Actor and Action Video Segmentation from a Sentence

Kirill Gavrilyuk, Amir Ghodrati, Zhenyang Li, Cees G. M. Snoek
QUVA Lab, University of Amsterdam
{kgavrilyuk, a.ghodrati, zhenyangli, cgmsnoek}@uva.nl

Abstract

This paper strives for pixel-level segmentation of actors and their actions in video content. Different from existing works, which all learn to segment from a fixed vocabulary of actor and action pairs, we infer the segmentation from a natural language input sentence. This allows to distinguish between fine-grained actors in the same super-category, identify actor and action instances, and segment pairs that are outside of the actor and action vocabulary. We propose a fully-convolutional model for pixel-level actor and action segmentation using an encoder-decoder architecture optimized for video. To show the potential of actor and action video segmentation from a sentence, we extend two popular actor and action datasets with more than 7,500 natural language descriptions. Experiments demonstrate the quality of the sentence-guided segmentations, the generalization ability of our model, and its advantage for traditional actor and action segmentation compared to the state-of-the-art.

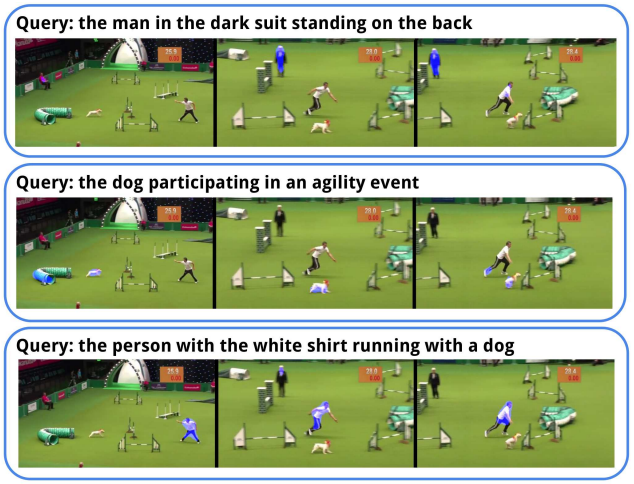

Figure 1: From a natural language input sentence our proposed model generates a pixel-level segmentation of an actor and its action in video content.

1. Introduction

The goal of this paper is pixel-level segmentation of an actor and its action in video, be it a person that climbs, a car that jumps or a bird that flies. Xu *et al*. [29] defined this challenging computer vision problem in an effort to lift video understanding beyond the more traditional work on spatio-temporal localization of human actions inside a tube, *e.g*. [19, 26, 32]. Many have shown since that joint actor and action inference is beneficial over their independent segmentation, *e.g*. [10, 28]. Where all existing works learn to segment from a fixed set of predefined actor and action pairs, we propose to segment actors and their actions in video from a natural language sentence input, as illustrated in Figure 1.

We are inspired by recent progress in vision and language solutions for challenges like object retrieval [6, 7, 17], person search [14, 30, 34], and object tracking [15]. To arrive at object segmentation from a sentence, Hu *et al*. [6] rely on an LSTM network to encode an input sentence into a vector representation, before a fully convolutional network extracts a spatial feature map from an image and outputs an upsampled response map for the target object. Li *et al*. [15] propose object tracking from a sentence. Without specifying a bounding box, they identify a target object from the sentence and track it throughout a video. The target localization of their network is similar to Hu *et al*. [6], be it that they introduce a dynamic convolutional layer to allow for dynamic adaptation of visual filters based on the input sentence. In effect making the textual embedding convolutional before the matching. Like [6, 15] we also propose an end-to-end trainable solution for segmentation from a sentence that embeds text and images into a joint model. Rather than relying on LSTMs we prefer a fully-convolutional model from the start, including dynamic filters. Moreover, we optimize our model for the task of segmenting an actor and its action in video, rather than in an image, allowing us to exploit both RGB and Flow.

The first and foremost contribution of this paper is the new task of actor and action segmentation from a sentence. As a second contribution we propose a fully-convolutional model for pixel-level actor and action segmentation using

an encoder-decoder neural architecture that is optimized for video and end-to-end trainable. Third, to show the potential of actor and action segmentation from a sentence we extend the A2D [29] and J-HMDB [9] datasets with more than 7,500 textual sentences describing the actors and actions appearing in the video content. And finally, our experiments demonstrate the quality of the sentence-guided segmentations, the generalization ability of our model, and its advantage for traditional actor and action segmentation compared to the state-of-the-art. Before detailing our model, we first discuss related work.

