Three Birds with One Stone: Multi-Task Temporal Action Detection via Recycling Temporal Annotations

Zhihui Li Lina Yao

School of Computer Science and Engineering, University of New South Wales

zhihuilics@gmail.com, lina.yao@unsw.edu.au

Abstract

Temporal action detection on unconstrained videos has seen significant research progress in recent years. Deep learning has achieved enormous success in this direction. However, collecting large-scale temporal detection datasets to ensuring promising performance in the real-world is a laborious, impractical and time consuming process. Accordingly, we present a novel improved temporal action localization model that is better able to take advantage of limited labeled data available. Specifically, we design two auxiliary tasks by reconstructing the available label information and then facilitate the learning of the temporal action detection model. Each task generates their supervision signal by recycling the original annotations, and are jointly trained with the temporal action detection model in a multi-task learning fashion. Note that the proposed approach can be pluggable to any region proposal based temporal action detection models. We conduct extensive experiments on three benchmark datasets, namely THUMOS'14 [15], Charades [35] and ActivityNet [14]. Our experimental results confirm the effectiveness of the proposed model.

1. Introduction

Video collections have been proliferating with the advance of devices with video recording capabilities. Most of these videos are untrimmed and only a small part contains events of interest, while the major part is background. In continuous videos, temporal action detection (TAD) refers to the task of simultaneously recognizing actions and precisely localizing them in time. Due to its apparent complexity and enormous usefulness in real-world applications including video surveillance, video summarization and skill assessment, TAD has drawn attention from researchers in the machine learning and computer vision communities [41, 32]. When sufficient labeled training data exists, deep convolutional neural networks can achieve remarkable performance [18, 11, 48]. However, it is costly, time-consuming and tedious to acquire a large amount of segmentation-level label information in continuous videos for real-world applications.

Researchers have explored different ways to address the problem of labeled training data shortages for deep learning approaches. Among these approaches, multi-task learning (MTL) [25, 28, 4] is one the most representative examples. MTL mitigates the label shortage problem by training multiple relevant tasks at the same time [25, 17]. Its goal is to jointly train multiple relevant tasks with limited supervision information in order to improve the performance of each task [25]. Its effectiveness has been widely explored. As the number of related tasks increases, MTL is able to decrease the upper bound of the amount of labeled training data required, thereby allowing better generalization. MTL approaches can be broadly grouped into two categories. The first category aims to maximise task-wise performance by optimising the structures of weight sharing, while the second focuses on weight clustering based on task-similarity. Both of these approaches have been widely applied in the field of computer vision tasks including person re-identification [36], depth estimation and scene parsing [40, 44], *etc*. For example, we can employ Mask R-CNN [12] to improve the performance of object detection by jointly training an instance segmentation task. However, due to the expensive cost of segmentation mask labels, this approach is of limited practical benefit.

In this paper, we propose a novel temporal action detection framework that leverages the benefits of multi-task learning. More specifically, we build the proposed model from a widely used supervised temporal action detection framework [41], where a temporal proposal based detector is provided along with segmentation label information. Using the limited supervision information provided, we construct two auxiliary tasks (*e.g.* multi-action classification and localisation confidence estimation), which are used to improve the performance of temporal action detection in a multi-task learning fashion. These two tasks generate their

own supervision information by recycling the given limited temporal annotations. Different from the principles of traditional multi-task learning, we are here only concerned with the performance of the main temporal action detection task. We generate the ground truth information for these auxiliary tasks by exploring the temporal segmentation information provided, after which we jointly train the temporal action detection with these auxiliary tasks. To evaluate the performance of the proposed approach, we conduct extensive experiments on several publicly available benchmark datasets, including THUMOS'14 [15], Charades [35] and ActivityNet [14]. The experimental results confirm that auxiliary tasks contribute to the improvement of temporal action detection.

To summarize, we make the following contributions in this work:

- To mitigate the label shortage problem of temporal action detection, we propose a novel multi-task temporal action detection algorithm via reusing temporal annotations. The proposed approach can be applied to any region proposal based temporal action detection models.
- We construct two auxiliary tasks by recycling the temporal segmentation information, thereby improving the performance of temporal action localisation in a multi-task learning fashion.
- To demonstrate the performance improvement achieved by the proposed method, we conduct extensive experiments on three benchmark datasets: THUMOS'14 [15], Charades [35] and ActivityNet [14]. The experimental results confirm the effectiveness of the proposed method.

