Improving Action Segmentation on Large Egocentric Cooking Dataset

Yun Cheng[*] **Yuxuan Liu**[*] **Tiffany Ma**[*] **Erin Zhang**[*]
{yuncheng, yuxuanli, tma1, xiaoyuz1}@andrew.cmu.edu

Abstract

The task of action segmentation involves identifying not only the start and end time of different actions in an untrimmed video but also the action types. Previous approaches take in only visual inputs, whereas we attempt to solve the task using additional text input. We test our methods on the EPIC-KITCHENS dataset, whose narration annotations allow us to learn a visual-textual joint-embedding. We build upon the existing MS-TCN model which produces the start and end time of segments in a video, and we uses the visual features of the predicted segment to retrieve the closest narration in terms of their distance in the joint space. Although the video-text retrieval component does not improve baseline performance, we analyze its strength in terms of action recognition and the causes of potential failure cases.

1 Introduction and Problem Definition

Localizing and classifying activities in long untrimmed video is of great importance in video understanding ranging from video indexing to surveillance and robotics. Action segmentation is a video understanding task about identifying when and what type of action in a given video. This is done by temporally locating action segments in the video and classifying the action category of each segment. To capture long-range dependencies of action events in long untrimmed videos, recent methods using dilated temporal convolutions (Farha and Gall, 2019; Li et al., 2020) and dilated temporal graphs for temporal reasoning (Huang et al., 2020; Wang et al., 2020) have been very successful. While they can achieve decent performance on less complex datasets (Ishikawa et al., 2021), the performance significantly drops when the models are evaluated on large, challenging datasets as indicated by Huang et al. (2020). We see there is still room for performance improvement due to the inherent complexity of the task and the dataset.

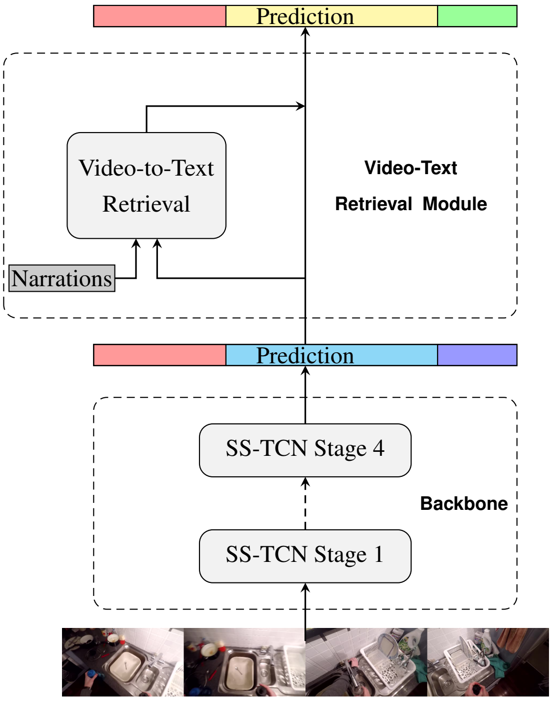


Figure 1: Overview of the proposed approach. The backbone model is MS-TCN, which is a stack of four single-stage TCNs (SS-TCN). Our video-text retrieval module extracts segments from the initial segmentation predicted by the backbone and retrieves the most relevant narration of each segment. Then the frame-wise classification label are updated with the verb in the retrieved narration.

There are two sub-challenges in action segementation, namely 1) localizing the events for action segments and 2) classifying the action in each segment. While the two sub-challenges are highly dependent on each other, we empirically found that misclassification is a more prominent issue. With the emerging field of multimodal machine learning, we are curious about whether additional information from textual modality will help mitigate the

[*] Everyone Contributed Equally – Alphabetical order

issue. Therefore, we propose a new architecture that combines temporal convolution networks with a video-to-text retrieval component. In contrast to previous works, our approach utilizes the semantic meaning of narrations to improve classification of localized action segments. We also experiment with textual inputs of different levels of complexity to see its impact on the performance. To the best of our knowledge, there is no prior work attempting at multimodal methods in solving the action segmentation task. The visual inputs are video frames extracted at a fixed frame rate, and the textual inputs are narrations describing action events in videos. We evaluate our model on the largest, egocentric cooking video dataset. A second contribution is that we experiment with textual inputs of various complexity levels and investigate their impact on the performance. This work is meaningful as it opens up a new research direction of multimodal action segmentation.

2 Related Work and Background

Early works Traditional approaches generally fall into three categories: sliding window approaches, segmental models, and recurrent networks (Huang et al., 2020). One of the earliest attempts is to detect action segments with temporal windows of different scales and non-maximum suppression (Rohrbach et al., 2012). However, this method is limited by the tradeoff between larger window size and computational costs. Others use segmental models like spatiotemporal CNNs with the semi-Markov model for tracking object relationships, action transitions, and environment change (Lea et al., 2016; Fathi and Rehg, 2013). With each action conditioned on the previous one, these methods are good at capturing local dependencies in consecutive visual patterns rather than long-range temporal relations. Another line of research focuses on temporal convolutional networks (TCNs) that perform fine-grained action segmentation using temporal convolutions (Lea et al., 2017). The method is extended to a multi-stage architecture with a set of dilated temporal convolutions in each stage. It is proven to be able to avoid temporal pooling and better capture long-range dependencies (Farha and Gall, 2019).

Text alignment Identifying the relationship between two or more modalities is one of the core challenges in multimodal settings (Baltrušaitis et al., 2019). An example of the unsupervised approaches to text alignment is to first perform temporal clustering individually on the video input and the text input, then use the two clusters to provide complementary information to one another (Alayrac et al., 2016). For instance, differences in two video segments can provide a temporal cue to a breaking point within the narrative script. Neighboring text and video input are used to assist the alignment of textual scripts and video frames as contextual information (Shi et al., 2020). It is built by pooling over each modality within K units. The mean representation of two modalities is then combined through a transformer model and concatenated to the embedding of the individual.

Our task concerns video and scripts in the cooking domain. In one of the similar experiments, the text script is parsed into action-object (i.e. verb-noun) classes, and the video frames are aligned to the text script by matching the tokens of action-objects to those present in the frames through a similarity measure in the joint embedding (Malmaud et al., 2015).

Text-Image matching To build better representation for the actions, we want to use the narrations features in the frame that are closely associated with the action that is conducted.