2. Related Work

2.1. Actor and action segmentation

Xu *et al*. [29] pose the problem of actor and action segmentation in video and introduce the challenging Actor-Action Dataset (A2D) containing a fixed vocabulary of 43 actor and action pairs. They build a multi-layer conditional random field model and assign to each supervoxel from a video a label from an actor-action product space. In [28], Xu and Corso propose a grouping process to add long-ranging interactions to the conditional random field. Yan *et al*. [31] show a multi-task ranking model atop supervoxel features allows for weakly-supervised actor and action segmentation using only video-level tags for training. Rather than relying on supervoxels, Kalogeiton *et al*. [10] propose a multi-task network architecture to jointly train an actor and action detector for a video. They extend their bounding box detections to pixel-wise segmentations by using state-of-the-art segmentation proposals [22] afterwards.

The above works are limited to model interactions between actors and actions from a fixed predefined set of label pairs. Our work models the joint actor and action space using an open set of labels as rich as language. This has the advantage that we are able to distinguish between fine-grained actors in the same super-category, *e.g*. a parrot or a duck rolling, and identify different actor and action instances. Thanks to a pre-trained word embedding, our model is also able to infer the segmentation from words that are outside of the actor and action vocabulary but exist in the embedding. Instead of generating intermediate supervoxels or segmentation proposals for a video, we follow a pixel-level model using an encoder-decoder neural architecture that is completely end-to-end trainable.

2.2. Actor localization from a sentence

Recently, works appeared that localize a human actor from an image [14] or video [30] based on a sentence. In [14], Li *et al*. introduce a person description dataset with sentence annotations and person samples from five existing person re-identification datasets. Their accompanying neural network model captures word-image relations and estimates the affinity between a sentence and a person image. Closer to our work is [30], where Yamaguchi *et al*. propose spatio-temporal person search in video. They supplement thousands of video clips from the ActivityNet dataset [1] with person descriptions. Their person retrieval model first proposes candidate tubes, ranks them based on a query in a joint visual-textual embedding and then outputs a final ranking. Similar to [14, 30], we also supplement existing datasets with sentence descriptions, in our case A2D [29] and J-HMDB [9], but for the purpose of actor *and* action segmentation. Where [30] demonstrates the value of sentences describing human actors for action localization in video, we generalize to actions performed by any actor. Additionally, where [14, 30], simplify their localization to a bounding box around the human actor of interest, we output a pixel-wise segmentation of both actor and action in video.

2.3. Action localization from a sentence

Both Gao *et al*. [4] and Hendricks *et al*. [5] consider retrieving a specific temporal interval containing actions via a sentence. In contrast, our work offers a unique opportunity to study spatio-temporal segmentation from a sentence, with a diverse set of actors and actions. Jain *et al*. [8] follow a zero-shot protocol and demonstrate spatio-temporal action localization is feasible from just a sentence describing a (previously unknown) action class. They first generate a set of action tubes, encode each of them by thousands of object classifier responses, and compute a word2vec similarity between the high-scoring object categories inside an action proposal and the action query. Mettes and Snoek [18] also follow a zero-shot regime and match sentences to actions in a word2vec space, but rather than relying on action proposals and object classifiers, they prefer object detectors only, allowing to query for spatio-temporal relations between human actors and objects. Different from their zero-shot setting, we operate in a supervised regime. We also aim for spatio-temporal localization of actions in video, but rather than generating bounding boxes, we prefer a pixel-wise segmentation over actions performed by any actor.

3. Model

Given a video and a natural language sentence as a query, we aim to segment the actor and its action in each frame of the video as specified by the query. To achieve this, we propose a model which combines both video and language information to perform pixel-wise segmentation according to the input query. We do so by generating convolutional dynamic filters from the textual representation and convolving them with the visual representation of different resolutions to output a segmentation mask. Our model consists of three main components: a textual encoder, a video encoder and a decoder, as illustrated in Figure 2.