2. Related Works

Action Detection. There are two categories of action detection approaches in the literature: temporal-only action detection [24] and spatio-temporal action detection [30]. The spatio-temporal action detection algorithms propose to localize actions within spatio-temporal tubes, which require heavy human labor to produce sufficient fine-grained labels. Most existing works develop spatio-temporal action detection algorithms by tracking bounding boxes of action of interest across frames [33, 43]. Some researchers also try to explore dense trajectories for extracting the action tubes [27, 6, 9]. They first generate an initial still-image based segmentation of the video frames, and then prune and temporally extended them using optical flow and transductive learning. Then they run the detectors on the videos to extract the final tubes. The combination of appearance-based static information, motion information and transductive learning make their model robust for temporal action localization [5].

In contrast, the temporal-only action detection algorithms aim to detect the start and end times of the action of interest within long untrimmed video streams and classify the overall action [39, 49]. Pioneer works on temporal action detection mainly employ sliding windows to generate temporal segments, followed by classifying them with action classifiers trained with multiple features [16, 26, 33, 37]. Although these sliding window-based algorithms have achieved promising results, they are computationally inefficient. To overcome this limitation, researchers propose to model the temporal evolution of actions using RNNs or LSTM and predict an action label at each time step [24, 43], thus bypassing the requirement of exhaustive sliding window search. For example, Shou *et al*. develop a bottom-up action detection approach by label prediction at frame-level and fusion at video-level [31]. Xu *et al*. propose to encode the frames with fully-convolutional 3D filters, generate action proposals, followed by classifying and refining them based on pooled features within their boundaries [41].

In addition to supervised temporal action detection, some researchers focus on the problem of weakly supervised action localization [38], where only video-level labels are provided. They propose to learn attention weights on shot based or uniformly sampled proposals. In this paper, we focus on improving temporal action detection by recycling temporal boundary annotations in a multi-task learning fashion.

Multi-Task Learning (MTL). The MTL aims to jointly train multiple related tasks to mitigate the label shortage problem. Its usefulness has been widely demonstrated in the field of computer vision, including person re-identification [36], depth estimation and scene parsing [40], *etc*. According to how the parameters between various task models are shared, MTL can be grouped into two categories, namely hard parameter sharing and soft parameter sharing. For the hard parameter sharing methods, all the task models share the exact same feature extractor and execute corresponding task with its own branch head. The major challenge of this category is to design proper tasks and loss functions. Some representative works in this direction include Mask R-CNN [12], LASSO architecture [8], *etc*. For the soft parameter sharing methods, each task has its own model with its own parameters. Thus, the major challenge for these methods are how to develop weight sharing approaches. Some representative works in this direction include Sluice Networks [29], cross-stitch network [25], *etc*.

3. The Proposed Model

In this work, we address the problem of temporal action detection in a multi-task learning fashion. We assume that only temporal segments of the actions of interest are provided. To improve the performance of temporal action detection using multi-task learning, we design two pretext tasks (including multi-action classification and localisation confidence estimation), and train them at the same time as the temporal action detection task is performed. The pretext

tasks construct their supervision information by recycling the temporal segment annotations in a way that does not incur additional annotation costs. By exploring the temporal segment information provided, we can automatically generate supervision signals for these pretext tasks. Note that, unlike traditional multi-task learning, we only care about the performance of temporal action detection.

The main task should benefit from the designed pretext tasks in both feature extraction and prediction. Firstly, these pretext tasks are jointly trained with the main task, and learn the shared features that help temporal action detection. Secondly, the outputs of the multi-action classification task provides useful contextual information to refine the final temporal action detection result [3]. We will discuss the detailed refining procedure in Section 3.2.

The overall framework of the proposed approach is illustrated in Figure 1. As can be seen from the overall framework, we train the main temporal action detection task and pretext tasks simultaneously with a focus on improving the performance of temporal action detection. We will discuss the design of pretext tasks in Section 3.1, and the training details in Section 3.3.

3.1. Pretext Task Design

In this part, we discuss the two pretext tasks designed in this work, which are multi-action classification and localisation confidence estimation.

Multi-Action Classification: The label information of temporal action detection includes both the start and end time of action of interest and its corresponding action label. By recycling this label information, we construct the pretext task of *multi-action classification*. Rather than assigning a hard label for each action instance, we randomly pick a shot from the video and assign a *soft* label to it, which denotes the probabilities of several actions in this shot. In this way, we can generate a large number of positive training samples, although their label information is not as clean as these ground truth samples. This process shares a similar motivation to *mixup* [46], in which the model is trained on virtual samples constructed as the linear interpolation of pairs of random images and their labels. These approaches are able to address the imbalance problem that commonly exists in temporal action detection, *i.e.* cases where there are far fewer positive instances than negative instances per video.

