Spatio-Temporal Activity Detection via Joint Optimization of Spatial and Temporal Localization

Md Atiqur Rahman and Robert Laganière
School of Electrical Engineering and Computer Science
University of Ottawa
Ottawa, Canada
{mrahm021, laganier}@uottawa.ca

Abstract

In this article, we address the problem of spatio-temporal activity detection which requires classifying as well as localizing human activities both in space and time from videos. To this end, we propose a novel single-stage and end-to-end trainable deep learning framework that can jointly optimize spatial and temporal localization of activities. Leveraging shared spatio-temporal feature maps, the proposed framework performs actor detection, activity tube building, as well as temporal localization of activities, all within a single network. The proposed framework outperforms the current state-of-the-art methods in spatio-temporal activity detection on the challenging UCF101-24 benchmark. By utilizing solely RGB input, it achieves a video-mAP of ***60.1%****, and further pushes the bar to* ***61.3%*** *when incorporating both RGB and FLOW inputs. Moreover, it attains a highly competitive frame-mAP of 74.9%.*

1. Introduction

Impressive strides have been made in human activity recognition in short and trimmed video clips that precisely surround the activity [3, 6, 7, 32, 38, 44]. However, real-world videos are often lengthy and contain untrimmed segments with a significant amount of temporal clutter. Consequently, understanding the contents of these videos requires the localization of the activities in time, which is known as temporal activity detection (TAD). While state-of-the-art (SOTA) TAD methods [1, 4, 12, 15, 16, 24, 45] have greatly advanced the fine-grain understanding of video contents, there are applications, such as autonomous driving, video surveillance, and advanced video search, that require not only the temporal boundaries of the activities but also the spatial extents of the actors within individual video frames to fully comprehend the scene dynamics. This essentially gives rise to the task called spatio-temporal activity detection (STAD) which requires classifying the activities while localizing them both in space and time in the input video. This paper focuses on tackling the STAD problem, which poses an even greater challenge compared to TAD, primarily due to the vast search space that needs to be explored in both spatial and temporal dimensions.

The typical recipe of STAD, as found in the literature, is mainly based on the object detection and linking pipeline, where actors are detected on individual video frames which are then linked via complex heuristics-based methods (e.g., dynamic programming, spatial overlap, or temporal sliding window). However, frame-based methods fail to fully capture the temporal structure of the activities and, as such, struggle to disambiguate activities that require the temporal contexts to comprehend (e.g., *sitting down* vs. *standing up*) [14]. Therefore, recent methods [10, 14] adopt a multi-frame approach that takes a short sequence of frames as input and performs localization of activities over short tubelets. However, due to the lack of direct temporal regression, these methods still need to rely on complex optimization for stitching the tubelets which may not result in optimal outputs [46]. Furthermore, the existing SOTA multi-frame methods [9, 35, 47] do not incorporate joint optimization for both spatial and temporal activity localization, instead relying on separate, disconnected pipelines. This disjoint approach incurs heightened computational demands likely due to redundant processing, increases the likelihood of sub-optimal outcomes, and most importantly, it impedes the ability to train these methods in a holistic, end-to-end fashion [29]. There has been limited research in this direction, and the few proposed methods, such as STAR [46], fell short of performance.

In this study, we aim to fill a gap in the existing literature by optimizing both the spatial and temporal localization of activities simultaneously. To achieve this goal, we introduce an end-to-end trainable single-stage STAD framework. This framework is built upon a two-stream 3D Convolutional Neural Network (3D CNN) that utilizes RGB and FLOW frames from the input video to create shared spatio-temporal feature maps.

Using these shared feature maps, we have integrated two branches within our framework. The first is dedicated to spatial localization, focusing on actor detection. The second branch handles temporal localization, pinpointing the timing of activities through direct temporal regression. Additionally, to enhance its ability to handle activities at varying spatial and temporal scales, our proposed framework incorporates a multi-scale architecture for both the spatial and temporal localization modules.

Contributions: The main contributions of this work are as follows:

- We propose a novel single-stage and end-to-end trainable deep learning framework to jointly optimize spatial and temporal localization of activities.
- We directly regress on temporal bounds of activities, and introduce spatio-temporal Non-maximum Suppression (NMS), a variant of the NMS technique, to improve the performance of the STAD task.
- We evaluate the effectiveness of our proposed approach on the challenging UCF101-24 benchmark and set new SOTA results on this benchmark.

2. Literature Review

The early approaches for STAD (e.g., [13,20]) attempted to extend the unsupervised 2D region proposal algorithms, (e.g, Selective search [39], Prime object proposal [19]) to their 3D counterparts with a view to generating supervoxels, which are then merged together based on color, texture, or motion information to form activity tubes. These activity tubes are then encoded with dense trajectories [40] followed by classification. Soomro *et al*. [36], on the other hand, used video segmentation to generate supervoxels and encoded them using bag-of-visual-words on improved dense trajectory features [41] before feeding them to classifiers. Nevertheless, supervoxel-based approaches have a drawback in terms of temporal accuracy, as supervoxels can span very lengthy time intervals.

Recent approaches address the STAD problem mainly in two separate stages – first, perform actor detection in individual video frames, then localize the activities in time based on the frame-based detections. Along this line, a number of methods [8, 21, 30, 34, 48, 50] used complex dynamic optimization to link the frame-level detections. Weinzaepfl *et al.* [42] applied temporal sliding stage over the frame-level detections to realize temporal localization. However, due to frame-based detections, these methods fail to fully capture the temporal structure of the activities. To overcome this issue, later methods [10, 14, 28] introduced clip-level detection on short video snippets. They aim to regress activity tubelets within these clips and subsequently link them to achieve temporal localization. However, these methods have limitations, especially when dealing with complex and prolonged activities. Their effectiveness is hindered by the need for optimizing tubelet linking and their inability to incorporate long-term temporal contexts. In contrast, our proposed method avoids the need for tubelet linking altogether by directly regressing activity over time.