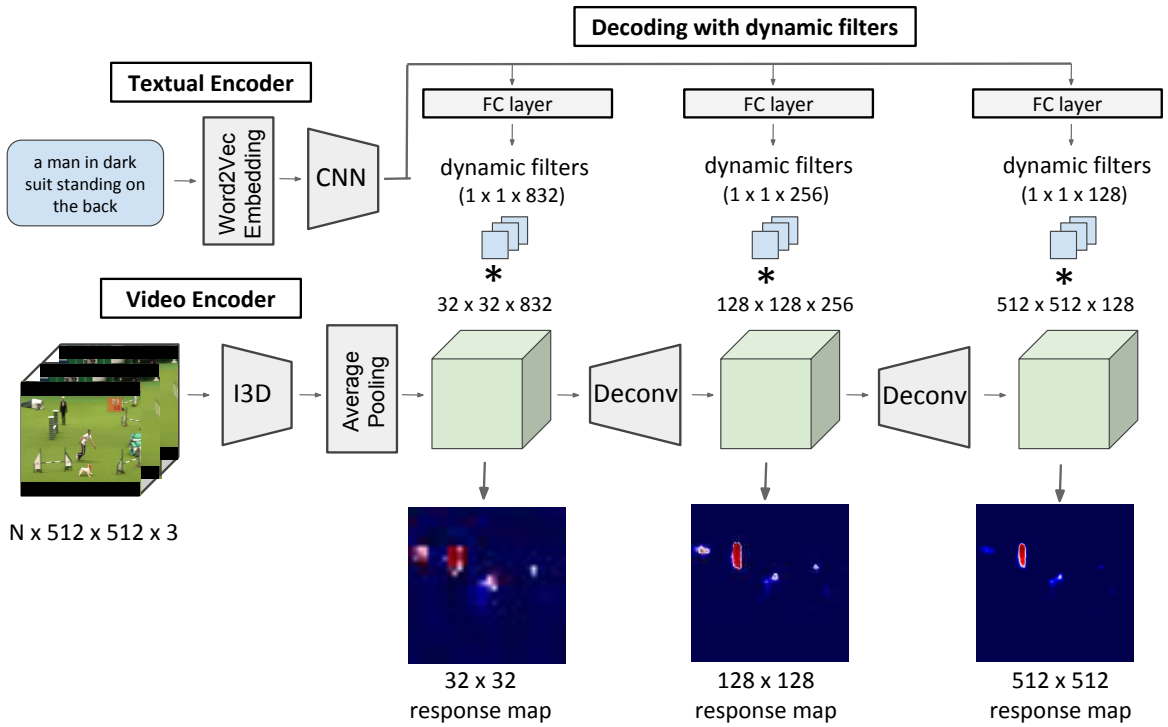

Figure 2: Our RGB model for actor and action video segmentation from a natural language sentence consists of three main components: a convolutional neural network to encode the expression, a 3D convolutional neural network to encode the video, and a decoder that performs a pixel-wise segmentation by convolving dynamic filters generated from the encoded textual representation with the encoded video representation. The same model is applied to the Flow input.

3.1. Textual Encoder

Given an input natural language sentence as a query that describes the actor and action, we aim to encode it in a way that enables us to perform segmentation of the specified actor and action in video. Different from [6, 15] who aim to train word embeddings from scratch on the ReferIt Dataset [12], we rely on word embeddings obtained from a large collection of text documents. Particularly, we are using a word2vec model pre-trained on the Google News Dataset [20]. It enables us to handle words beyond the ones of the sentences in the training set. In addition, we are using a simple 1D convolutional neural network instead of an LSTM to encode input sentences, which we will further detail in our ablation study.

Details. Each word of the input sentence is represented as a 300-dimensional word2vec embedding, without any further preprocessing. All the word embeddings are fixed without fine-tuning during training. The input sentence is then represented as a concatenation of its individual word representations, *e.g.* a 10-word sentence is represented by a 10×300 matrix. Each sentence is additionally padded to have the same size. The network consists of a single 1D convolutional layer with a temporal filter size equal to 2 and with the same output dimension as the word2vec representation. After the convolutional layer we apply the ReLU activation function and perform max-pooling to obtain a representation for the whole sentence.

3.2. Video Encoder

Given an input video, we aim to obtain a visual representation that encodes both the actor and action information, while preserving the spatial information that is necessary to perform pixel-wise segmentation. Different from [6, 15] who use a 2D image-based model our model takes advantage of the temporal dynamics of the video as well. Recently, Carreira and Zisserman [2] proposed to inflate the 2D filters of a convolutional neural network to 3D filters (I3D) to better exploit the spatio-temporal nature of video. By pre-training on both image object dataset ImageNet [23] and video action dataset Kinetics [11] their model achieves state-of-the-art results for action classification. We adopt the I3D model to obtain a visual representation from video.

Moreover, we also follow the well-known two-stream approach [24] to combine appearance and motion information, which was successfully applied earlier to a wide range of video understanding tasks such as action classifi-

cation [3, 27] and detection [21, 33]. We study the effect of having RGB and Flow inputs for actor and action segmentation in our ablation study.

Details. Frames of all videos are padded to have the same size. As visual feature representation for both the RGB and Flow input, we use the output of the inception block before the last max-pooling layer of the I3D network followed by an average pooling over the temporal dimension. To obtain a more robust descriptor at each spatial location, $L2$-normalization is applied to every spatial position in the feature map. Following [6, 15], we also append the spatial coordinates of each position as extra channels to the visual representation to allow learning spatial qualifiers like "left of" or "above".

