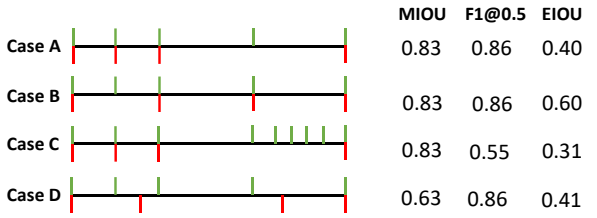

Figure 6. Comparison of different metrics. The green line denotes the ground-truth and the red line denotes predicted segments.

Case A. However, the MIOU score keeps the same, while F1@0.5 and EIOU give low scores. This shows that EIOU considers both precision and recall.

- Comparison of Case B and Case D: In case D, the predicted segments are overlapped with the ground-truth segments but not perfectly matched as in Case B. F1@0.5 gives the same score, but MIOU and EIOU can distinguish such cases.

In summary, EIOU is a trade-off metric that comprehensively considers procedure length, precision, and recall.

6. Experiments

6.1. Implementation Details

For the dataset, we randomly split the 63k data with 44k as training and 19k as validation dataset. In the feature extraction part, we extract video features with 16 fps and resize the video frame into the size of $112 \times 112 \times 3$, the maximum video frame M is 96. The maximum of transcript tokens N is 512. The output dimension of Slow-Fast model $d^{'}$ is 2560, and The the dimension of video embedding and transcript embedding d is 1024. In the multimodal transformer part, the number of stacked transcript attention layers a is 12, the number of stacked video attention layers b is 6, the number of stacked cross layers c is 2. The hidden dimension d is 768. In the Gaussian detection part, the variance of the Gaussian distribution is 1, and the maximum influence range is 3 frames on both left and right side of the ground-truth boundary according to our experiment. All the nonlinear layers of the model all use the ReLu activation function and dropout [20] to prevent overfitting. We use Adam [8] to train the model with a learning rate of 10^{-4} and a batch size of 16. We will publish the dataset and model incorporating with this paper soon.

6.2. Performance of MT-GBD Method

Comparison with SOTA We compare MT-GBD with a set of benchmarks and study the effectiveness of different modules of our model. We re-implement "MS-TCN" [3], a strong baseline of frame-wise prediction method, and Procnet[27], a proposal based method. Our proposed MT-GBD outperforms all baseline methods in a large margin.

Ablation Study We did an ablation study to gradually study the impact of each component including the Transformer, Temporal network, pretrain, speech text, and Gaussian Boundary Detection (GBD), as shown in Tab. 2. "T" is a *Transformer* model without loading pre-training weights and the convolutional structures. It improves the metric of EIOU by 4.22%, which shows the effectiveness of the self-attention structure. "T w/ C" adds additional *Temporal convolution* layers on top of the transformer (see Eq. (8)), which further improves the performance by 3.90%. This shows the complementary of the Convolution and Self-Att structure. "T w/ CP" share the same structure with "T w/ C" and loads *pre-training* weights, and the performance further improves by 1.98%. "MT" is our proposed *Multimodal-Transformer*, which considers both video evidence and speech text to infer the final segmentation. MT further improves the performance by 3.64%, which shows that the speech text give additional semantic information for procedure segmentation. Finally, MT-GBD with additional *GBD* further improves the performance by 2.91, which shows the effectiveness of using blurred Gaussian labels and sliding windows to predict final segments. Furthermore, Tab. 3 shows the performance of MT-GBD with different *video lengths* on the TIPS dataset. When the video is longer than 6 minutes, the performance drops significantly. This shows the procedure segmentation of long videos is quite a challenge.

6.3. Qualitative Analysis

We study the cases in the TIPS data as well as the results of our proposed MT-GBD method. Figure 7 demonstrates two cases representing typical result types including: a) video with explicit *scene dynamics*; b) video with distinct *actions*; c) video with *fine-grained objects and actions*. From case 1, we can see that our MT-GBD model performs well on videos with explicit scene dynamic, distinct objects and actions. The segmentation is quite similar with high overlap between corresponding ground truth segmentation. The Gaussian smoothing is capable of handling the boundary shift. However, for the case 2, the model fails to segment the details of the video and predict rather not to segment. This video reveals the rarely-appeared objects with fine-grained actions like "UTP cable", "rotate" etc. in the similar scene which is hard to for the model to predict correctly.