Given a set of image features, encoding regions in the image, and a set of word features extracted from the sentence, Stacked Cross Attention (Lee et al., 2018) determines the similarity between image-sentence pair by inferring how important a region is to the sentence, and it can also reversely infer how important a sentence is to the image. An additional position feature is concatenated for the object with the visual feature extracted by ResNet (Wang et al., 2019). The image is divided into blocks, and embedding vectors representing the positions of the blocks are combined with weights determined by overlap between the block and the visual feature. The addition is motivated by the fact that the positions of objects in the image are related to the semantics of the image. This intuition aligns with our task since we expect that the relative positions of objects are associated with the action during cooking.

3 Task Setup and Data

The main task is to segment egocentric (first-person) cooking videos from EPIC-KITCHENS dataset into action-object pairs. Given a video clip in the form of a sequence of frames, we want to identify the type of actions as well as their start and end time in the given video.

3.1 Dataset

We use the largest egocentric (first-person) dataset EPIC-KITCHENS-100, which features 100 hours, 700 variable-length videos with 90K actions of 37 participants (Damen et al., 2020). The egocentric view provides a unique perspective on people-object interactions, attention, and intention. Meanwhile, it also imposes extra challenges compared to third-person datasets like YouCook2 (Zhou et al., 2018). One of the challenges is that certain actions, such as eating and drinking, cannot be directly observed due to the limited field of view. Other challenges include unseen participants, unseen cooking actions, frame noises from different sources (i.e. background and lighting), long videos with many action instances.

3.2 Task formulation

The dataset consists of two modalities: video frames of egocentric cooking scenes and narrations describing the action in the scenes. The narrations are transcribed from the audio in the form of imperative phrases: verb-noun with optional propositional phrase. The goal is to predict a verb class for each frame to identify the action in the segments.

Formally, the visual input consists of a sequence of M RGB frames in temporal order, denoted as $F = (\mathbf{f}_i)_{i=1}^{M}$. The RGB frames are sampled from untrimmed videos at a rate of 50 frames per second. The textual input is a sequence of N audio-transcribed narrations in temporal order, denoted as $C = (\mathbf{c}_i)_{i=1}^{N}$. Our goal is to infer the action class label for each frame. The ground truth is given by $Y = (\mathbf{y}_i)_{i=1}^{M}$. Each $\mathbf{y}_i \in \{0, 1\}^K$ is a tuple of one-hot vectors encoding the true verb class, where K is the number of verb classes.

3.3 Dataset Statistics

3.3.1 Text Analysis

Narrations in EPIC-KITCHENS-100 are mainly imperative phrases in the form of verb-noun with optional propositional phrase (e.g. *put down plate*, *put container on top of counter*). Each annotation includes start/stop frame indices. Action verbs and object nouns, which are extracted from the corresponding narration. Verbs and nouns are further classified into classes based on their semantic meaning. There are a total of 97 verb classes and 300 noun classes in the training and validation set.

We define the frequency of a verb/noun class as the number of narrations that contain a verb/noun from that class. Both verb and noun classes have a heavy tailed distribution with tail classes ($\leq$ 1/15 of the maximum frequency) accounting for 13.02% and 11.67% total verbs and 5.38% and 1.85% total nouns in the training and validation set respectively (Figure 11). Such a distribution indicates the intrinsic complexity and entropy of the text data. Since there were no constraints on the recording duration, we observe a great variability across videos. Average sentence length of training and validation set is 15.1 and 14.8 with standard deviation of 6.3 and 6.0 words, respectively. Average number of actions per video is 135.8 (training) and 70.1 (validation) with standard deviation of 167.7 and 93.2. More distribution statistics can be found in Appendix A Table 5.

3.3.2 Video Frame Analysis

We extract RGB frames from the videos at a sampling rate of 50 FPS. Each frame is identified by participant id, video id, and a start/end frame number. More than half of the total 700 videos in the dataset have less than 25,000 frames.

Videos in EPIC-KITCHENS-100 have varied length with the longest video of 3708 seconds and shortest video of 10 seconds. 85.7 % of the videos are shorter than 1000 seconds and 66.0 % are less than 500 seconds (Appendix A Table 5, Figure 6). We also see that the number of narrations grows roughly linearly with video length (Figure 12).

We compile all training and validation samples of a given verb class and compute the average number of frames for this class (Appendix Figure 9). For most verb classes, the average number of frames in each class are roughly the same in both training and validation set, except a few where the validation sets have more frames. We also count the total number of frames for each verb class, summed over all training and validation samples in the class. We notice that such frequency corresponds to the trend of verb-class frequency in the annotations (Appendix A Figure 10). This indicates that within the dataset, the frequency of the verb class correlates to the amount of visual information in the dataset.

3.4 Metrics

We measure our performance based on segmental F1 score, edit score, and frame wise accuracy (Lea et al., 2017). For each predicted action segment, we calculate its IoU with respect to the corresponding ground truth. If the score is above a threshold τ, then the prediction is considered as a true positive (TP) otherwise a false positive (FP). Over-segmentation is addressed since if more than one correct segments lie within a single true action, only one is labelled as TP and all others are FP.

4 Models

4.1 Baselines

We have three baseline models: FC, MS-TCN (Li et al., 2020), and DTGRM (Wang et al., 2020).

FC We implement a vanilla 2-layer fully connected neural network that performs frame-wise classification on the input video frames. The inputs are features of dimension 1024 extracted using pretrained I3D (Carreira and Zisserman, 2017).

MS-TCN MS-TCN (Farha and Gall, 2019) is a multi-stage architecture using TCN. The first layer of a single-stage TCN (SS-TCN) adjusts inputs dimension, followed by several dilated 1D temporal convolution layers with dilation factor doubled at each layer. All layers have ReLU activation with the residual connection. MS-TCN stacks multiple SS-TCNs so that each takes initial prediction probabilities from the previous stage and refines it. The overall architecture is trained with the cross entropy classification loss and a truncated mean squared error over the frame-wise log probabilities that penalizes over-segmentation.

DTGRM Wang et al. (2020) proposed DTGRM which refines a predicted result given by the backbone model (e.g. I3D) iteratively. The model stacks K dilated graph convolution layers to perform temporal reasoning across long timescales, where each layer updates the hidden representation of every input frame. To reduce over-segmentation error, an additional self-supervised task is introduced to simulate over-segmentation error by randomly exchanging part of input frames. Both the original and exchanged frame sequences are fed into the model as input, with the output being action class likelihood for two frame sequences as well as exchange likelihood for each frame.