Several recent methods have demonstrated superior ability in modeling longer temporal information, thus achieving better performance on the STAD task. Among the notable methods include TACNet [35], which proposed a temporal context detector to extract long-term contextual information and a transition-aware classifier to further distinguish the ambiguous states from real activity sequences. STEP [47], on the other hand, progressively processes longer sequences and adaptively extends proposals to follow the action movement. However, none of the above methods perform spatial and temporal regression in an end-to-end fashion, rather employs separate processing pipelines, thereby incurring increased and likely redundant computations. Moreover, the disjoint optimization of the two tasks possibly leads to sub-optimal results. A very recent work called STAR [46] addressed this issue by proposing an end-to-end pipeline for joint optimization of spatial and temporal localization of activities. However, STAR lacks a multi-scale model for spatio-temporal localization of activities, and is susceptible to failure in complex multi-actor scenarios due to it's reliance on a naive tube building heuristic.

Our proposed approach is closely aligned with STAR [46], as we simultaneously perform spatial and temporal regression of activities end-to-end. However, unlike STAR, we propose a single-stage pipeline that employs multi-scale feature hierarchy in an effort to capture actors and activities at varying scales. Moreover, STAR extracts features from the whole scene rather than tube–specific scene contexts during the temporal detection stage. This is likely to lead poor temporal localization for unrelated activities, mainly due to the confluence of the features from the whole scene. Furthermore, the simple heuristic used by STAR to form activity tubes may fail in complex multi-actor scenario. In our proposed approach, we aim to address these limitations.

3. Problem Statement

Given a temporally untrimmed long video sequence $\mathcal{I} = \{i_1, i_2, \ldots, i_T\}$ containing T frames, the goal of the STAD task is to output a set $\Psi = \{\psi_1, \psi_2, \ldots, \psi_N\}$ of predicted spatio-temporal activity segments such that a predicted activity segment $\psi_n = (\phi^n_{start}, \phi^n_{end}, c^n, p^n, R^n = \{r^n_1 \ldots r^n_M\})$ has associated with it, the activity (start, end)

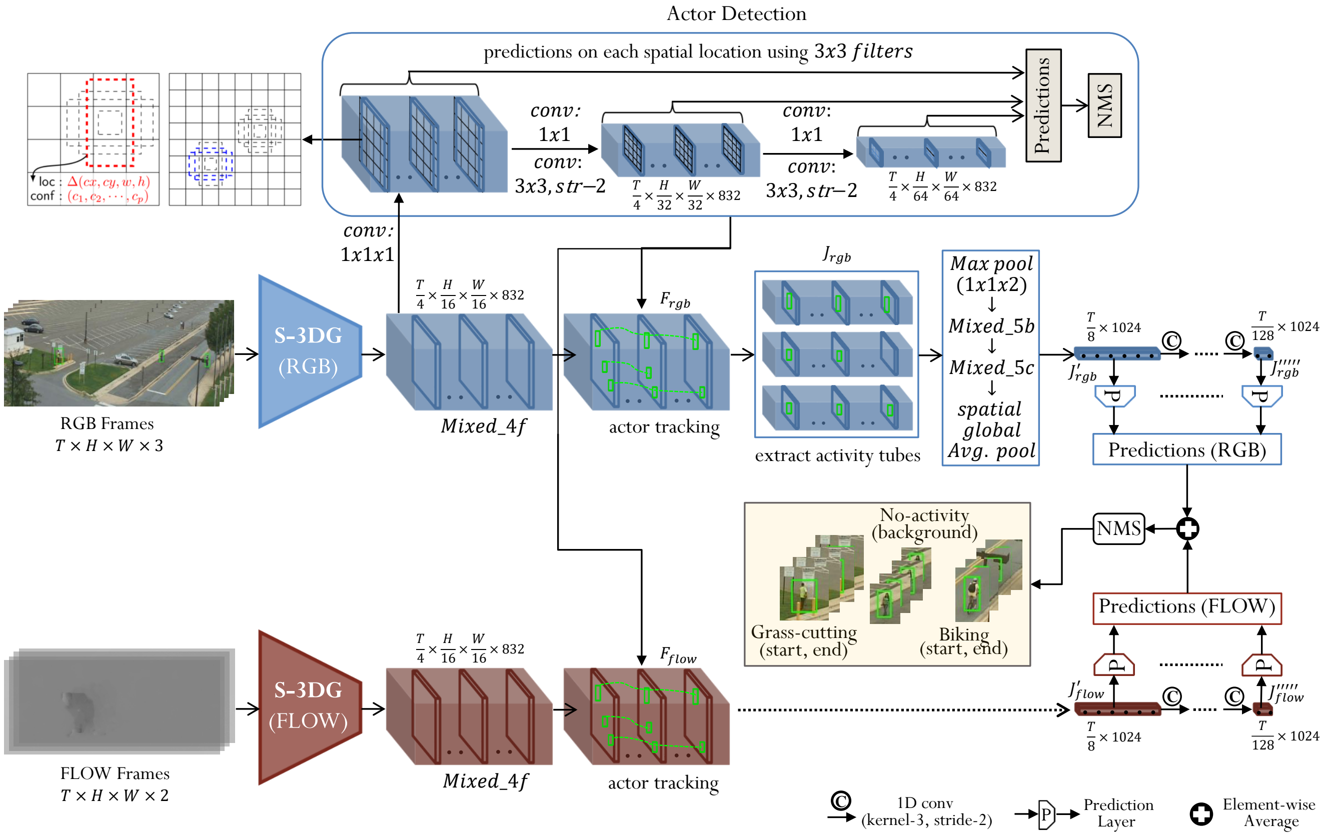

Figure 1. Architecture of the proposed end-to-end STAD framework. Building upon a two-stream 3D CNN called S-3DG [44], we add a spatial localization branch on top of the *'Mixed_4f'* block of the RGB stream, employing a multi-scale feature hierarchy on the temporal slices of *'Mixed_4f'* to perform actor detection followed by tracking of the actors. The actor tracks are then projected back to the *'Mixed_4f'* blocks of both RGB and FLOW streams to extract actor tube features, which are then fed to a multi-scale temporal localization branch for direct temporal regression of activities.

times $(\phi^n_{start}, \phi^n_{end})$, the predicted activity category $c^n \in \{1, \dots, C\}$, the prediction confidence score p^n, as well as a set $R^n = \{r^n_1 \dots r^n_M\}$ consisting of M predicted bounding boxes corresponding to the actor on each of the M frames belonging to the video segment between ϕ^n_{start} and ϕ^n_{end}. Here, C denotes the total number of activity categories.