Specifically, we first sample N_t temporal windows by randomly picking their starting point and temporal length. We apply the constraint that the temporal windows should maintain an intersection with one of the ground truth temporal segments in the video. Next, we compute a soft label l^m for each temporal window according to Algorithm 1, denoting l^m as the supervision label for the multi-action classification task. We assign a label probability for each temporal window by computing its area overlap ratio with the ground truth category. We then add an addition background category, resulting in a $K+1$ dimensional probability vector l^m 2 R^{K+1}.

Algorithm 1 Multi-Action Label Generation

Input: Video V, ground truth temporal segments $fS_i g_{i=1}^K$, temporal window T

Output: The multi-action soft label l^m for T

assign a $K+1$ dimensional array to l^m;

$l^m[0] = \sqrt{area(T) \quad area((\bigcup_{i2K} fS_i g))}$;

for *i = 1 to K* **do**

$l^m[i] = \sqrt{area(T \bigcap fS_i g)}$

Return: $l^m / \sum l^m$

Localisation Confidence Estimation: In this pretext task, we aim to consider the ambiguities of the ground truth segments. Based on two-stage temporal activity detection [41], we propose to regress the boundaries of each segment separately. Let (s, e) 2 R^2 be representation of the segmentation as a two-dimensional vector, where the dimensions are the starting and ending location of the segment. We here use the parameterizations of the (s, e) coordinates rather than those of the (c, l) coordinates used in [41]:

$$\delta s = (s \quad s)/l, \quad \delta e = (e \quad e)/l \tag{1}$$

$$\delta \hat{s} = (s \quad \hat{s})/l, \quad \delta \hat{e} = (e \quad \hat{e})/l \tag{2}$$

where s and e denote the starting and ending locations of the ground truth activity segments, while s and e represent the starting and ending locations of the predicted anchor segments or proposals. We further denote $t = f\delta\hat{s}, \delta\hat{e}g$ as the predicted relative offset to the anchor segments or proposals, and $t = f\delta s, \delta e g$ as the coordinate transformation of the ground truth segments to anchor segments or proposals. In the two-stage temporal activity detection framework, the regression term optimizes the relative displacement between proposals and ground truths.

This pretext task predicts a probability distribution for confidence estimation. Based on the assumption that the coordinates are independent, we employ single variate gaussian for simplicity:

$$P_\Theta(x) = p\frac{1}{2\pi\sigma^2} e^{\frac{(x-x_e)^2}{2\sigma^2}} \tag{3}$$

where Θ indicates the learnable parameters of the network, while x_e is the estimated segmentation location. In Equation 3, the standard deviation σ measures the uncertainty of the estimation. When the network is extremely confident about the estimated segment location, σ ! 0. Similarly, we also formulate the ground-truth segment as a Gaussian