3.3. Decoding with dynamic filters

To perform pixel-wise segmentation from a natural language sentence we rely on dynamic convolutional filters, as earlier proposed in [15]. Unlike static convolutional filters that are used in conventional convolutional neural networks, dynamic filters are generated depending on the input, in our case on the encoded sentence representation. It enables us to transfer textual information to the visual domain. Different from [15], we notice better results with a $\tanh$ activation function and $L2$-normalization on the features. In addition, we generate dynamic filters for several resolutions with different network parameters.

Given a sentence representation T, we generate dynamic filters f^r for each resolution $r \in R$ with a separate single layer fully-connected network:

$$f^r = \tanh(W_f^r T + b_f^r), \tag{1}$$

where $\tanh$ is the hyperbolic tangent function and f^r has the same number of channels as representation V_t^r for video input at timestep t and resolution r. Then the dynamic filters are convolved with V_t^r to obtain a pixel-wise segmentation response map for resolution r at timestep t:

$$S_t^r = f^r * V_t^r, \tag{2}$$

To obtain a segmentation mask with the same resolution as the input video, we further employ a deconvolutional neural network. Different from [6, 15], who apply deconvolution on the segmentation response maps, we use the deconvolutional layers on the video representation V_t^r directly. It enables us to better handle small objects and output smoother segmentation predictions. In addition, it helps to obtain more accurate segmentations for high overlap values as we will show in the experiments.

Details. Each of our deconvolutional networks consists of two blocks with one deconvolutional layer with kernel size 8×8 and stride 4, followed by a convolutional layer with a kernel size of 3×3 and a stride of 1. We use only the highest-resolution response map for the final segmentation prediction.

3.4. Training

Our training sample consists of an input video clip, an input sentence and a binary ground truth segmentation mask Y^r for each resolution $r \in R$ of the frame in the middle of each input video clip. For each training sample we define a loss, while taking into account multiple resolutions, which helps for better flow of gradients in the model similar to a skip-connection approach:

$$\mathcal{L} = \sum_{r \in R} \alpha_r \mathcal{L}^r \tag{3}$$

$$\mathcal{L}^r = \frac{1}{r^2} \sum_{i=1}^{r} \sum_{j=1}^{r} \mathcal{L}_{ij}^r \tag{4}$$

where α_r is a weight for resolution r. In this paper we consider $R = \{32, 128, 512\}$ and we further discuss the importance of using losses of all resolutions in our ablation study.

The pixel-wise $\mathcal{L}_{ij}^r$ loss is a logistic loss defined as follows:

$$\mathcal{L}_{ij}^r = log(1 + \exp(-S_{ij}^r Y_{ij}^r)) \tag{5}$$

where S_{ij}^r is a response value of our model at pixel (i, j) for resolution r and Y_{ij}^r is a binary label at pixel (i, j) for resolution r.

Details. We train our model using the Adam optimizer [13] with a learning rate of 0.001 and other parameters of the optimizer set to the default values. We divide the learning rate by 10 every $5,000$ iterations and train for $15,000$ iterations in total. We finetune only the last inception block of the video encoder.

4. Datasets

4.1. A2D Sentences

The Actor-Action Dataset (A2D) by Xu *et al*. [29] serves as the largest video dataset for the general actor and action segmentation task. It contains 3,782 videos from YouTube with pixel-level labeled actors and their actions. The dataset includes eight different actions, while a total of seven actor classes are considered to perform those actions. We follow [29], who split the dataset into 3,036 training videos and 746 testing videos.

As we are interested in pixel-level actor and action segmentation from sentences, we augment the videos in A2D with natural language descriptions about what each actor is doing in the videos. Following the guidelines set forth in [12], we ask our annotators for a discriminative referring expression of each actor instance if multiple objects

are considered in a video. The annotation process resulted in a total of 6,656 sentences, including 811 different nouns, 225 verbs and 189 adjectives. Our sentences enrich the actor and action pairs from the A2D dataset with finer granularities. For example, the actor *adult* in A2D may be annotated with *man*, *woman*, *person* and *player* in our sentences, while action *rolling* may also refer to *flipping*, *sliding*, *moving* and *running* when describing different actors in different scenarios. Our sentences contain on average more words than the ReferIt dataset [12] (7.3 vs 4.7), even when we leave out prepositions, articles and linking verbs (4.5 vs 3.6). This makes sense as our sentences contain a variety of verbs while existing referring expression datasets mostly ignore verbs.

4.2. J-HMDB Sentences

J-HMDB [9] contains 928 video clips of 21 different actions annotated with a 2D articulated human puppet that provides scale, pose, segmentation and a coarse viewpoint for the humans involved in each action. We augment the videos with sentences following the same protocol as for A2D Sentences. We ask annotators to return a natural language description of what the target object is doing in each video. We obtain 928 sentences, including 158 different nouns, 53 verbs and 23 adjectives. The most popular actors are *man*, *woman*, *boy*, *girl* and *player*, while *shooting*, *pouring*, *playing*, *catching* and *sitting* are the most popular actions.