4. Proposed Approach

The proposed STAD framework is illustrated in Fig. 1, and comprises five main modules – (i) base network, (ii) spatial localization branch, (iii) actor tube building, (iv) temporal localization branch, and (v) sptio-temporal NMS.

4.1. Base Network

The proposed STAD framework leverages a SOTA two-stream 3D CNN called S-3DG [44] that was originally proposed for video activity recognition. We repurpose this two-stream 3D CNN for encoding video features by extracting spatio-temporal feature representation of the input video. As shown in Fig. 1, the RGB stream of S-3DG is fed with a video segment $I_{\text{rgb}} \in R^{T \times H \times W \times 3}$ consisting of T number of RGB frames, each with height H, width W, and channel dimension 3. The FLOW stream, on the other hand, takes the optical FLOW frames $I_{\text{flow}} \in R^{T \times H \times W \times 2}$ corresponding to the RGB frames as input, thus having the same temporal and spatial dimensions as the RGB input but with channel dimension 2 (i.e., FLOW in X and Y directions). The basic idea behind this two-stream architecture is to capture the appearance information of the video through the RGB stream, while harvesting the motion information through the FLOW stream. We extract rich two-stream spatio-temporal feature representations $(F_{\text{rgb}}, F_{\text{flow}}) \in R^{\frac{T}{4} \times \frac{H}{16} \times \frac{W}{16} \times 832}$ of the input video from the *Mixed_4f* block of S-3DG. We then exploit $(F_{\text{rgb}}, F_{\text{flow}})$ as base feature maps that are shared among the other modules, thus allowing these feature maps to be learned end-to-end with respect to the overall objective of the STAD task. The base network being fully-convolutional, the length of the video sequence T can be arbitrarily long, consequently constrained only by the amount of physical memory.

4.2. Spatial Localization Branch

The spatial localization branch is responsible for detecting the actors in the video frames. To this end, we feed the base RGB feature map F_{rgb} as input to this branch, after having it passed through a $1\times1\times1$ convolution layer as shown in Fig. 1. We then employ a multi-scale feature hierarchy on top of the temporal slices of F_{rgb} with a view to capturing actors (i.e., persons) at varying scales. To this end, we fuse the temporal dimension of F_{rgb} with the batch dimension before repeatedly applying 1×1 convolutions (with stride 1), followed by 3×3 convolutions (with stride 2) to generate a feature hierarchy having 3 different scales.

Leveraging the feature hierarchy, we build a multi-scale and anchor-based prediction architecture as commonly used in the single-stage object detection methods (e.g., [17, 25, 26]). Anchor-based object detection employs a set of default bounding boxes with varying scales and aspect ratios at each spatial location in a feature map. Each default box has it's own default center, width, and height. Predictions about the activity class confidence scores for each actor along with the actor widths and heights w.r.t. the default boxes' widths and heights are then produced using 3×3 convolutions as shown on the top-left part in Fig. 1. Finally, the predictions are post-processed using NMS.

4.3. Actor Tube Building

With the actors detected and localized by the spatial localization branch, each actor is tracked using an online tracker called DeepSORT [43]. The actor tracks are then projected back to the two-stream base feature maps $(F_{\text{rgb}},\ F_{\text{flow}})$ to extract spatio-temporal features corresponding to the tubes containing the actors. To this end, for each actor track, the spatial region corresponding to the actor is cropped out of the temporal slices of $(F_{\text{rgb}},\ F_{\text{flow}})$ in order to create two-stream spatio-temporal tube feature maps $(J_{\text{rgb}},\ J_{\text{flow}})$ having the same temporal and channel dimensions as $(F_{\text{rgb}},\ F_{\text{flow}})$. The spatial region is cropped based on the minimal square box that covers all detections of the actor in the track, after having the box slightly expanded to capture sufficient contexts around the actor.

4.4. Temporal Localization Branch

The purpose of the temporal localization branch is to locate the activities in time and classify them. To this end, the tube feature maps $(J_{\text{rgb}},\ J_{\text{flow}})$ of all detected actors are arranged into a batch after having them resized spatially to the dimension $D\times D$. These resized feature maps are then *max-pooled* using $1\times1\times2$ filter to reduce the temporal dimension by half. They are subsequently passed through the *Mixed_5b* and *Mixed_5c* blocks of S-3DG before their spatial dimension is completely collapsed using *global spatial average pooling* to produce two-stream temporal-only feature maps $(J^{'}_{\text{rgb}},\ J^{'}_{\text{flow}}) \in R^{\frac{T}{8}\times1024}$. We then feed $(J^{'}_{\text{rgb}},\ J^{'}_{\text{flow}})$ to a temporal activity detection (TAD) pipeline proposed in [24] to realize temporal localization and classification of the activities.