6.4. Effectiveness of EIOU Metric

We designed a user study to evaluate the metrics. In details, we invited 3 labelers to evaluate the EIOU metric. We borrow the idea for ranking evaluation and apply a side-by-side pair-wise comparison. Specifically, we first collected 19 segmentation cases, which are composed of different quality predictions, some cases are composed of perfect

0.12em0pt1pt **Model**	**Transf**	**Conv**	**Pretrain**	**Speech**	**GBD**	**F1@0.10**	**F1@0.25**	**F1@0.50**	**MIOU**	**EIOU**
MS-TCN[3]	×	✓	×	×	×	54.94	52.65	35.07	51.88	45.49
ProcNet[27]	×	×	×	×	×	67.95	67.87	60.51	63.88	58.72
T	✓	×	×	×	×	63.22	61.23	44.47	56.10	49.19
T w/ C	✓	✓	×	×	×	70.57	69.19	55.38	60.00	53.81
T w/ CP	✓	✓	✓	×	×	73.15	71.9	58.79	61.98	56.02
MT	✓	✓	✓	✓	×	78.43	77.48	65.6	65.62	59.61
MT-GBD	✓	✓	✓	✓	✓	**82.24**	**81.36**	**70.43**	**68.53**	**63.88**

0.12em0pt1pt

Table 2. Comparisons between MT-GBD and benchmarks on our proposed TIPS dataset. The column "Transf" means Transformer, and "Conv" means Temporal convolution backbone.

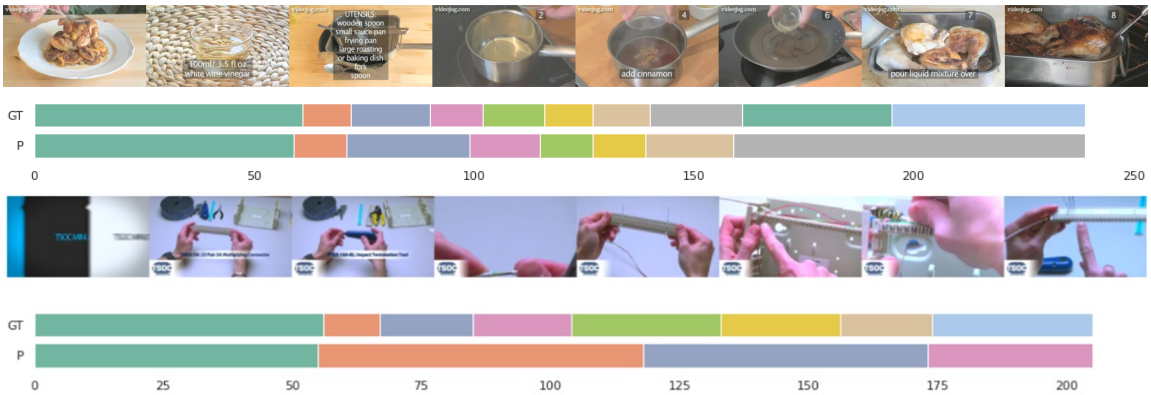

Figure 7. Qualitative results including two cases: 1. video with explicit scenes dynamics; 2. video with distinct actions; 3. video with fine-grained objects and actions. We present the frame thumbnail sequence, and the segmentations for each case. "GT" represents the ground truth segmentation, "P" represents the prediction segmentation

0.12em0pt1pt **Length**	**F1@0.10**	**F1@0.25**	**F1@0.50**	**MIOU**	**EIOU**
<6min	86.21	85.45	75.75	72.32	69.09
6-12min	74.97	73.80	61.82	64.17	57.09
>12min	64.61	62.97	45.08	54.52	45.48

0.12em0pt1pt

Table 3. Performance of MT-GBD on different durations.

predictions, some contain missed predictions, some contain superfluous predictions. Next, for each of the 19 cases, we asked the annotator to compare the predicted segmentation with all other 18 cases and label whether the segmentation is better (1) or worse (0). In the labeling tool, annotators are presented with both the ground truth segmentation (to help user understand the video) and predicted segmentation of two samples for pairwise comparison. Then, we aggregate those scores for each sample as a final satisfaction score. Finally, we order the 19 cases by a decreasing order of the segmentation satisfaction score. Ideally, the segmentation metric should perform a continuous decrease based on this satisfaction-decreasing order. From the figure 8, we can see clearly that EIOU's trend is very closely aligned with human sensed segmentation quality change. MIOU however, performs worse especially on the 15th and 17th cases, which missed long segments. Neither MIOU nor F1 perform well in the whole trend. EIOU has a clear drop aligned with user segmentation experience.

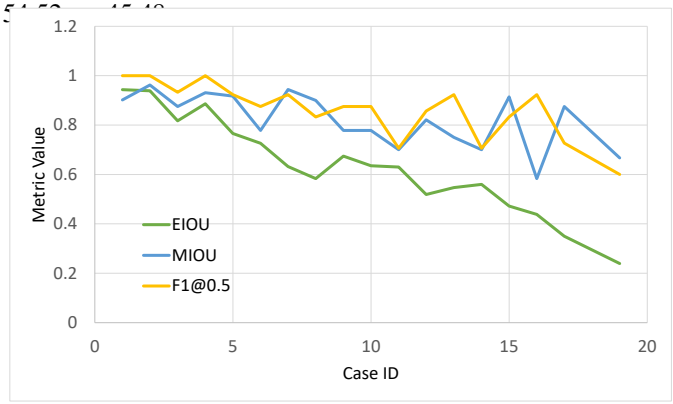

Figure 8. 19 segmentation cases decreasingly ordered by human satisfaction.