We show sentence-annotated examples of both datasets in Figure 3 and provide more details on the datasets in the supplemental material. The sentence annotations and the code of our model will be available at `https://kgavrilyuk.github.io/publication/actor_action/`.

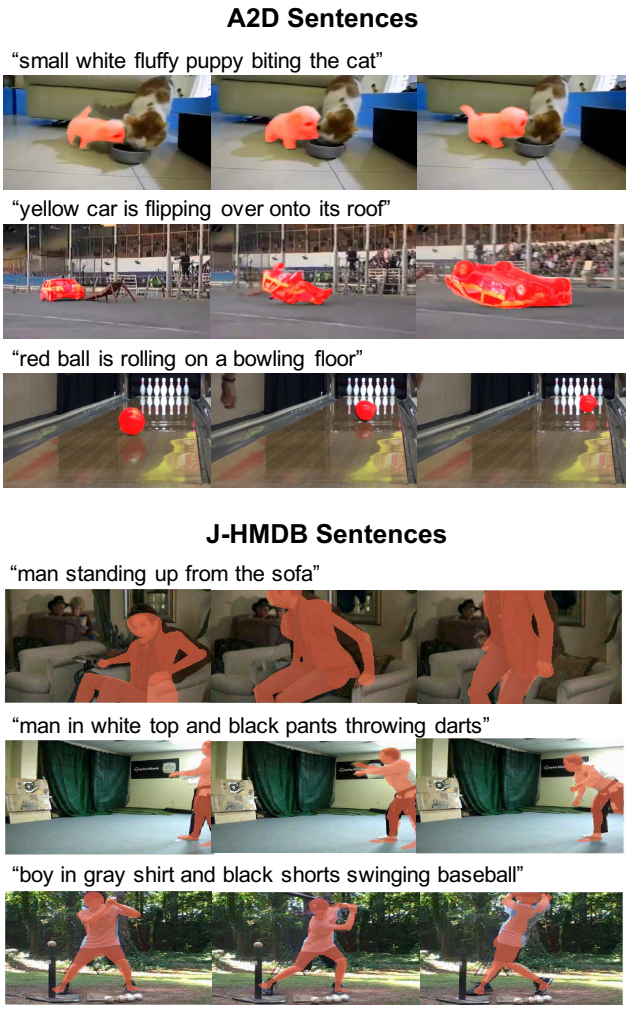

Figure 3: A2D Sentences and J-HMDB Sentences example videos, ground truth segments and sentence annotations.

5. Experiments

5.1. Ablation Study

In the first set of experiments we study the impact of individual components on our proposed model.

Setup. We select A2D Sentences for these set of experiments and use the train split for training and the test split for evaluation. The input to our model is a sentence describing what to segment and a video clip of N RGB frames around the frame to be segmented.

Evaluation. We adopt the widely used intersection-over-union (IoU) metric to measure segmentation quality. As aggregation metric we consider *overall IoU*, which is computed as total intersection area of all test data over the total union area.

Results on A2D Sentences. We first evaluate the influence of the number of input frames on our visual encoder and the segmentation result. We run our model with $N = 1, 4, 8, 16$ and we get 48.2%, 52.2%, 52.8%, and 53.6% respectively in terms of *overall IoU*. It reveals the important role of the large temporal context for actor and action video segmentation. Therefore, we choose $N = 16$ for all remaining experiments.

Next we compare our 1D convolutional textual encoder with an LSTM encoder. We follow the same setting for LSTM as in [6, 15], we use a final hidden state of LSTM as textual representation for the whole sentence. The dimension of the hidden state is set to $1,000$. We represent words by the same word2vec embedding model for both models. We observe that our simple 1D convolutional textual encoder outperforms LSTM in terms of *overall IoU*: 53.6% for our encoder and 51.8% for LSTM. We also experimented with bidirectional LSTM which slightly improves results over vanilla LSTM to 52.1%. Therefore, we select the convolutional neural network to encode the textual input

	Overlap					mAP	IoU	
	P@0.5	P@0.6	P@0.7	P@0.8	P@0.9	0.5:0.95	Overall	Mean
Hu *et al*. [6]	7.7	3.9	0.8	0.0	0.0	2.0	21.3	12.8
Li *et al*. [15]	10.8	6.2	2.0	0.3	0.0	3.3	24.8	14.4
Hu *et al*. [6] $\star$	34.8	23.6	13.3	3.3	0.1	13.2	47.4	35.0
Li *et al*. [15] $\star$	38.7	29.0	17.5	6.6	0.1	16.3	51.5	35.4
This paper: RGB	47.5	34.7	21.1	8.0	0.2	19.8	53.6	42.1
This paper: RGB + Flow	**50.0**	**37.6**	**23.1**	**9.4**	**0.4**	**21.5**	**55.1**	**42.6**

Table 1: Segmentation from a sentence on A2D Sentences. Object segmentation baselines [6, 15] as proposed in the original papers, or fine-tuned on the A2D Sentences train split (denoted by $\star$). Our model outperforms both baselines for all metrics. Incorporating Flow in our video model further improves results.

in the remaining experiments.