To briefly recap, the TAD pipeline employs a multi-scale temporal feature hierarchy based on 1D temporal convolutional layers (kernel size 3, strides 2) cascaded on top of the feature maps $(J^{'}_{\text{rgb}},\ J^{'}_{\text{flow}})$ to produce four two-stream and temporal-only feature maps with decreasing temporal resolution and increasing scale, namely, $(J^{''}_{\text{rgb}},\ J^{''}_{\text{flow}}) \in R^{\frac{T}{16}\times1024} \ldots\ldots (J^{'''''}_{\text{rgb}},\ J^{'''''}_{\text{flow}}) \in R^{\frac{T}{128}\times1024}$. Analogous to the spatial localization branch, a set of K default temporal segments with varying scales are employed at each temporal location on each feature map in the temporal feature hierarchy. Predictions about the activity segments are made on top of each feature map for both streams separately, using 1D convolutional filters (kernel size 3, stride 1). These predictions are then combined using element-wise averaging as shown in bottom-right of Fig. 1

To be specific, at each feature location, the following predictions are made – i) activity class scores $\{p_j\}_{j=1}^{C}$ over C activity classes, ii) the center offset Δ_m and width offset Δ_w w.r.t. the default center d_m and default width d_w of the default activity segment; and iii) an overlap score p_{ov} indicating the overlap between the default activity segment and the closest ground-truth segment. p_{ov}, after having it passed through a *sigmoid* function, is used as a confidence value for ranking the predictions during inference. Finally, following [15], the start time ϕ_{start} and end time ϕ_{end} of the predicted activity segment are computed as follows:

$$\phi_m = d_m + \alpha_1 d_w \Delta_m \ \text{ and } \ \phi_w = d_w \exp(\alpha_2 \Delta_w) \quad (1)$$

$$\phi_{start} = \phi_m - \frac{\phi_w}{2} \ \text{ and } \ \phi_{end} = \phi_m + \frac{\phi_w}{2} \quad (2)$$

Here, ϕ_m and ϕ_w refer to the actual center and width of the predicted activity segment, while α_1 and α_2 are hyperparameters used to control the effect of Δ_m and Δ_w, respectively.

Since the TAD pipeline employs multi-scale temporal feature hierarchy, our proposed STAD framework is capable of handling wide variations in activity lengths, a common phenomenon for activity detection from realistic videos. Please refer to the original work [24] for additional details about the TAD pipeline.

4.5. Spatio-Temporal NMS

We introduce spatio-temporal NMS in this work to post-process the activity predictions generated by the temporal

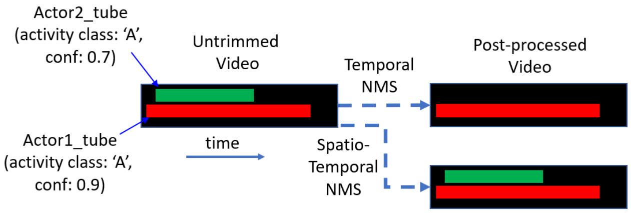

Figure 2. Illustration of the spatio-temporal NMS technique. Temporal NMS would remove 'Actor2_tube' as it overlaps in time with the higher confidence and same class prediction 'Actor1_tube'. The proposed spatio-temporal NMS would retain both tubes as they do not overlap both in time and space.

localization branch. Unlike the temporal NMS technique commonly used in the literature (e.g., [1, 4, 12, 15, 18, 24]), which cancels out duplicate temporal segments solely based on temporal overlaps among candidate predictions, the proposed spatio-temporal NMS eliminates duplicates by considering both spatial and temporal overlaps among the predicted spatio-temporal segments. This is motivated by the observation that in an untrimmed and unconstrained video, activities of the same type may occur concurrently, with the actors located spatially close to each other (e.g., two persons cycling side-by-side). Under this scenario, the temporal NMS technique would remove all but the temporal segment with the highest confidence score, as they overlap in time and have the same class prediction. On the other hand, the proposed spatio-temporal NMS technique would be able to retain all spatio-temporal segments as long as they do not overlap both in space and time. This is illustrated in Fig. 2.

4.6. Training and Inference:

4.6.1 Loss Function

The proposed STAD framework was trained using a multi-task loss $\mathcal{L}$ having two components – object detection loss $\mathcal{L}^{\text{obj}}$, and temporal detection loss $\mathcal{L}^{\text{tem}}$. $\mathcal{L}^{\text{obj}}$, in turn, comprises object classification loss $\mathcal{L}^{\text{obj}}_{\text{cls}}$, and object localization loss $\mathcal{L}^{\text{obj}}_{\text{loc}}$ (for actor width, height, center-X, and center-Y). On the other hand, $\mathcal{L}^{\text{temp}}$ consists in temporal classification loss $\mathcal{L}^{\text{tem}}_{\text{cls}}$, temporal localization loss $\mathcal{L}^{\text{tem}}_{\text{loc}}$ (for activity start and end times), and temporal overlap loss $\mathcal{L}^{\text{tem}}_{\text{ov}}$. Equation (3) shows the formulation of the loss function.

$$\begin{aligned}\mathcal{L} &= \mathcal{L}^{\text{obj}} + \lambda\mathcal{L}^{\text{tem}} \\ &= \mathcal{L}^{\text{obj}}_{\text{cls}} + \mathcal{L}^{\text{obj}}_{\text{loc}} + \lambda(\mathcal{L}^{\text{tem}}_{\text{cls}} + \mathcal{L}^{\text{tem}}_{\text{loc}} + \mathcal{L}^{\text{tem}}_{\text{ov}})\end{aligned} \tag{3}$$

Here, λ is a hyper-parameter used to trade-off between the two loss components. We used *multi-class cross-entropy loss* for $\mathcal{L}^{\text{obj}}_{\text{cls}}$ and $\mathcal{L}^{\text{tem}}_{\text{cls}}$, while *Smooth-L1 loss* was used for $\mathcal{L}^{\text{obj}}_{\text{loc}}$, $\mathcal{L}^{\text{tem}}_{\text{loc}}$, and $\mathcal{L}^{\text{tem}}_{\text{ov}}$.