7. Conclusion

In this paper, we introduce a large and diverse TIPS dataset, an end-to-end Multimodal Transformer Gaussian Boundary Detection (MT-GBD) model, and a new Expe-

rienced IOU (EIOU) metric for video procedure segmentation task. The VPS datasets contains two types of annotations including the proposal based dataset like Youcook2 and segmentation based dataset like TIPS. As a future work, we plan to investigate a unified model for both types of datasets considering the procedural continuity and the boundary smoothness.

References

[1] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, and Will Price. Scaling egocentric vision: The epic-kitchens dataset. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 720–736, 2018.

[2] Victor Escorcia, Fabian Caba Heilbron, Juan Carlos Niebles, and Bernard Ghanem. Daps: Deep action proposals for action understanding. In *European Conference on Computer Vision*, pages 768–784. Springer, 2016.

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

[4] Alireza Fathi, Ali Farhadi, and James M. Rehg. Understanding egocentric activities. In *2011 International Conference on Computer Vision*, pages 407–414. IEEE, 2011.

[5] Alireza Fathi and James M. Rehg. Modeling actions through state changes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2579–2586, 2013.

[6] Alireza Fathi, Xiaofeng Ren, and James M. Rehg. Learning to recognize objects in egocentric activities. In *CVPR 2011*, pages 3281–3288, June 2011.

[7] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6202–6211, 2019.

[8] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[9] Irena Koprinska and Sergio Carrato. Temporal video segmentation: A survey. *Signal processing: Image communication*, 16(5):477–500, 2001.

[10] Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. Dense-captioning events in videos. In *Proceedings of the IEEE international conference on computer vision*, pages 706–715, 2017.

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

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

[13] Hei Law and Jia Deng. Cornernet: Detecting objects as paired keypoints. In *Proceedings of the European conference on computer vision (ECCV)*, pages 734–750, 2018.

[14] Colin Lea, Michael D. Flynn, Rene Vidal, Austin Reiter, and Gregory D. Hager. Temporal convolutional networks for action segmentation and detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 156–165, 2017.

[15] Peng Lei and Sinisa Todorovic. Temporal deformable residual networks for action segmentation in videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6742–6751, 2018.

[16] Huaishao Luo, Lei Ji, Botian Shi, Haoyang Huang, Nan Duan, Tianrui Li, Xilin Chen, and Ming Zhou. UniViLM: A Unified Video and Language Pre-Training Model for Multimodal Understanding and Generation. Feb. 2020.

[17] George Papandreou, Tyler Zhu, Liang-Chieh Chen, Spyros Gidaris, Jonathan Tompson, and Kevin Murphy. Personlab: Person pose estimation and instance segmentation with a bottom-up, part-based, geometric embedding model. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 269–286, 2018.

[18] Jing Shi, Jia Xu, Boqing Gong, and Chenliang Xu. Not all frames are equal: Weakly-supervised video grounding with contextual similarity and visual clustering losses. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 10444–10452, 2019.

[19] Suriya Singh, Chetan Arora, and C. V. Jawahar. First person action recognition using deep learned descriptors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2620–2628, 2016.

[20] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[21] Sebastian Stein and Stephen J. McKenna. Combining embedded accelerometers with computer vision for recognizing food preparation activities. In *Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pages 729–738, 2013.

[22] Chen Sun, Austin Myers, Carl Vondrick, Kevin Murphy, and Cordelia Schmid. Videobert: A joint model for video and language representation learning. *arXiv preprint arXiv:1904.01766*, 2019.

[23] Kevin Tang, Li Fei-Fei, and Daphne Koller. Learning latent temporal structure for complex event detection. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1250–1257. IEEE, 2012.

[24] Yansong Tang, Dajun Ding, Yongming Rao, Yu Zheng, Danyang Zhang, Lili Zhao, Jiwen Lu, and Jie Zhou. Coin: A large-scale dataset for comprehensive instructional video analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1207–1216, 2019.

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

[26] Runhao Zeng, Haoming Xu, Wenbing Huang, Peihao Chen, Mingkui Tan, and Chuang Gan. Dense regression network for video grounding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10287–10296, 2020.

[27] Luowei Zhou, Chenliang Xu, and Jason J. Corso. Towards automatic learning of procedures from web instructional videos. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.