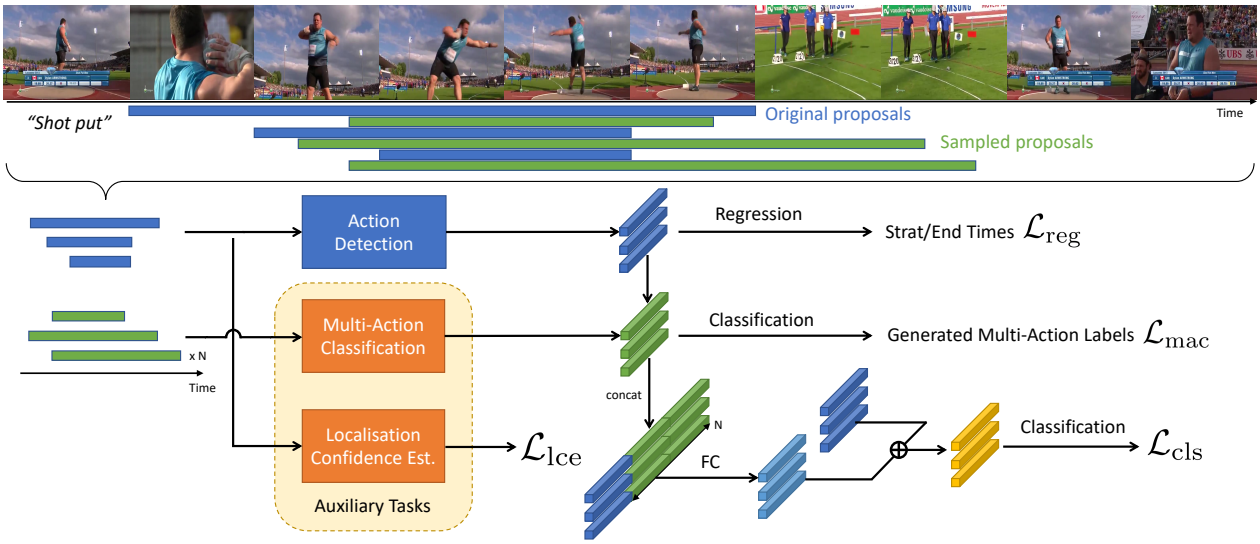

Figure 1: Overall architecture of the proposed approach for temporal action detection. We train the main task of temporal action detection and two pretext tasks (multi-action classification and localisation confidence estimation) at the same time. These pretext tasks are beneficial to the main tasks in both feature extraction and prediction. Firstly, they are jointly trained with temporal action detection task, which facilitates shared feature learning for the main task. Secondly, the outputs of the pretext task, multi-action classification, can provide useful contextual information for the main task (details can be found in Section 3.2).

distribution:

$$P_D(x) = \delta(x \quad x_g) \tag{4}$$

where x_g denotes the location of the ground truth segment.

Intuitively, the auxiliary tasks should benefit both the feature learning and the prediction tasks. Thus, we use the outputs of the auxiliary tasks to refine the detection prediction. More specifically, we concatenate the outputs of the original detection and multi-action classification, and feed the result into a fully connected layer with a residual connection.

3.2. Refining

We argue that the pretext tasks should boost performance of the main task in terms of both feature extraction and prediction. In the first stage of region proposal learning, the pretext tasks are firstly jointly trained with the main task to learn shared features that facilitate temporal action detection. In the prediction stage, the outputs of the pretext task should help refine the prediction results of the detection. For example, classification of the region proposals can provide contextual information to refine the prediction results of the detection. In this part, we will discuss how we refine the detection results in the prediction stage.

As discussed in Section 3.1, the multi-action classification module can predict soft labels for a given temporal proposal and windows close to it. Our intuition is that the soft labels generated by the multi-action classification module can provide useful contextual information for the temporal action detector to make better decision for classification. Thus, we require the temporal action detector to take advantage of the predictions of multi-action classification module. In the inference stage, given a temporal proposal, the multi-action classification module generates soft label prediction for the local and global context close to the proposal. Empirically, we do not use the prediction result of the localisation confidence estimation task, as we did not find improvement in the experiments.

With a traditional temporal action detector (*e.g.* R-C3D [41]), the detector head predicts a classification result $\mathbf{x}\ 2\ \mathsf{R}^{K+1}$ for a given temporal proposal, and then achieves a class probability $\mathbf{y}$. The refining procedure will convert $\mathbf{x}$ into $\mathbf{x}^0$ with the outputs from the pretext tasks as follows. Firstly, we generate N_t temporal windows close to the proposal with different sizes. Empirically, we set N_t as 5 in the experiments. Thus, we get multi-action soft labels for these N_t temporal windows, denoted by $f\mathbf{t}_1, \mathbf{t}_2, \quad , \mathbf{t}_{N_t}g$. Then we get the refined $\mathbf{x}^0$ by:

$$\mathbf{x}^0 = \mathbf{W}[\mathbf{x}, \mathbf{t}_1, \quad , \mathbf{t}_N] + \mathbf{x}, \tag{5}$$

where $\mathbf{W}$ is a projection matrix, $[\mathbf{x}, \mathbf{t}_1, \quad , \mathbf{t}_N]$ denotes that we concatenate $\mathbf{x}$ and $\mathbf{t}_1, \quad , \mathbf{t}_N$, and pass it into a fully connected layer with a residual connection [13].

3.3. Training

Loss functions. In terms of the loss of multi-action classification, we define its loss as a cross-entropy loss, since it can be formulated as a prediction of class labels:

$$L_{\text{mac}} = \frac{1}{N_t}\sum_{j=1}^{N_t} {y_j}^T \log(\text{softmax}(\mathbf{t}_j)), \quad (6)$$

where N_t is the number of temporal segments, y_j is the ground truth soft label for the j-th temporal segment, and softmax($\mathbf{t}_j$) achieves its corresponding predicted category probability.