4.2 Proposed Approach

4.2.1 Backbone Model

We use the original implementation of MSTCN in Farha and Gall (2019) as the backbone model. The backbone takes in features extracted by I3D, same as in Farha and Gall (2019). Given the feature vectors $(\mathbf{x}_1, \ldots, \mathbf{x}_M)$ of a video, the model outputs an initial segmentation $(\hat{\mathbf{y}}_1, \ldots, \hat{\mathbf{y}}_M)$ where M is the number of frames and $\hat{\mathbf{y}}_i$ is the action class label of the predicted verb of frame i. From the prediction, we can generate N' segments and their corresponding start-end frame number $\{(s_i, e_i)\}_{i\in[1\ldots N']}$, by treating consecutive frames that are predicted with the same class as in the same segment.

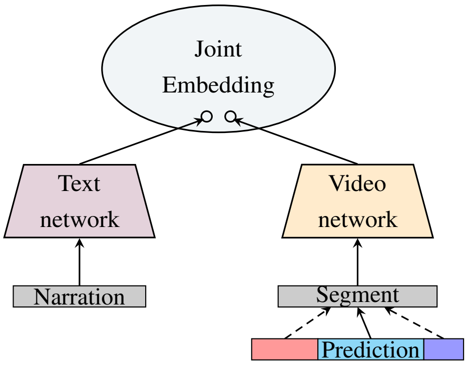


Figure 2: Illustration of the video-text retrieval module that computes the similarity score for a given pair of narration and video segment.

4.2.2 Video-Text Matching

Since misclassification a more prominent issue in the baseline experiments, our proposed solution utilizes an enriched, pretrained video-text embedding to improve the labeling. We first extracted frame-level and video-level features similarly as in Miech et al. (2019). 2D features are extracted with the ImageNet pre-trained Resnet-152 (He et al., 2016) at the rate of about 1 FPS, and 3D features are extracted with the Kinetics (Carreira and Zisserman, 2017) pre-trained ResNeXt-101 16-frames model (Hara et al., 2018) to obtain about 0.78 feature per second. We freeze the ResNet and ResNeXt-101 components for feature extraction and only finetune on the final projection functions f and g, which are composed of linear layers and gated linear units and explained in details below.

Denote the 2D features as $(\mathbf{x}_1^{2D}, \ldots, \mathbf{x}_{M_{2D}}^{2D})$ and the 3D features as $(\mathbf{x}_1^{3D}, \ldots, \mathbf{x}_{M_{3D}}^{3D})$ where $\mathbf{x}_i^{2D}, \mathbf{x}_i^{3D} \in \mathbb{R}^{2048}$. Given pairs of start-end frame number $(s_i, e_i)_{i\in[1\ldots N]}$, marking the start and end of a segment, we first find the set of 2D and 3D feature indices corresponding to segment i as $(s_i^{2D})_{i=1}^N$ and $(s_i^{3D})_{i=1}^N$ respectively, where N is the number of segments. In other words, all 2D features with indices in s_i^{2D} and 3D features with indices in s_i^{3D} describe the segment from the s_i-th frame to the e_i-th frame. Then we aggregate the features of one segment using temporal maxpooling and concatenate 2D and 3D features to form a single 4096-dimensional feature vector

$$\begin{aligned}\mathbf{v}^{2D} &= maxpool(\{\mathbf{x}_j^{2D}\}_{j\in s_i^{2D}})\\ \mathbf{v}^{3D} &= maxpool(\{\mathbf{x}_j^{3D}\}_{j\in s_i^{3D}})\\ \mathbf{v}_i &= concat(\mathbf{v}^{2D}, \mathbf{v}^{3D})\end{aligned}$$

Similar to Miech et al. (2019), we also use the GoogleNews pre-trained word2vec embedding model to obtain a word embedding $\mathbf{c}_i$ of the text input. We then transform $\mathbf{v}_i, \mathbf{c}_i$ using the learned projection function finetuned on EPIC-KITCHENS $f : \mathbb{R}^{2048} \to \mathbb{R}^d, g : \mathbb{R}^{2048} \to \mathbb{R}^d$ where d is the dimension of the common video-text embedding space. Finally, we perform video-text matching between a segment $\mathbf{v}_i$ and every verb $\mathbf{c}_i$ by computing the cosine similarity score as

$$s(\mathbf{v}_i, \mathbf{c}_j) = \frac{\langle f(\mathbf{v}_i), g(\mathbf{c}_j)\rangle}{\|f(\mathbf{v}_i)\|_2\|g(\mathbf{c}_j)\|_2}$$

which is high when the action $\mathbf{c}_j$ is likely to take place in the segment represented by $\mathbf{v}_i$.

In order to determine the action class of the i-th segment with visual feature $\mathbf{v}_i$, we calculate $s(\mathbf{v}_i, \mathbf{c}_j)$ for a set of 18003 $\mathbf{c}_j$'s, which is the total number of possible narrations in the EPIC-KITCHENS dataset, and the word embeddings are pre-computed. We then used the action class of the j^*-th narration, where $j^* = \max_{j\in[18003]} s(\mathbf{v}_i, \mathbf{c}_j)$, to be the class prediction of the i-th segment. Figure 2 shows how we utilizes the joint embedding to determine the action class of each segment.

4.2.3 Loss Function

Backbone We use a combination of cross-entropy classification loss

$$\mathcal{L}_c = \frac{1}{M}\sum_{t=1}^{M} -\log(\hat{\mathbf{y}}_{t,c}^*)$$

and truncated mean squared smoothing loss that aims to reduce over-segmentation errors as in (Farha and Gall, 2019)

$$\begin{aligned}\Delta_{t,c} &= |\log\hat{\mathbf{y}}_{t,c}^* - \log\hat{\mathbf{y}}_{t-1,c}^*|\\ \tilde{\Delta}_{t,c} &= \begin{cases}\Delta_{t,c} & \text{if } \Delta_{t,c} \le \tau\\ \tau & \text{otherwise}\end{cases}\\ \mathcal{L}_s &= \frac{1}{MK}\sum_{t,c}\tilde{\Delta}_{t,c}^2\end{aligned}$$

where M is the number of frames, K is the number of action classes, $\hat{\mathbf{y}}_{t,c}^*$ is the output probability of action class c of frame t. We use $\tau = 4, \lambda = 0.15$ as in the original experiment. The final loss function is given as the sum of loss at each stage of temporal convolution

$$\begin{aligned}\mathcal{L}_{stage} &= \mathcal{L}_c + \lambda\mathcal{L}_s\\ \mathcal{L} &= \sum_{stage}\mathcal{L}_{stage}\end{aligned}$$

Video-Text Retrieval The joint embedding is trained separately using the max-margin ranking loss as in (Miech et al., 2019). The loss is given by

$$\sum_{i\in\mathcal{B}}\sum_{j\in N(i)} \max(0, \delta + s_{i,j} - s_{i,i}) + \max(0, \delta + s_{j,i} - s_{i,i})$$

where $\mathcal{B}$ is a mini-batch sample of segments-verb training pairs, s is the similarity score matrix of all training pairs, $N(i)$ denotes the set of negative pairs for pair i and δ is the margin. We fix $\delta = 0.1$ as in the original experiment.