4.6.2 Final Predictions

The actor detection from the spatial localization branch is available at every l^{th} frame, which is dictated by the temporal stride of the input feature map F_{rgb} and the frame sampling rate. For example, since F_{rgb} has a temporal stride of 4, l will be $4/8/16$ for a frame sampling rate of $1/2/4$. Therefore, in order to obtain the detections on the intermediate frames, we use linear interpolations. Finally, an spatio-temporal activity prediction instance ψ_n, as defined in Sec. 3, is made up by combining the actor bounding boxes $R^n = \{r_1^n \dots r_M^n\}$ with the activity predictions $(\phi^n_{start}, \phi^n_{end}, c^n, p^n)$ as output from the spatio-temporal NMS module, such that $\psi_n = (\phi^n_{start}, \phi^n_{end}, c^n, p^n, R^n = \{r_1^n \dots r_M^n\})$.

5. Experimental Setup

5.1. Dataset

To evaluate the proposed STAD framework, we performed experiments on the challenging UCF101-24 [27] benchmark, which is one of the largest and most diversified and challenging spatio-temporal action detection datasets containing temporally untrimmed videos. It is derived from the UCF101 [37] dataset by providing spatio-temporal annotations for 24 classes in the form of bounding box and tube annotations for the actors. The average number of action instances in a video is 1.5, with each action instance spanning 70% of the video duration on average. However, certain classes may have instances with an average duration as low as 30% of the video length. Keeping aligned with the standard practice on this dataset, and to be able to compare results with the SOTA methods, we used the revised annotations provided by Singh *et al.* [34] that includes 3,207 videos, with 2,293 for training and 914 for testing.

5.2. Evaluation Metrics

The proposed approach is evaluated using the established performance metric for the STAD task, which is mean average precision (mAP), both at the video-level and frame-level, as documented in existing literature [14, 31, 34, 42].

video-mAP is used to evaluate the performance of the STAD task at the video-level detections, and involves regressing a series of temporally linked bounding boxes, also known as "activity tubes", along with the relevant class label. It is defined as the mean of the average precision (AP) over all classes, where AP for a specific class is defined as the area under the precision-recall curve at a specific Intersection-over-Union (IoU) threshold. For video-mAP, the IoU is computed as the product of the temporal IoU between ground-truth and predicted activity tubes and the average of the spatial IoUs between the ground-truth and predicted bounding boxes.

frame-mAP is used to evaluate the detection performance at the frame-level and involves detecting the instance bounding boxes in each video frame along with the associated class label. It is defined analogously as video-mAP except that the IoU is computed spatially.

5.3. Implementation Details

We set the input sequence length $T = 352$ frames for the best model configuration (i.e., sampling every frame). During training, input images first undergo center-cropping to the desired resolution (e.g., 224x224 for the best model configuration), followed by various data augmentations including random consistent horizontal flipping, random cropping with aspect-ratio resizing. In random consistent left-right flipping, either all or none of the frames belonging to the input sequence are flipped horizontally. However, no data augmentation is applied during test. The base network S-3DG is initialized with pre-trained weights from the Kinetics-600 [2] dataset, while the spatial and temporal localization branches are initialized randomly. The batch size for the temporal branch is set to 4, thus generating a batch size of $4 \times T$ for the spatial branch. The spatial branch employs anchors with 6 different scales linearly ranging from $0.2 - 0.95$ and 3 different aspect ratios $\{0.5, 1, 2\}$. On the other hand, the number of default temporal segments K for the temporal branch is set to 5 with scale ratios $\{0.5, 0.75, 1.0, 1.5, 2.0\}$. The whole framework is trained using Adam optimizer with a fixed learning rate of 0.0005. D for feature cropping in the temporal localization branch is set to 10. To generate consistent actor tubes, DeepSORT [43] is applied on detections having a confidence score above 0.3. The whole framework was implemented based on the TensorFlow Object Detection API [11] and trained on 2× NVIDIA RTX 3090 GPUs.

6. Results

In this section, we present the results of our proposed approach on the STAD task based on the UCF101-24 dataset. We first study different model configurations and perform ablation studies in order to validate our design choices while also allowing us to determine the optimal model configurations. We then compare the results with the SOTA methods based on the optimal model configuration.

6.1. Model Configurations

Different input configurations were explored with a view to selecting the best model configuration. To be specific, the effect of spatial resolution, temporal sampling rate, as well as the impact of FLOW inputs were studied. Table 1 shows the results of the proposed STAD framework based on the different model configurations. The top and middle part of the table show results as the spatial resolution and temporal sampling rate of the input video are varied, respectively. Fixing upon the best configurations (i.e., frame resolution of 224×224, while sampling every frame), the bottom part of the table shows the impact of optical FLOW for the STAD task. As revealed from the table, higher spatial resolution and increased frame rate contributes to superior spatio-temporal detection of the activities, with the addition of optical FLOW input further boosting the performance. The optimal model configuration with RGB and FLOW inputs achieves a frame-mAP of **74.9%**, while reaching a video-mAP of **61.3%**.

Table 1. Exploration study on the model configuration w.r.t. spatial and temporal resolution as well as input modalities based on the UCF101-24 dataset. Both mAP values reported at IoU=0.5.

Input	Resolution	Sampling Rate	Frame mAP (%)	Video mAP (%)
RGB	160x160	every frame	65.7	54.8
	192x192		71.5	58.2
	224x224		**74.9**	**60.1**
RGB	224x224	every 2nd frame	69.4	57.5
		every 4th frame	64.2	53.1
RGB+FLOW	**224x224**	**every frame**	**74.9**	**61.3**

Table 2. Ablation results on the effect of temporal localization and spatio-temporal NMS based on RGB input with IoU=0.5.