For the loss of localization confidence estimation, we aim to measure the KL-Divergence between $P_\Theta(x)$ and $P_D(x)$ over N samples as follows:

$$L_{\text{lce}} = \frac{1}{N} D_{\text{KL}}(P_D(x) || P_\Theta(x)) \quad (7)$$

For the loss of the main task, we follow R-C3D [41] to use softmax loss function L_{cls} for proposal binary classification and smooth L1 loss function L_{reg} for regression. Thus, the loss of the main task is as follows:

$$L_{\text{main}} = L_{\text{cls}} + \lambda L_{\text{reg}} \quad (8)$$

Finally, the overall loss L_{overall} is the weighted sum of the main task loss L_{main}, the multi-action classification loss L_{mac} and the localization confidence estimation loss L_{lce}, as follows:

$$L_{\text{overall}} = L_{\text{main}} + \lambda_{\text{mac}} L_{\text{mac}} + \lambda_{\text{lce}} L_{\text{lce}} \quad (9)$$

In the experiments, we empirically set $\lambda = 1$, $\lambda_{\text{mac}} = 1$ and $\lambda_{\text{lce}} = 0.7$.

Implementation. We simultaneously train the entire network using the ground truths of both the main task and the pretext tasks. Our implementation of the model in this paper is based on [41]. Following their work, we initialize the backbone network for each dataset separately. More specifically, for THUMOS'14, we initialize the backbone network by pretraining on Sports-1M and finetuning on UCF101. For Charades, we initialize the backbone network by finetuning the Sports-1M pretrained model on the Charades training set. Finally, for AcitivityNet, we initialize the backbone network via pretraining on Sports-1M, and finetuning on the training videos of ActivityNet.

4. Experiments

We test the performance of the proposed model on three benchmark datasets: THUMOS'14 [15], Charades [35] and ActivityNet [14]. We present and discuss the experimental results on these three datasets in Sections 4.1, 4.3 and 4.2, respectively.

Table 1: We report the performance of temporal action localization on THUMOS'14 in percentages. Mean average precision (mAP) at different IoU thresholds α are used as evaluation metrics. Best performance is highlighted in bold. Baseline is R-C3D [41].

	α				
	0.1	0.2	0.3	0.4	0.5
Baseline	54.5	51.5	44.8	35.6	28.9
+ Task 1	57.3	53.7	46.1	37.2	31.1
+ Task 2	56.2	52.8	45.3	36.3	30.5
+ Task 1,2	**57.7**	**54.3**	**47.6**	**38.5**	**31.9**

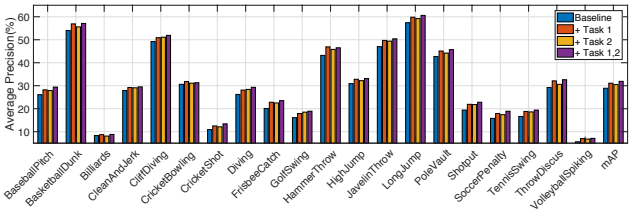

Figure 2: Detailed performance comparison of per-class AP at IoU threshold $\alpha = 0.5$ on THUMOS'14. The performance is reported in percentages. Baseline is R-C3D [41].

4.1. Experiments on THUMOS'14

Dataset Description. The THUMOS'14 dataset consists of more than 24 hours of videos featuring more than 20 different sports actions. It contains 2765 training trimmed videos, 200 untrimmed validation videos and 213 untrimmed testing videos. Temporal action localization with this dataset is extremely difficult because while each video can be as long as a few hundreds of seconds, each action instance can be as short as a few tens of seconds.

Experimental Setup. Following [41], we split the 200 untrimmed validation videos into 180 for training and the remaining 20 for hyperparameter tuning. We use all the 200 videos for training, and report the final results on 213 testing videos. We initialize the backbone (3D ConvNet part of the proposed model) with C3D weights pretrained on the Sports-1M dataset and finetuned on the UCF101 dataset. We train the overall framework on THUMOS'14 with a fixed learning rate of 0.0001.

Results. We show the performance improvement of the proposed approach over the baseline on THUMOS'14 in Table 1. The R-C3D [41] is employed as the baseline. We use mAP at IoU thresholds 0.1-0.5 (denoted as α) as the evaluation metric. To be more specific, we show the detailed performance in Figure 2. That is, we show the detailed detection performance over all the 20 action categories of THUMOS'14 in Figure 2. The experimental results reported in Table 1 and Figure 2 confirm that the designed pretext tasks consistently boost the performance of main task. For example, the performance was improved from 28.9% to 31.9% at

Table 2: We report performance of temporal action localization on ActivityNet v1.3 in percentages. Mean average precision (mAP) at different IoU thresholds α are used as evaluation metrics. Best performance is highlighted in bold. Baseline is R-C3D [41].