4.2.4 Novelty and Challenges

Our approach is the first attempt to solve the action segmentation task of a dataset as large and complex as EPIC-KITCHENS in the multimodal setting. We aim to learn a visual-textual joint embedding where the embedding of a video segment is close to the embedding of the narration describing the segment.

Without extensive training and fine-tuning, the video-text retrieval module gives comparable result on recall metrics, R@$\{1, 5, 10\}$, as in the original pretrained HowTo100M model (Miech et al., 2019). However, using the joint embedding space for action segmentation is challenging because retrieving the correct text requires good initial segmentation from MS-TCN. If the output from MSTCN differs greatly from the ground truth segmentation, the result may be much less desirable. A potential solution to this issue may be to train MSTCN with better visual features to obtain a more stabilized and credible segmentation, which are discussed in Section B. Another challenge is that the provided narrations, which we treat as captions, are not full sentences and detailed descriptions of the video, since HowTo100M (Miech et al., 2019) worked well with several short captions like ours concatenated together, we experiment with concatenating neighboring narrations to provide more context.

5 Results

5.1 HowTo100M Experiments

We conducted experiments to evaluate the quality of the video-text joint-embedding learnt after finetuning the HowTo100M model. For a video which is made up from N segments, we use the ground truth start and end frame number, (s_i, e_i), to segment out the i-th actions. Following the feature extraction procedure described in Section 4.2.2, we extract visual feature $\mathbf{v}_i$ for the i-th segment. In order to test what kinds of text input is helpful in building the joint-embedding, we consider two kinds of input in describing the action in a given segment and use the word2vec model mentioned in 4.2.2 to extract the text embeddings: embedding $\mathbf{c}_i^{verb}$ denoting single action verb, and $\mathbf{c}_i^{narr}$ denoting entire narration including verb and noun.

In Table 3 and Table 1, the *Label* column shows different $\mathbf{c}'_i$ used to calculate the similarity score $s(\mathbf{v}'_i, \mathbf{c}'_i)$ for video-text retrieval. For *Narration*, $\mathbf{c}'_i$ is $\mathbf{c}_i^{narr}$; for *Verb*, $\mathbf{c}'_i = \mathbf{c}_i^{verb}$; for *Verb+Context*, $\mathbf{c}'_i = concat(\mathbf{c}_{i-1}^{verb}, \mathbf{c}_i^{verb})$; for *Narration+Context*, $\mathbf{c}'_i = concat(\mathbf{c}_{i-1}^{narr}, \mathbf{c}_i^{narr})$. We also run experiments on a subset of all segments by filtering out segments that are less than *Segment Threshold*$\times 64$ frames long (in the original 50fps sampling rate). Moreover, in Table 4, we vary $\mathbf{v}_i^{'}$ to see the impact of including visual features from neighboring segments. When $l_v = l$, $\mathbf{v}_i^{'} = concat(\{\mathbf{v}_j\}_{j \in [i-l...i]})$.

5.2 MS-TCN Experiments

We selected the best performing HowTo100M joint embedding model and used it after the prediction of MS-TCN to generate the final prediction, as explained in detail in Section 4.2.2. We evaluate the trained joint embedding on action segmentation task by using it as a post-processing mechanism on MS-TCN's predicted output. Selecting the two better performing embedding setting, we evaluate our embedding on *Narration* and *Narration + Context* setting. After obtaining the segments' start-end predictions $(s_i, e_i)_{i \in [N']}$, instead of retrieving the closest narration, we retrieve from word embeddings in the form of $conat(\mathbf{c}_{i-1}^{pred}, \mathbf{c}_j)$, where $\mathbf{c}_{i-1}^{pred}$ is the narration that is the closest to the $(i-1)$-th segment in the joint-embedding. Moreover, we skip over segments that are predicted as background by MS-TCN. We show results for both retrieval with the original text input $\mathbf{c}_j$ and the text input with context $conat(\mathbf{c}_{i-1}^{pred})$ in Table 2.

6 Analysis

6.1 Analysis of Baselines

FC Since the classification is performed frame-wise and considers no temporal relations, the result is highly fragmental. We further note that the model tends to overfit at an early stage. The poor generalizability is indicated by the relatively low Edit score and Figure 3.

MSTCN MSTCN demonstrates its effectiveness in segmenting out the most frequent label classes. We observe that the model assigns one of the most frequent verb classes when it struggles to label the action classes. The result implies that the model tends to memorize the label frequency.

To account for the complexity introduced by the large number of actions classes of EPIC-KITCHENS, we have tried 12-layer and 15-layer single-stage TCNs. We have also tried to increase the output channel size of the 1D convolutions from 64 to 128 or 256. However, results from the experiments achieve similar performance as the ones for MS-TCN presented in Table 2; We believe that increasing number of layers and channels cannot improve the representation power of MS-TCN.

DTGRM Similar to MSTCN, DTGRM is able to output reasonable segmentation results. 3 shows that one of its improvements from MSTCN is its capability in clearly segmenting out smaller segments, which proves the effectiveness of the additional fine-tuning component even on a more complex dataset like EPIC-KITCHENS. However, we also notice that DTGRM tend to over-segment on videos with fewer segments.