Ablation Experiment	Video mAP (%)
Temporal Classification	56.4
Temporal Localization	**60.1**
Temporal NMS	59.7
Spatio-Temporal NMS	**60.1**

6.2. Ablation Study

6.2.1 Effect of Temporal Localization

In order to validate our design choice of performing end-to-end temporal localization, we conducted experiments with and without temporal localization of the discovered actor tubes. To this end, the best model configuration with RGB input, as shown in top part of Tab. 1, was used to train two separate models – one employing temporal localization and classification of the activities in the discovered actor tubes, and the other employing only classification of the actor tubes without performing any temporal localization. Top part of Tab. 2 shows the results of this exercise which reveals that temporal localization is important to precisely localize the start and end times of each activity as it improves the performance of the STAD task by an absolute 3.7% video-mAP.

The impact of temporal localization on the STAD task would be more pronounced in situations where an actor performs consecutive sequential activities. For example, in an autonomous driving scenario, understanding a pedestrian's activities around an intersection will involve localizing the pedestrian in time as it waits to cross the intersection, followed by localizing the actual crossing activity in time, both of which will be performed consecutively by the same actor. As a result, the inclusion of direct temporal regression to localize the activities is important for real-world situations.

6.2.2 Effect of Spatio-Temporal NMS

We also performed experiments to tease out the impact of using spatio-temporal NMS, as opposed to temporal NMS. To this end, similar to the ablation study of temporal localization, the best model configuration with RGB input was evaluated under two different settings – using temporal NMS, and using spatio-temporal NMS. The bottom part of Tab. 2 shows the results of the ablation which indicates that use of spatio-temporal NMS improves the video-mAP of the proposed STAD framework by 0.4% over using temporal NMS.

6.3. State-of-the-Art Comparison

We perform comparisons of our proposed STAD framework with the SOTA methods on the STAD task based on the UCF101-24 dataset. Table 3 shows the comparisons in terms of frame-mAP at the standard IoU threshold of 0.5 and video-mAP at IoU thresholds of 0.2 and 0.5. As obvious from the table, the multi-frame methods generally perform better than the single-frame methods when it comes to video-mAP, primarily because these methods enjoy better temporal localization by leveraging multiple frames as input.

Our proposed method, which employs end-to-end temporal localization of activities as opposed to localization and linking of tubelets, outperforms the SOTA methods in video-mAP while producing competitive results in frame-mAP. We achieved a video-mAP of **60.1%** at IoU=0.5 using RGB input only; with the addition of FLOW input further pushing the video-mAP to **61.3%**. The proposed method achieved a frame-mAP of **74.9%**, trailing by a mere 1.4% mAP from the SOTA results. It is noteworthy to mention that the methods that achieved a higher frame-mAP than ours either used an external object detector for spatial localization of the actors (e.g., Gu *et al.* [9]), or relied on much higher input resolutions (e.g., 400×400 for STEP [47], and 512×682 for 3D-RetinaNet [33]), thus incurring increased computations.

Table 4 shows class-wise video-mAP for the best model configuration.

Table 3. Comparison of the SOTA methods for spatio-temporal activity detection on the UCF101-24 dataset based on frame-mAP (IoU=0.5) and video-mAP (IoU=0.2 and 0.5). 'R' and 'F' denote RGB and FLOW, respectively.

Temporal Localization			Frame mAP (%)	Video mAP (%)	
	Method	Input	0.5	0.2	0.5
Frame-based	Peng *et al.* [22]	R+F	39.6	42.3	-
	Saha *et al.* [31]	R+F	-	66.8	35.9
	Weinzaepfel *et al.* [42]	R+F	-	58.9	-
	AMTnet [28]	R+F	-	78.5	49.7
	Gurkirt *et al.* [34]	R+F	-	73.5	46.3
	Pramono *et al.* [23]	R+F	73.7	80.4	49.5
	Zhao *et al.* [49]	R+F	-	78.5	50.3
Tubelet linking	Chéron *et al.* [5]	R+F	-	76.0	50.1
	Gu *et al.* [9]	R+F	**76.3**		59.9
	T-CNN [10]	R	41.4	-	47.1
	ACT [14]	R+F	67.1	77.2	51.4
	TACNet [35]	R+F	72.1	77.5	52.9
	STEP [47]	R+F	75.0	76.6	-
	3D-RetinaNet [33]	R	75.2	82.4	58.2
End-to-End	STAR [46]	R	63.0	77.9	53.0
	Ours	R	74.9	82.5	**60.1**
	Ours	R+F	74.9	**83.4**	**61.3**

6.4. Qualitative Results

Figure 3 shows some sample qualitative results of the proposed framework on two test videos from the UCF101-24 dataset. Top row shows predictions on a test video from the 'Diving' class with a length of 9.3s, while the bottom row shows predictions for the 'TrampolineJumping' class over a 8.3s long video. As can be seen, our proposed model is able to detect the actor tubes while localizing the temporal bounds of the activities with higher precision. It is also obvious from these visualizations that the proposed method is capable of adapting to varying scales of the actors as well as lengths of the activities, thanks to the multi-scale architecture incorporated in the spatial and temporal localization modules of the framework.

6.5. Inference Speed

Since it is devoid of any external object detector, combined with the fact that the spatio-temporal feature maps are shared among the spatial and temporal localization modules, our proposed approach runs at a moderately higher speed, achieving a frame rate of 250 frames per second with an input size of 224 × 224 on NVIDIA RTX 3090 GPU.