	α			Average
	0.5	0.75	0.95	
Baseline	26.8	10.8	0.5	12.7
+ Task 1	29.2	12.6	**0.7**	14.2
+ Task 2	28.3	11.5	0.6	13.5
+ Task 1,2	**29.3**	**13.4**	**0.7**	**14.5**

Table 3: We report the temporal action localization performance on Charades in percentages. Mean average precision (mAP) is used as evaluation metrics. Best performance is highlighted in bold. Baseline is R-C3D [41].

	mAP	
	standard	post-process
baseline	12.4	12.7
+ Task 1	14.3	14.8
+ Task 2	14.1	14.4
+ Task 1,2	**14.6**	**14.9**

IoU threshold of 0.5 (that is 10.4% relative improvement).

From these experimental results, we have the following observations. Firstly, it is very encouraging that the proposed approach is constantly effective on all all the action categories in THUMOS'14 (shown in Figure 2), regardless of the complexity of the actions. Secondly, pretext task 1 (multi-action classification) is more helpful for temporal action detection than pretext task 2 (localisation confidence estimation). This is because Task 1 generates a bunch of temporal windows, which provides useful contextual information for the main task. Thirdly, when we train task 1 and task 2 jointly, we get the best improvement, compared to training with each task alone.

4.2. Experiments on ActivityNet

Dataset Description. The ActivityNet 1.3 dataset contains 100,244,926 and 5,044 videos, which are split into 200 different categories of activities in the train, validation and test sets respectively. Most of the videos in this dataset have activity instance of a single class. This is a much larger dataset than THUMOS'14 in terms of activity category number and the amount of videos. Since the ground truth annotations are not public, we evaluate the proposed approach on the validation set.

Experimental Setup. Follwoing [41], we sample frames at 3 fps to fit it in the GPU memory. Also, we set the number of anchor segments K to 20. As there is vast domain difference between Sports-1M and ActivityNet, we pretrain 3D ConvNet model on Sports-1M, and finetune with the training videos of AcvitivityNet. Then we initialize the 3D ConvNet with these finetuned weights. To improve efficiency, we freeze the first two convlutional layers in our model during training. We fix the learning rate at 10^{-4} for first 10 epochs and decrease it to 10^{-5} for the last 5 epochs.

Results. We report the performance improvement of the propsed approach on ActivityNet v1.3 in Table 2. We use mAP at IoU thresholds f0.5, 0.75, 0.95g and average result as evaluation metrics. The experimental results shown in Table 2 demonstrate that the proposed approach non-trivially increase the detection performance at all IoU thresholds. For example, the performance of temporal action detection is improved from 10.8 to 13.4 when the IoU is 0.75 (that is 24.1% relative improvement).

When we take a close look at the results in Table 2, we have similar observations as on the THUMOS'14 dataset. Pretext task 1 is also more useful than pretext task 2 for temporal action detection, mainly because a lot of temporal segments generated by pretext task 1 provide useful contextual information, thus boost the performance of the main task. Also, when we jointly train the pretext tasks with the main task, we achieve the biggest improvement at all evaluation metrics.

We show some qualitative examples of temporal action detection results of the proposed model on the THUMOS'14 dataset in Figure 3. In each example, we can show the ground truth (upper) and the results of our model (lower). From these examples, we can see that the proposed model can achieve very promising results on the benchmark datasets.

4.3. Experiments on Charades

Dataset Description. The Charades dataset [35] is proposed for simultaneous action detection and classification. It contains videos recorded from daily life activities, which are grouped into 157 categories. In the process of dataset collection, Amazon Mechanical Turk (AMT) users are employed to record videos based on video scripts. This dataset is very challenging because of low illumination, diversity and casual nature of the videos containing daily activities, as well as the abundance of overlapping activities. In this dataset, some activities may even have exactly the same start and end times. For example, the activities "holding a towel" and "tidying up a towel" have exactly the same segments in the original videos.

Experimental Setup. Following [41], we sample frames at 5 fps and set the input buffer to contain 768 frames. We set the number of anchor segments K to be 18 with specific scale values [1,2,3,4,5,6,7,8,10,12,14,16,20,24,28,32,40,48]. Similarly, we first pre-train the C3D model with the Sports-1M dataset, and finetune on the training set of the Charades dataset. Then we initialise the 3D ConvNet part of our

Figure 3: Qualitative visualization of the predicted activities by the proposed approach. (best viewed in color). Corresponding start-end times and confidence score are shown under the video frames.

model with the finetuned weights. After that, we fix the first two convolutional layers to accelerate training. While training, we keep the learning rate fixed at 10^{-3} for the first 10 epochs and then decrease it to 10^{-4} for the remaining 5 epochs.