We further investigate the importance of our multi-resolution loss. We compare the setting when we are using all three resolutions to compute the loss ($\alpha_r = 1, r \in \{32, 128, 512\}$) with the setting when only the highest resolution is used ($\alpha_{32,128} = 0, \alpha_{512} = 1$). In terms of *overall IoU* the multi-resolution setting performs 53.6% while single resolution performs 49.4%. This demonstrates the benefit of the multi-resolution loss in our model.

In the last experiment we study the impact of the two-stream [24] approach for our task. We make a comparison for two type of inputs - RGB and Flow. For both streams we use 16 frames as input. The RGB stream produces better results than Flow: 53.6% for RGB and 49.5% for Flow. We then explore a fusion of RGB and Flow streams by computing a weighted average of the response maps from each stream. When we set the weight for RGB 2 times larger than Flow, it further improves our results to 55.1%.

5.2. Segmentation from a sentence

In this experiment, we segment a video based on a given natural language sentence on the newly annotated A2D Sentences and J-HMDB Sentences datasets and compare our proposed model with the baseline methods.

Setup. As there is no prior work for video segmentation from a sentence, we select two methods [6, 15], which can be used for the related task of image segmentation from a sentence, as our baselines. To be precise, we compare with the segmentation model of [6] and the lingual specification model of [15]. We report baseline results in two training settings. In the first one, the baselines are trained solely on the ReferIt dataset [12], as indicated in the original papers. In the second setting we further fine-tune the baseline models using the training videos from A2D Sentences. We train our model only on the train split of A2D Sentences. During test, we follow [29] and evaluate the models on each frame of the test videos for which segmentation annotation is available - around one to three frames per video. The input to both baseline models is an RGB frame with a sentence description. For our model, we use the same sentence as input but instead of a single RGB frame we employ 16 frames around the frame to be segmented as this setting shows the best results in our ablation study.

Evaluation. In addition to *overall IoU*, we also consider *mean IoU* as aggregation. The *mean IoU* is computed as the average over the IoU of each test sample. While the *overall IoU* favors large segmented regions, *mean IoU* treats large and small regions equally. In addition, following [6, 15], we also measure precision at five different overlap values ranging from 0.5 to 0.9 as well as the mean average precision over $.50 : .05 : .95$ [16].

Results on A2D Sentences. In Table 1, we report the results on the A2D Sentences dataset. The model of [6] and [15], pretrained on ReferIt [12], performs modestly as this dataset contains rich sentences describing objects, but it provides less information about actions. Fine-tuning these two baselines on A2D Sentences helps improve their performance by incorporating the notion of actions into the models. Our model outperforms both baselines for all metrics using RGB frames as input, bringing 3.5% absolute improvement in mAP, 2.1% in *overall IoU* and 6.7% in *mean IoU*. Fusion of RGB and Flow streams further improves our results. The larger improvement in *mean IoU* compared to *overall IoU* indicates our model is especially better on segmenting small objects. The results in mAP show the benefit of our model for larger overlap values. We visualize some of the sentence-guided segmentation results in Figure 4. First of all, our model can tackle the scenarios when the actor is not in the frame, *e.g*. in the second video. The model stops generating the segmentation once the man has left the camera's view. Our model can also tackle the scenarios when the actor is performing an action which is different from the one specified in the sentence, *e.g*. in the first video. The model doesn't output any segmentation for the frames in

Figure 4: Visualized segmentation results from our model on A2D Sentences. The first row shows a video with single actor and action, while the video in the second row contains similar types of actors performing the same action. In the third row, we illustrate a video with three sentences describing not only different actors, but also the same type of actor performing different actions. The colored segmentation masks are generated from the sentence with the same color above each video.

which the car is not in the *jumping* state. It shows the potential of our model for spatio-temporal video segmentation. Second, in contrast to segmentation from actor-action labels, we can see from the second video that our segmentation from a sentence enables to distinguish the instances of the same actor-action pair by richer descriptions. In the third video, our model confuses two dogs, still we easily segment different types of actors.