7. Conclusion

In this paper, we presented a novel end-to-end STAD framework that is capable of performing joint optimization of spatial and temporal localization of activities from

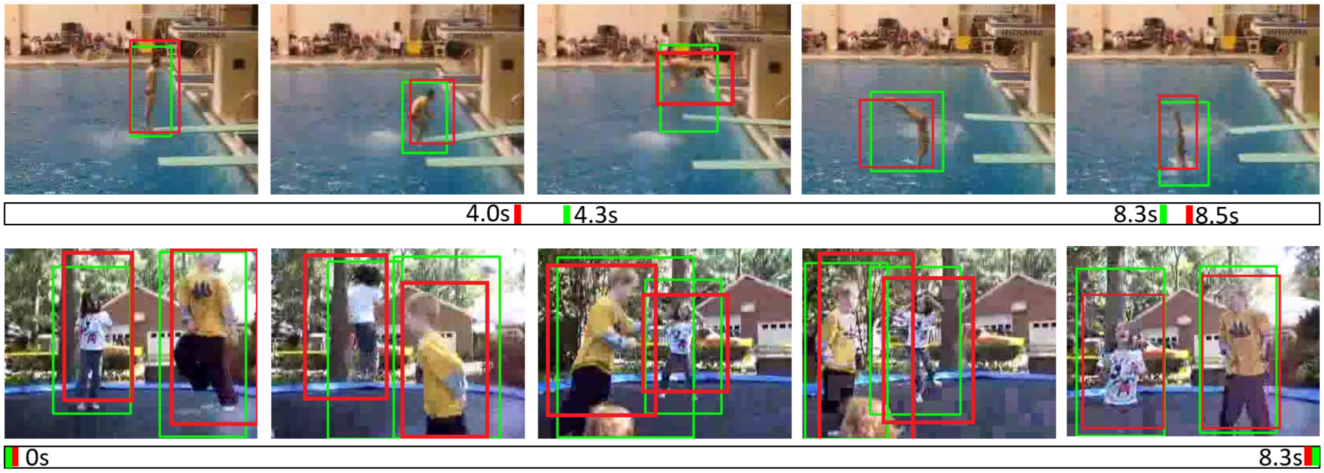

Figure 3. Visualization of spatio-temporal detections generated by the proposed framework on two test videos from the UCF101-24 dataset. Green and red denote ground-truth and predictions, respectively.

Table 4. Class-wise video-mAP (%) at IoU=0.5 on the UCF101-24 dataset.

Basketball	Basketball Dunk	Biking	Cliff Diving	Cricket Bowling	Diving	Fencing	Floor Gymnastics	Golf Swing	Horse Riding	Ice Dancing	Long Jump	PoleVault	Rope Climbing	Salsa Spin	Skate Boarding	Skiing	Skijet	Soccer Juggling	Surfing	Tennis Swing	Trampoline Jumping	Volleyball Spiking	Walking With Dog
71.1	76.8	56.9	75.2	71.6	85.7	46.3	84.6	76.4	81.1	46.8	78.7	73.5	74.9	35.4	74.5	29.5	30.1	61.3	29.4	56.5	22.1	59.9	77.8

temporally untrimmed videos. Leveraging shared feature maps, and multi-scale spatial and temporal feature hierarchy, the proposed framework achieved new SOTA results on the highly challenging UCF101-24 benchmark.

References

[1] Shyamal Buch, Victor Escorcia, Bernard Ghanem, Li Fei-Fei, and Juan Carlos Niebles. End-to-end, single-stream temporal action detection in untrimmed videos. In *BMVC*, 2017. 1, 5

[2] João Carreira, Eric Noland, Andras Banki-Horvath, Chloe Hillier, and Andrew Zisserman. A short note about kinetics-600. *CoRR*, abs/1808.01340, 2018. 6

[3] J. Carreira and A. Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, 2017. 1

[4] Yu-Wei Chao, Sudheendra Vijayanarasimhan, Bryan Seybold, David A. Ross, Jun Deng, and Rahul Sukthankar. Rethinking the faster r-cnn architecture for temporal action localization. In *CVPR*, 2018. 1, 5

[5] Guilhem Chéron, Jean-Baptiste Alayrac, Ivan Laptev, and Cordelia Schmid. A flexible model for training action localization with varying levels of supervision. In *NeurIPS*, 2018. 7

[6] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. *CoRR*, abs/1812.03982, 2018. 1

[7] Rohit Girdhar, Deva Ramanan, Abhinav Gupta, Josef Sivic, and Bryan Russell. Actionvlad: Learning spatio-temporal aggregation for action classification. In *CVPR*, 2017. 1

[8] Georgia Gkioxari and Jitendra Malik. Finding action tubes. In *CVPR*, pages 759–768. IEEE Computer Society, 2015. 2

[9] Chunhui Gu, Chen Sun, David A. Ross, Carl Vondrick, Caroline Pantofaru, Yeqing Li, Sudheendra Vijayanarasimhan, George Toderici, Susanna Ricco, Rahul Sukthankar, Cordelia Schmid, and Jitendra Malik. Ava: A video dataset of spatio-temporally localized atomic visual actions. In *CVPR*, 2018. 1, 7

[10] Rui Hou, Chen Chen, and Mubarak Shah. Tube convolutional neural network (t-cnn) for action detection in videos. In *ICCV*, 2017. 1, 2, 7

[11] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama, and K. Murphy. Speed/accuracy trade-offs for modern convolutional object detectors. In *CVPR*, 2017. 6

[12] Yupan Huang, Qi Dai, and Yutong Lu. Decoupling localization and classification in single shot temporal action detection. In *2019 IEEE International Conference on Multimedia and Expo (ICME)*, 2019. 1, 5

[13] M. Jain, J. Van Gemert, H. Jégou, P. Bouthemy, and C. G. M. Snoek. Action localization with tubelets from motion. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 2014. 2

[14] Vicky Kalogeiton, Philippe Weinzaepfel, Vittorio Ferrari, and Cordelia Schmid. Action Tubelet Detector for Spatio-

Temporal Action Localization. In *ICCV*, pages 4415–4423, 2017. 1, 2, 5, 7

[15] Tianwei Lin, Xu Zhao, and Zheng Shou. Single shot temporal action detection. In *Proceedings of the 25th ACM International Conference on Multimedia*, 2017. 1, 4, 5

[16] Tianwei Lin, Xu Zhao, Haisheng Su, Chongjing Wang, and Ming Yang. Bsn: Boundary sensitive network for temporal action proposal generation. In *ECCV*, 2018. 1

[17] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott E. Reed, Cheng-Yang Fu, and Alexander C. Berg. Ssd: Single shot multibox detector. In *ECCV*, 2016. 4