Results. We present the performance gain of the proposed approach on the Charades dataset in Table 3. We map the activity segment prediction to 25 equidistant frames and use mean average precision (mAP) as an evaluation metric. To make a fair comparison, we follow [41] to post-process and average the frame level predictions across 20 frames, thus spatial consistency is improved. As can seen from the results shown in Table 3 that the proposed model consistently improves the performance of temporal action detection on the Charades dataset. For example, without post-processing, the performance is improved from 12.4 to 14.6 (which is 17.7% relative improvement).

A significant challenge of this dataset is that there are a great number of temporally overlapping activities in the videos. From the experimental results we can see that even under such complex scenarios, the proposed model can still improve the performance of the main task. That means the temporal segments generated by the multi-action classification task can still provide useful contextual information for the main task under such scenarios. We can also observe that, although the improvement of pretext task 2 is smaller than pretext task 1, we can achieve the largest achievements when we train both pretext tasks with the main task. This phenomenon is consistent over the three datasets used in this paper.

4.4. Comparison with State-of-the-Art

In this section, we compare the proposed approach with several state-of-the-art techniques on the three benchmark datasets. For our approach, we use R-C3D [41] and PBR-Net [21] as baseline models, as the proposed approach can be applied to any region proposal based temporal action detection models. Table 4 compares the action detection results of the proposed approach and various state-of-the-art methods on the THUMOS'14 dataset. From the experimen-

tal results in Table 4, we can observe that the proposed approach consistently achieve the best performance across all the thresholds. For example, at IoU 0.5, the proposed approach with PBRNet as a baseline model reaches a mAP of 54.8%, which is obviously better than its baseline model, which is the second best model. This confirms the advantages of recycling the segementation annotations.

Table 5 compares the proposed approach with state-of-the-art detectors. We report the performance at different tIoU thresholds in terms of mAP, as well as average mAP. The proposed approach reports the best average mAP results on this large-scale and diverse dataset. Notably, the proposed approach achieves a mAP of 9.6% at IoU 0.95, demonstrating that the localization performance of the proposed approach is much better than others. In addition, we compare the proposed approach against related approaches on Charades in Table 6. The experimental results reported in Table 6 confirms the effectiveness of the proposed approach in improving detection performance via recycling the segmentation annotations.

Table 4: Comparison against state-of-the-art on THUMOS'14, measured by mAP (%) at different tIoU thresholds.

	α				
	0.3	0.4	0.5	0.6	0.7
R-C3D [41]	44.8	35.6	28.9	20.2	14.5
SS-TAD [2]	45.7	-	29.2	-	9.6
SSN [47]	51.9	41.0	29.8	-	-
CBR [10]	50.1	41.3	31.0	19.1	9.9
BSN [20]	53.5	45.0	36.9	28.4	20.0
MGG [22]	53.9	46.8	37.4	29.5	21.3
GTAN [23]	57.8	47.2	38.8	-	-
BMN [19]	56.0	47.4	38.8	29.7	20.5
CMS-RC3D [1]	54.7	48.2	40.0	-	-
TAL-Net [7]	53.2	48.5	42.8	33.8	20.8
PBRNet [21]	58.5	54.6	51.3	41.8	29.5
G-TAD [42]	54.5	47.6	40.2	30.8	23.4
Ours (R-C3D as baseline)	47.6	38.5	31.9	23.7	18.4
Ours (PBRNet as baseline)	**63.2**	**58.5**	**54.8**	**44.3**	**32.4**

Table 5: Comparison against state-of-the-art on ActivityNet, measured by mAP (%) at different tIoU thresholds and avearge mAP.

	α			Average
	0.5	0.75	0.95	
SCC [14]	40.0	17.9	4.7	21.7
R-C3D [41]	26.8	10.8	0.5	12.7
CDC [31]	45.3	26.0	0.2	23.8
BSN [20]	46.5	30.0	8.0	30.0
Chao *et al.* [7]	38.2	18.3	1.3	20.2
P-GCN [45]	48.3	33.2	3.3	31.1
BMN [19]	50.1	34.8	8.3	33.9
PBRNet [21]	54.0	35.0	9.0	32.7
G-TAD [42]	50.4	34.6	9.0	34.1
Ours (R-C3D as baseline)	29.3	13.4	0.7	14.5
Ours (PBRNet as baseline)	**57.8**	**37.6**	**9.6**	**35.0**