6.2 Analysis of Text-Video Retrieval

With the aim of building more robust joint embedding space, we experimented Howto100m retrieval model on EPIC-KITCHENS dataset under different settings, as explained in Section 5.1. From

Label	MR	R1	R5	R10
Verb		Fail to learn		
Verb+Context	132	0.01	0.03	0.05
Narration	40	0.04	0.14	0.22
Narration+Context	16	0.19	0.49	0.66

Table 1: Results of Video-Text Retrieval using different types of label

Methods	Acc	Edit	F1@$\{10, 25, 50\}$		
FC	34.90	18.58	17.47	13.66	8.04
MS-TCN (Farha and Gall, 2019)	38.65				
DTGRM (Wang et al., 2020)	37.71				
Proposed Method (*narration*)	20.93	24.20			2.49
Proposed Method (*narration+context*)	22.81	24.40			2.69

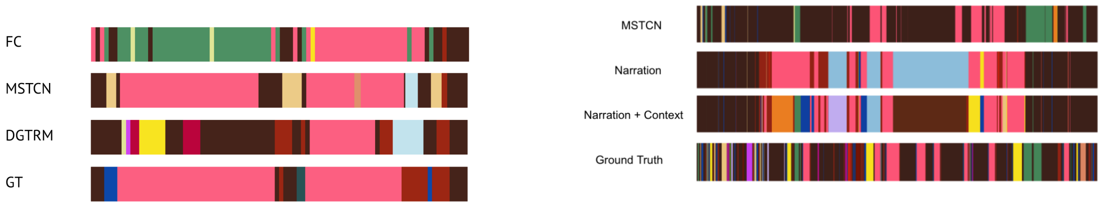
Table 2: Results of baseline models

Label	Segment Threshold (l_s)	MR	R1	R5	R10
Narration	0	46	0.03	0.11	0.18
	1	40	0.04	0.14	0.22
	3	13	0.10	0.30	0.45
Verb+Context	1	132	0.01	0.03	0.05
	3	104	0.01	0.04	0.07
	5	26	0.03	0.13	0.24
Narration+Context	1	6	0.19	0.49	0.66
	3	5	0.23	0.54	0.70
	5	2	0.36	0.81	0.92

Table 3: Results of Video-Text Retrieval using different segment threshold

Visual Context Threshold (l_v)	Segment Threshold (l_s)	MR	R1	R5	R10
2	0	135	0.01	0.03	0.07
3	0	50	0.02	0.08	0.16
4	0	26	0.03	0.16	0.27
3	1	36	0.03	0.14	0.23
3	3	102	0.02	0.09	0.14

Table 4: Results of Video-Text Retrieval using different segment threshold on visual features



(a) Variation of Baseline Models

(b) MS-TCN with Joint Embedding

Figure 3: Qualitative results of Methods

Table 1, we see a strong positive correlation between richness of text and performance of retrieval. With plain verbs as label, the retrieval result is orders of magnitudes worse than other settings. By adding more textual information to just verb, we see improvements in median-rank of true label and recall scores in both *Narration* (current narration) and *Narration+Context* (previous and current narrations). The reason that using noun helps is more obvious: for verbs that cannot be easily visualized or doesn't correspond to a single action (e.g. *take*), the less ambiguous object can narrow down the search space for appropriate verbs.

In addition to text input, length of segments also affects performance (see Table 3). By only considering segments that contain more than *Segment Threshold* number of features, we found that the longer the segments, the better the retrieval performance. Because the model uses maxpooling across video features from the same segment, the final video feature for the segment contains the most eminent features within the segment. However, since ground truth and video feature are not perfectly aligned (videos are downsampled before feature extraction), there are inevitable noise at the boundary of segments. Thus the shorter the segment, the noisier the video feature.

6.3 Analysis of post-MSTCN Text-Video Retrieval

From Tabel 1, we see that using video-text matching to predict the action class of a given segment outputted by MS-TCN gives worse performance. By comparing the results in Table 3 and Table 4, we see that although enhancing the text input with context (*Narration+Context*) boosts the matching performance considerably, maxpooling across frames of different actions ($l_v > 1$) worsen the performance, comparing to *Narration+Context*. Moreover, we have observed earlier matching on longer segments, $l_s \geq 3$, performs better, but this behavior is not consistent when $l_v > 1$, since from Table 4, when $l_v = 3, l_s = 3$, the median rank is much higher than $l_v = 3, l_s = 1$. The learnt joint-embedding is very brittle when it comes to long video segments containing multiple actions. The high performance of *Narration+Context* suggests that although text input assists in identifying the action, the quality of the visual feature is equally important in ensuring the correct retrieval.

Another issue comes from inferring the action class after video-text matching is done. We only know that the ground-truth paired narration is among the top-matching narrations, and majority-voting among top-matching narrations does not produce consistently the correct narration, which hurts performance.

6.4 Qualitative Analysis and Examples

From 3a we can see that it is very challenging for FC to get the class label correctly, as it predicts the most common verb “wash” (olive-green) instead of the ground truth “mix” (hot pink). For videos with a few number of long segments, DGTRM tends to over-segment. In 3b we see that the joint embedding trained with *Narration+Context* gives a slightly better classification result than one trained only with *Narration*, suggesting richness in textual modality is key.

7 Conclusion

We presented a multi-modal approach to the action segmentation task. Given the difficulty of the action segmentation task and the EPIC-KITCHENS dataset, we want to produce better prediction by post-processing the baseline model’s prediction with a trained visual-textual joint embedding. The baseline model follows a multi-stage temporal convolution architecture, and the joint embedding is trained with max-margin ranking loss. Although our experiments show that the proposed methodology does not improve the prediction, we identified potential caveats through analysis of textual and visual information. We also experimented with ways to extract better visual features using SlowFast network, which can be find in Appendix B, but even though features from SlowFast network work well on action recognition, feeding them into MS-TCN does not improve baseline performance. Similarly, the video-text retrieval model works well with well-trimmed segments but not so when fed with noisy visual features, so in the future, we would like to investigate more about how visual and text input interact across time, since the additional temporal dimension results in most of the complexity in our experiments.

References

J. Alayrac, P. Bojanowski, N. Agrawal, J. Sivic, I. Laptev, and S. Lacoste-Julien. 2016. Unsupervised learning from narrated instruction videos. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4575–4583.

T. Baltrušaitis, C. Ahuja, and L. Morency. 2019. Multimodal machine learning: A survey and taxonomy. volume 41, pages 423–443.

J. Carreira and A. Zisserman. 2017. Quo vadis, action recognition? a new model and the kinetics dataset. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4724–4733.

Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Antonino Furnari, Jian Ma, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. 2020. Rescaling egocentric vision. *CoRR*, abs/2006.13256.

Y. A. Farha and J. Gall. 2019. Ms-tcn: Multi-stage temporal convolutional network for action segmentation. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3570–3579.

A. Fathi and J. M. Rehg. 2013. Modeling actions through state changes. In *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2579–2586.

Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. 2019. Slowfast networks for video recognition.

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

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778.