[18] Fuchen Long, Ting Yao, Zhaofan Qiu, Xinmei Tian, Jiebo Luo, and Tao Mei. Gaussian temporal awareness networks for action localization. In *CVPR*, 2019. 5

[19] Santiago Manen, Matthieu Guillaumin, and Luc Van Gool. Prime object proposals with randomized prim's algorithm. In *ICCV*, 2013. 2

[20] Dan Oneata, Jerome Revaud, Jakob Verbeek, and Cordelia Schmid. Spatio-temporal object detection proposals. In *ECCV*, pages 737–752, 2014. 2

[21] Xiaojiang Peng and Cordelia Schmid. Multi-region two-stream R-CNN for action detection. In *ECCV*, volume 9908, pages 744–759, 2016. 2

[22] Xiaojiang Peng and Cordelia Schmid. Multi-region two-stream R-CNN for action detection. In *ECCV*, 2016. 7

[23] Rizard Renanda Adhi Pramono, Yie-Tarng Chen, and Wen-Hsien Fang. Hierarchical self-attention network for action localization in videos. In *ICCV*, 2019. 7

[24] Md Atiqur Rahman and Robert Laganière. Mid-level fusion for end-to-end temporal activity detection in untrimmed video. In *BMVC2020*, 2020. 1, 4, 5

[25] Joseph Redmon, Santosh Kumar Divvala, Ross B. Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *CVPR*, 2015. 4

[26] J. Redmon and A. Farhadi. Yolo9000: Better, faster, stronger. In *CVPR*, 2017. 4

[27] Mikel D. Rodriguez, Javed Ahmed, and Mubarak Shah. Action mach: a spatio-temporal maximum average correlation height filter for action recognition. In *ICCV*, 2008. 5

[28] Suman Saha, Gurkirt Singh, and Fabio Cuzzolin. Amtnet: Action-micro-tube regression by end-to-end trainable deep architecture. In *ICCV*, 2017. 2, 7

[29] Suman Saha, Gurkirt Singh, and Fabio Cuzzolin. Two-stream amtnet for action detection. *CoRR*, abs/2004.01494, 2020. 1

[30] Suman Saha, Gurkirt Singh, Michael Sapienza, H. S. Philip Torr, and Fabio Cuzzolin. Deep learning for detecting multiple space-time action tubes in videos. In *BMVC*, 2016. 2

[31] Suman Saha, Gurkirt Singh, Michael Sapienza, Philip Torr, and Fabio Cuzzolin. Deep learning for detecting multiple space-time action tubes in videos. In *BMVC*, 2016. 5, 7

[32] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 1*, 2014. 1

[33] Gurkirt Singh, Stephen Akrigg, Manuele Di Maio, Valentina Fontana, Reza Javanmard Alitappeh, Suman Saha, Kossar Jeddisaravi, Farzad Yousefi, Jacob Culley, Tom Nicholson, et al. Road: The road event awareness dataset for autonomous driving. *IEEE PAMI*, 2022. 7

[34] Gurkirt Singh, Suman Saha, Michael Sapienza, Philip H. S. Torr, and Fabio Cuzzolin. Online real-time multiple spatiotemporal action localisation and prediction. In *ICCV*, 2017. 2, 5, 7

[35] Lin Song, Shiwei Zhang, Gang Yu, and Hongbin Sun. Tacnet: Transition-aware context network for spatio-temporal action detection. In *CVPR*, 2019. 1, 2, 7

[36] K. Soomro, H. Idrees, and M. Shah. Action localization in videos through context walk. In *ICCV*, 2015. 2

[37] Khurram Soomro, Amir Roshan Zamir, Mubarak Shah, Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *CoRR*, 2012. 5

[38] D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun, and M. Paluri. A closer look at spatiotemporal convolutions for action recognition. In *CVPR*, 2018. 1

[39] J.R.R. Uijlings, K.E.A. van de Sande, T. Gevers, and A.W.M. Smeulders. Selective search for object recognition. *International Journal of Computer Vision*, 2013. 2

[40] Heng Wang, Alexander Kläser, Cordelia Schmid, and Liu Cheng-Lin. Action Recognition by Dense Trajectories. In *CVPR*, pages 3169–3176, 2011. 2

[41] Heng Wang and Cordelia Schmid. Action Recognition with Improved Trajectories. In *ICCV*, 2013. 2

[42] Philippe Weinzaepfel, Zaid Harchaoui, and Cordelia Schmid. Learning to track for spatio-temporal action localization. In *ICCV*, pages 3164–3172, 2015. 2, 5, 7

[43] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. Simple online and realtime tracking with a deep association metric. In *ICIP*, page 3645–3649, 2017. 4, 6

[44] Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin Murphy. Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In *ECCV*, 2018. 1, 3

[45] H. Xu, A. Das, and K. Saenko. R-c3d: Region convolutional 3d network for temporal activity detection. In *ICCV*, 2017. 1

[46] Huijuan Xu, Lizhi Yang, S. Sclaroff, Kate Saenko, and Trevor Darrell. Spatio-temporal action detection with multi-object interaction. *ArXiv*, abs/2004.00180, 2020. 1, 2, 7

[47] Xitong Yang, Xiaodong Yang, Ming-Yu Liu, Fanyi Xiao, Larry S. Davis, and Jan Kautz. Step: Spatio-temporal progressive learning for video action detection. In *CVPR*, 2019. 1, 2, 7

[48] Yuancheng Ye, Xiaodong Yang, and Yingli Tian. Discovering spatio-temporal action tubes. *J. Visual Communication and Image Representation*, 2019. 2

[49] Jiaojiao Zhao and Cees G. M. Snoek. Dance with flow: Two-in-one stream action detection. In *CVPR*, 2019. 7

[50] Jiyang Gao Zhenheng Yang and Ram Nevatia. Spatio-temporal action detection with cascade proposal and location anticipation. In *BMVC*, September 2017. 2