Results on J-HMDB Sentences. We further evaluate the generalization ability of our model and the baselines. We test the models, finetuned or trained on A2D Sentences, on all 928 videos of J-HMDB Sentences dataset without any additional finetuning. For each video, we uniformly sample three frames for evaluation following the same setting as in the previous experiment. We report our results in Table 2.

J-HMDB Sentences focuses exclusively on human actions and 4 out of 21 actions overlap with actions in A2D Sentences, namely *climb stairs*, *jump*, *walk*, and *run*. Consistent with the results on A2D Sentences, our method provides a more accurate segmentation for higher overlap values which is shown by *mAP*. We attribute the better generalization ability to two aspects. The baselines rely on the VGG16 [25] model to represent images, while we are using the video-specific I3D model. The second aspect comes from our textual representation, which can exploit similarity in descriptions of A2D Sentences and J-HMDB Sentences.

5.3. Segmentation from actor and action pairs

Finally, we segment a video from a predefined set of actor and action pairs and compare it with the state-of-the-art

	Overlap					mAP	IoU	
	P@0.5	P@0.6	P@0.7	P@0.8	P@0.9	0.5:0.95	Overall	Mean
Hu *et al*. [6]	63.3	35.0	8.5	0.2	0.0	17.8	**54.6**	52.8
Li *et al*. [15]	57.8	33.5	10.3	0.6	0.0	17.3	52.9	49.1
This paper	**69.9**	**46.0**	**17.3**	**1.4**	0.0	**23.3**	54.1	**54.2**

Table 2: Segmentation from a sentence on J-HMDB Sentences using best settings per model on A2D Sentences, demonstrating generalization ability. Our model generates more accurate segmentations for higher overlap values.

	Actor			Action			Actor and Action		
	Class-Average	Global	Mean IoU	Class-Average	Global	Mean IoU	Class-Average	Global	Mean IoU
Xu *et al*. [29]	45.7	74.6	-	47.0	74.6	-	25.4	76.2	-
Xu *et al*. [28]	58.3	85.2	33.4	60.5	85.3	32.0	43.3	84.2	19.9
Kalogeiton *et al*. [10]	**73.7**	90.6	49.5	60.5	89.3	42.2	47.5	88.7	29.7
This paper	71.4	**92.8**	**53.7**	**69.3**	**92.5**	**49.4**	**52.4**	**91.7**	**34.8**

Table 3: Semantic segmentation results on the A2D dataset using actor, action and actor+action as input respectively. Even though our method is not designed for this setting, it outperforms the state-of-the-art in most of the cases.

segmentation models on the original A2D dataset [29].

Setup. Instead of input sentences, we train our model on the 43 valid actor and action pairs provided by the dataset, such as *adult walking* and *dog rolling*. We use these pairs as textual input to our model. Visual input is kept the same as before. As our model explicitly requires a textual input for a given video, we select a subset of pairs from all possible pairs as queries to our model. For this purpose, we finetune a multi-label classification network on A2D dataset and select the pairs with a confidence score higher than 0.5. We use this reduced set of pairs as queries to our model and pick the class label with the highest response for each pixel. The classification network contains an RGB and a Flow I3D model where the number of neurons in the last layer is set to 43 and the activation function is replaced by a $sigmoid$ for multi-label classification. During training, we finetune the last inception block and the final layer of both models on random 64-frame video clips. We randomly flip each frame horizontally in the video clip and then extract a 224×224 random crop. We train for $3,000$ iterations with the Adam optimizer and fix the learning rate to 0.001. During test, we extract 32-frame clips over the video and average the scores across all the clips and across RGB and Flow streams to obtain the final score for a given video. For this multi-label classification we obtain mean average precision of 70%, compared to 67% in [29].

Evaluation. We report the class-average pixel accuracy, global pixel accuracy and *mean IoU* as in [10]. Pixel accuracy is the percentage of pixels for which the label is correctly predicted, either over all pixels (global) or first computed for each class separately and then averaged over classes (class-average).

Results on A2D. We compare our approach with the state-of-the-art in Table 3. Even though our method is not designed for this setting, it outperforms all the competitors for joint actor and action segmentation (last 3 columns of Table 3). Particularly, we improve the state-of-the-art by a margin of 4.9% in terms of class-average accuracy and 5.1% in terms of Mean IoU. In addition to joint actor and action segmentation, we report results for actor and action segmentation separately. For actor segmentation the method by Kalogeiton *et al*. [10] is slightly better in terms of class-average accuracy, for all other metrics and settings our method sets a new state-of-the-art. Our improvement is particularly notable on action segmentation where we outperform the state-of-the-art by 8.8% in terms of class-average accuracy and 7.2% in terms of Mean IoU. It validates that our method is suitable for both actor and action segmentation, be it individually or combined.