Y. Huang, Y. Sugano, and Y. Sato. 2020. Improving action segmentation via graph-based temporal reasoning. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14021–14031.

Yuchi Ishikawa, Seito Kasai, Yoshimitsu Aoki, and Hirokatsu Kataoka. 2021. Alleviating over-segmentation errors by detecting action boundaries. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 2322–2331.

C. Lea, M. D. Flynn, R. Vidal, A. Reiter, and G. D. Hager. 2017. Temporal convolutional networks for action segmentation and detection. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1003–1012.

Colin Lea, Austin Reiter, Rene Vidal, and Gregory D. Hager. 2016. Segmental spatiotemporal cnns for fine-grained action segmentation.

Kuang-Huei Lee, Xi Chen, Gang Hua, Houdong Hu, and Xiaodong He. 2018. Stacked cross attention for image-text matching.

S. J. Li, Y. AbuFarha, Y. Liu, M. M. Cheng, and J. Gall. 2020. Ms-tcn++: Multi-stage temporal convolutional network for action segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1.

Jonathan Malmaud, Jonathan Huang, Vivek Rathod, Nicholas Johnston, Andrew Rabinovich, and Kevin Murphy. 2015. What's cookin'? interpreting cooking videos using text, speech and vision. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 143–152, Denver, Colorado. Association for Computational Linguistics.

Antoine Miech, Jean-Baptiste Alayrac, Ivan Laptev, Josef Sivic, and Andrew Zisserman. 2021. Thinking fast and slow: Efficient text-to-visual retrieval with transformers.

Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic. 2019. HowTo100M: Learning a Text-Video Embedding by Watching Hundred Million Narrated Video Clips. In *ICCV*.

M. Rohrbach, S. Amin, M. Andriluka, and B. Schiele. 2012. A database for fine grained activity detection of cooking activities. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1194–1201.

Botian Shi, Lei Ji, Zhendong Niu, Nan Duan, Ming Zhou, and Xilin Chen. 2020. Learning semantic concepts and temporal alignment for narrated video procedural captioning. In *Proceedings of the 28th ACM International Conference on Multimedia*, MM '20, page 4355–4363, New York, NY, USA. Association for Computing Machinery.

Dong Wang, Di Hu, Xingjian Li, and Dejing Dou. 2020. Temporal relational modeling with self-supervision for action segmentation.

Yaxiong Wang, Hao Yang, Xueming Qian, Lin Ma, Jing Lu, Biao Li, and Xin Fan. 2019. Position focused attention network for image-text matching.

Luowei Zhou, Chenliang Xu, and Jason J Corso. 2018. Towards automatic learning of procedures from web instructional videos. In *AAAI Conference on Artificial Intelligence*, pages 7590–7598.

Appendix A Data Analysis

In this section, we present the full details of our data analysis.

	Training				Validation			
	Max.	Min.	Avg.	Std.	Max.	Min.	Avg.	Std.
Verb class frequency	14848	73	1314	2829	1937	71	191	398
Noun class frequency	3617	178	724	655	430	25	108	92
Sentence length	77	3	15.1	6.3	71	3	14.8	6.0
Actions per video	940	1	136	168	564	3	70	93
Frames per verb class	2129212	20165	225170	408408	407425	2702	42016	76950
Video length	3708	10	543	645	1969	11	344	377

Table 5: Statistics of EPIC-KITCHENS-100 training and validation set

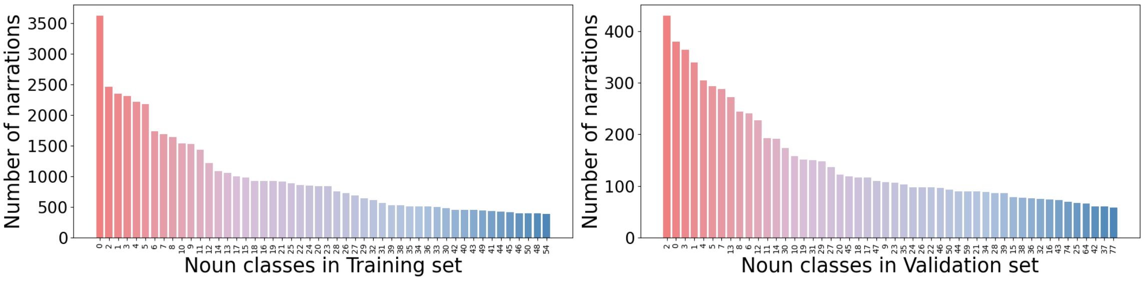


Figure 4: Frequency distribution of 50 most frequent noun classes in training and validation set

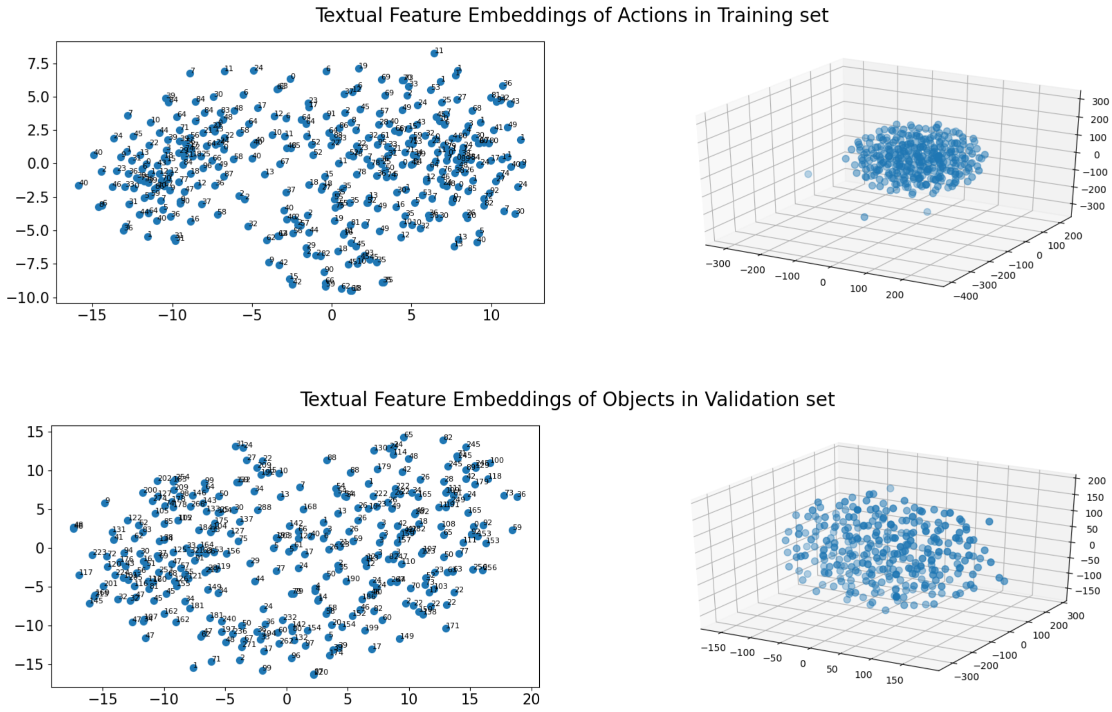


Figure 5: Example of visualizing feature embeddings of verb and noun classes in 2D and 3D space