6. Conclusion

We introduce the new task of actor and action video segmentation from a sentence. Our encoder-decoder neural architecture for pixel-level segmentation explicitly takes into account the spatio-temporal nature of video. To enable sentence-guided segmentation with our model, we extended two existing datasets with sentence-level annotations describing actors and their actions in the video content. Experiments show the feasibility and robustness, as well as the model's ability to adapt to the task of semantic segmentation of actor and action pairs, outperforming the state-of-the-art.

References

[1] F. Caba Heilbron, V. Escorcia, B. Ghanem, and J. C. Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *CVPR*, 2015.

[2] J. Carreira and A. Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, 2017.

[3] C. Feichtenhofer, A. Pinz, and R. P. Wildes. Spatiotemporal multiplier networks for video action recognition. In *CVPR*, 2017.

[4] J. Gao, C. Sun, Z. Yang, and R. Nevatia. Tall: Temporal activity localization via language query. In *ICCV*, 2017.

[5] L. A. Hendricks, O. Wang, E. Shechtman, J. Sivic, T. Darrell, and B. Russell. Localizing moments in video with natural language. In *ICCV*, 2017.

[6] R. Hu, M. Rohrbach, and T. Darrell. Segmentation from natural language expressions. In *ECCV*, 2016.

[7] R. Hu, H. Xu, M. Rohrbach, J. Feng, K. Saenko, and T. Darrell. Natural language object retrieval. In *CVPR*, 2016.

[8] M. Jain, J. van Gemert, T. Mensink, and C. G. M. Snoek. Objects2action: Classifying and localizing actions without any video example. In *ICCV*, 2015.

[9] H. Jhuang, J. Gall, S. Zuffi, C. Schmid, and M. J. Black. Towards understanding action recognition. In *ICCV*, 2013.

[10] V. Kalogeiton, P. Weinzaepfel, V. Ferrari, and C. Schmid. Joint learning of object and action detectors. In *ICCV*, 2017.

[11] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, M. Suleyman, and A. Zisserman. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017.

[12] S. Kazemzadeh, V. Ordonez, M. Matten, and T. L. Berg. ReferIt game: Referring to objects in photographs of natural scenes. In *EMNLP*, 2014.

[13] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

[14] S. Li, T. Xiao, H. Li, B. Zhou, D. Yue, and X. Wang. Person search with natural language description. In *CVPR*, 2017.

[15] Z. Li, R. Tao, E. Gavves, C. G. M. Snoek, and A. W. M. Smeulders. Tracking by natural language specification. In *CVPR*, 2017.

[16] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014.

[17] J. Mao, H. Jonathan, A. Toshev, O. Camburu, A. Yuille, and K. Murphy. Generation and comprehension of unambiguous object descriptions. In *CVPR*, 2016.

[18] P. Mettes and C. G. M. Snoek. Spatial-aware object embeddings for zero-shot localization and classification of actions. In *ICCV*, 2017.

[19] P. Mettes, C. G. M. Snoek, and S.-F. Chang. Localizing actions from video labels and pseudo-annotations. In *BMVC*, 2017.

[20] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *NIPS*, 2013.

[21] X. Peng and C. Schmid. Multi-region two-stream r-cnn for action detection. In *ECCV*, 2016.

[22] P. O. Pinheiro, T.-Y. Lin, R. Collobert, and P. Dollár. Learning to refine object segments. In *ECCV*, 2016.

[23] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. Berg, and L. Fei-Fei. ImageNet large scale visual recognition challenge. *IJCV*, 2015.

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

[25] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.

[26] Y. Tian, R. Sukthankar, and M. Shah. Spatiotemporal deformable part models for action detection. In *CVPR*, 2013.

[27] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *ECCV*, 2016.

[28] C. Xu and J. J. Corso. Actor-action semantic segmentation with grouping process models. In *CVPR*, 2016.

[29] C. Xu, S.-H. Hsieh, C. Xiong, and J. J. Corso. Can humans fly? action understanding with multiple classes of actors. In *CVPR*, 2015.

[30] M. Yamaguchi, K. Saito, Y. Ushiku, and T. Harada. Spatio-temporal person retrieval via natural language queries. In *ICCV*, 2017.

[31] Y. Yan, C. Xu, D. Cai, and J. Corso. Weakly supervised actor-action segmentation via robust multi-task ranking. In *CVPR*, 2017.

[32] G. Yu and J. Yuan. Fast action proposals for human action detection and search. In *CVPR*, 2015.

[33] Y. Zhao, Y. Xiong, L. Wang, Z. Wu, X. Tang, and D. Lin. Temporal action detection with structured segment networks. In *ICCV*, 2017.

[34] T. Zhou and J. Yu. Natural language person retrieval. In *AAAI*, 2017.