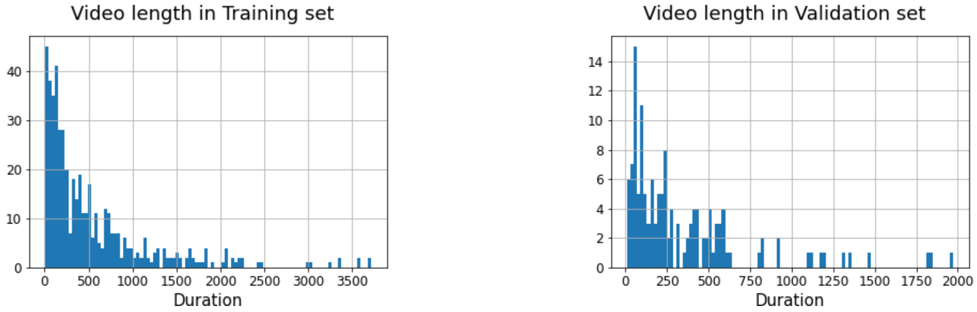


Figure 6: Distribution of video length (in seconds)

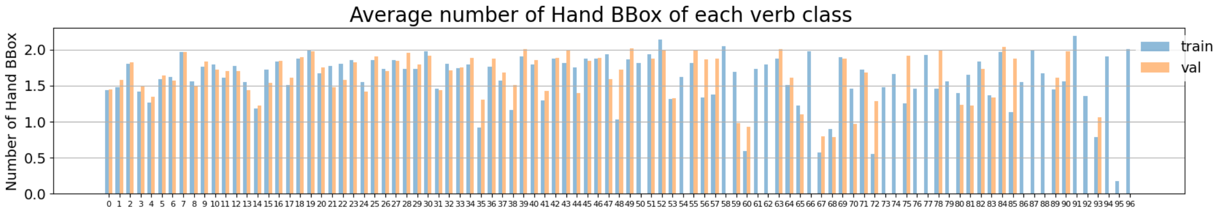


Figure 7: Average number of hand bounding-boxes in each frame of given verb class

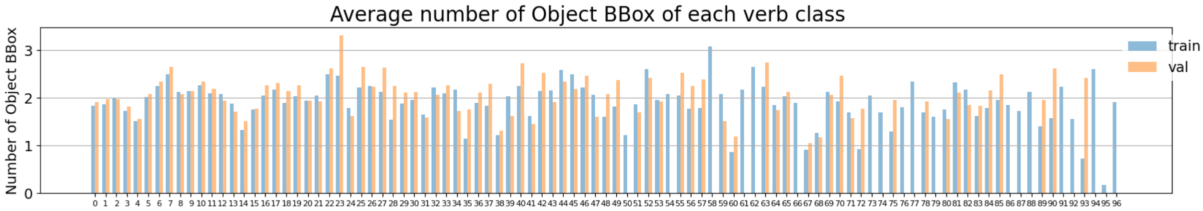


Figure 8: Average number of object bounding-boxes in each frame of given verb class

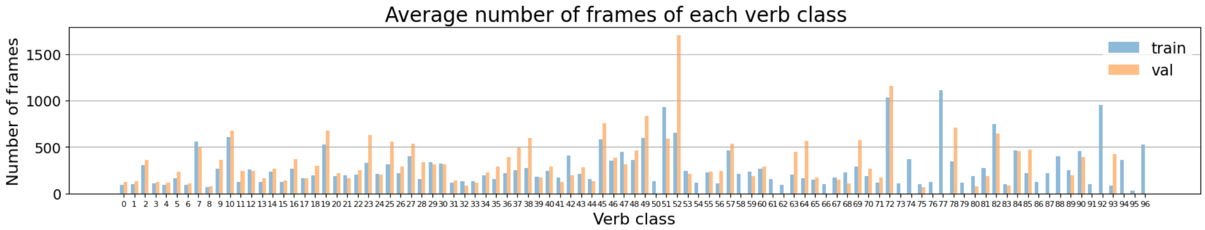


Figure 9: Average number of frames in a narration of a given verb class in training and validation set

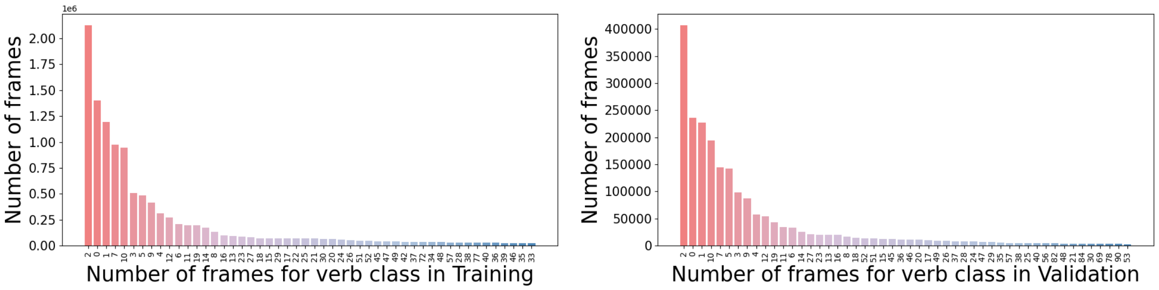


Figure 10: Distribution of number of frames in each video

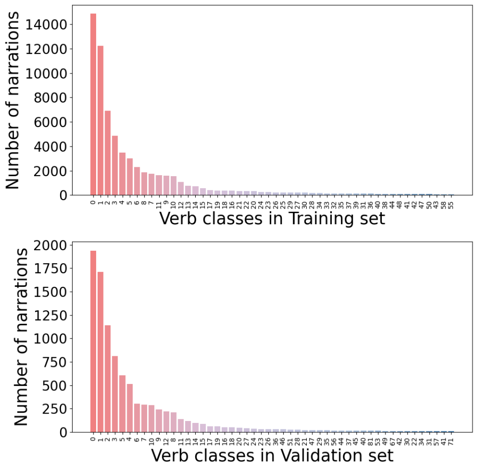


Figure 11: Frequency distribution of 50 most frequent verb class in training and validation set

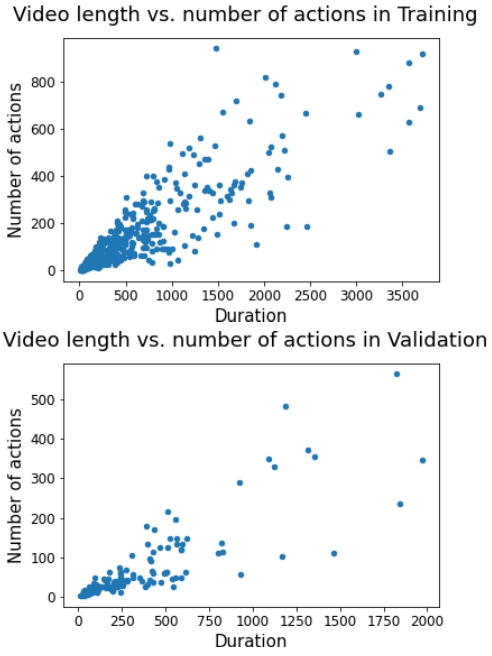


Figure 12: Number of action in each video against video length (in seconds)

A.0.1 Improve Video-Text Matching with Cross-Modal Attention

The above describes a dual encoder model that independently maps text and video to a joint embedding. It has the advantage in scalability as it can results in efficient evaluation during test time. However, as Miech et al. (2021) points out, it has limited accuracy since the simple dot product is unlikely to capture the complex vision-text interactions. Analogous to how human perform video-text retrieval, one solution is to roughly select a few promising candidates then do fine-grained search for the best candidate by paying more *attention* to visual details. Therefore, we adapt the *Fast* and *Slow* models of Miech et al. (2021) in which the *fast* dual encoder quickly eliminates candidates with low relevance while the *slow* cross-attention model improves retrieval performance with grounding. Given an input segment $\mathbf{v}_i$, we perform retrieval by searching for an action class $\mathbf{c}_j$ such that segment $\mathbf{v}_i$ is most likely to match action class based on the joitnn embedding $\mathbf{c}_j$. Specifically, given segment and action class pair $(\mathbf{v}_i, \mathbf{c}_j)$, we compute their similarity by

$$h(\mathbf{v}_i, \mathbf{c}_j) = \log(p(\mathbf{c}_j|\phi(\mathbf{v}_i); \theta))$$

where $\phi(\mathbf{v}_i)$ is extracted feature of segment $\mathbf{v}_i$ and θ is the parameters of the transformer model. To combine results from dual encoder model and cross-attention model, given input segment $\mathbf{v}_i$ and action class set $\mathcal{C}$ containing K action classes. we first obtain a subset of m action classes $\mathcal{C}_m$ (where $m \ll K$) that have the highest score according to the fast dual encoder model. We then retrieve the final top ranked action class by re-ranking the candidates using the cross attention model:

$$\mathbf{y}_i^* = \text{argmax}_{\mathbf{c}_j \in \mathcal{C}_m} h(\mathbf{v}_i, \mathbf{c}_j) + \beta s(\mathbf{v}_i, \mathbf{c}_j)$$

where β is a positive hyper-parameter that weights the output scores of the two models. We output $(\hat{\mathbf{y}}_{i,c}^*)$ as the classification probability of frame i as action c based on the similarity score and $(\mathbf{y}_i^*)_{i \in s_i^{3D}}$ as new labels for segment $i, i \in [t]$.

Appendix B SlowFast Visual Feature

In addition to using text to improve action segmentation performance, we also experiment with ways to extract visual features that could give better performance than the original I3D features. Since text-retrieval componenet did not improve the MSTCN prediction, we experimented without the component. In order to extract better visual features, we decide to use the SlowFast network (Feichtenhofer et al., 2019) for feature extraction, because it contains two pathways: the Slow and the Fast pathway. Our intuition was that information on changes in the scene, captured by the Fast pathway operating on a set of densely sampled frames, and contents in the scene, encoded by the Slow pathway outputting activations with a large number of channels, are both important to recognizing the action and differentiating between neighboring actions.

We plan to use region of interest (RoI) proposals of the frames, which are fed into RoiAlign after

	verb-top-1-acc	verb-top-5-acc	noun-top-1-acc	noun-top-5-acc
SlowFast (original)	52.98	84.05	38.27	63.99
SlowAlign	52.26	83.84	38.42	63.59
FastAlign	25.94	70.59	9.87	26.46
SlowAlign + Slow	51.76	83.31	37.35	62.30
FastAlign + Fast	52.03	83.53	37.85	62.70

Table 6: Results of applying RoiAlign at different places of the SlowFast network.

the SlowFast ResNet backbone. Our motivation is that by excluding the distracting information in the context of the video, focusing on the manipulated objects that are near the hand regions will help with identifying the action, since the text information lacks context; moreover, we assume that changes in how the objects are handled indicate the action performed.

In order to determine the quality of the features before passing into MS-TCN, we use performance on action-recognition, the original task described in Feichtenhofer, Fan, Malik, and He (2019) but performed on EPIC-KITCHENS segments, as an indicator. Table 6 presents the noun and verb accuracy of different modifications made to the original SlowFast network. We first tried to pass the activations from the Fast pathway before the prediction head into RoiAlign, since Fast pathway has much higher sampling rate; the results is shown in the *FastAlign* row. Similarly, *SlowAlign* corresponds to passing activations from the Slow pathway into RoiAlign. *SlowAlign* + *Slow* shows results of preserving the activations from both pathways but adding an additional branch of output after performing RoiAlign on activations of the Slow pathway, similar idea for *FastAlign* + *Fast*.

Poor performance of *SlowAlign* shows that the slow pathway, which contains rich spatial information due to its large channel size, needs full image information, and applying RoIAlign limits its representation significantly. Moreover, similar performances among the other model variations indices that RoiAlign does not provide better representation, and one reason could be that although context in images are not the actively manipulated objects, to determine an action like "open", changes in the surrounding between frames carry useful information, such as changes in the position of an object relative to the background.