Table 6: Comparison against state-of-the-art on Charades in percentages.

	mAP	
	standard	post-process
Two-Steam [34]	7.7	10.0
Two-Stream+LSTM [34]	8.3	8.8
Sigurdsson *et al.* [34]	9.6	12.1
R-C3D [41]	12.4	12.7
PBRNet [21]	21.5	22.1
Ours (R-C3D as baseline)	14.6	14.9
Ours (PBRNet as baseline)	**23.2**	**23.6**

4.5. Ablation Study

In this part, we conduct an ablation study on the effect of refining. For traditional MTL algorithms, we do not directly use the outputs of pretext tasks to refine the results of the main task. Different from tradtional MTL algorithms, our pretext tasks is capable of boosting the performance of the main task, because they provide useful contextual information about the neighbors of segments of interest. We present the improvement achieved by refining in Table 7. From the results shown in Table 7, we observe that the refinement consistently improves the performance of the main task. To step further, we also employ the *stop_gradient* to test the performance of using refining alone. From the experimental results, we confirm that both MTL and refining contribute to the improvement of the proposed model.

Table 7: Ablation study of MTL and refining on the three benchmark datasets. Baseline is R-C3D [41].

	THUMOS'14		ActivityNet	Charades	
	0.4	0.5	Average	stand	post-process
baseline	35.6	28.9	12.7	12.4	12.7
+ MTL	37.9	30.2	13.8	13.5	13.9
+ Refine	37.5	29.8	13.6	13.9	14.1
+ Both	**38.5**	**31.9**	**14.5**	**14.6**	**14.9**

5. Conclusion

In this paper, we have proposed a novel multi-task learning paradigm for temporal action detection. Our model aims to simultaneously predict the action label and detect the start and end times of each action instance from untrimmed videos. We have designed two auxiliary tasks, namely multi-action classification task and localisation confidence estimation task, to improve the temporal action detection performance in a multi-task learning fashion. These two auxiliary tasks generate their supervision information by recycling the label information of TAD. We have conducted extensive experiments to confirm the effectiveness of the proposed model, as well as an ablation study that confirms the usefulness of each auxiliary task. In the future, we plan to recycle other labels, such as video object segmentation, to further improve the temporal action detection performance.

Acknowledgement

This research was partially supported by grant ONRG NICOP N62909- 19-1-2009, and National Natural Science Foundation of China under grant no. 61906109.

References

[1] Yancheng Bai, Huijuan Xu, Kate Saenko, and Bernard Ghanem. Contextual multi-scale region convolutional 3d network for activity detection. *CoRR*, abs/1801.09184, 2018.

[2] 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.

[3] Xiaojun Chang, Po-Yao Huang, Yi-Dong Shen, Xiaodan Liang, Yi Yang, and Alexander G. Hauptmann. RCAA: relational context-aware agents for person search. In *ECCV*, 2018.

[4] Xiaojun Chang, Haoquan Shen, Sen Wang, Jiajun Liu, and Xue Li. Semi-supervised feature analysis for multimedia annotation by mining label correlation. In *PAKDD*, pages 74–85, 2014.

[5] Xiaojun Chang, Yaoliang Yu, Yi Yang, and Eric P. Xing. They are not equally reliable: Semantic event search using differentiated concept classifiers. In *CVPR*, 2016.

[6] Xiaojun Chang, Yaoliang Yu, Yi Yang, and Eric P. Xing. Semantic pooling for complex event analysis in untrimmed videos. *IEEE Trans. Pattern Anal. Mach. Intell.*, 39(8):1617–1632, 2017.

[7] Yu-Wei Chao, Sudheendra Vijayanarasimhan, Bryan Seybold, David A. Ross, Jia Deng, and Rahul Sukthankar. Rethinking the faster R-CNN architecture for temporal action localization. In *CVPR*, 2018.

[8] Carl Doersch and Andrew Zisserman. Multi-task self-supervised visual learning. In *ICCV*, 2017.

[9] Hehe Fan, Xiaojun Chang, De Cheng, Yi Yang, Dong Xu, and Alexander G. Hauptmann. Complex event detection by identifying reliable shots from untrimmed videos. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pages 736–744, 2017.

[10] Jiyang Gao, Zhenheng Yang, and Ram Nevatia. Cascaded boundary regression for temporal action detection. In *BMVC*, 2017.

[11] Rohit Girdhar, João Carreira, Carl Doersch, and Andrew Zisserman. Video action transformer network. In *CVPR*, 2019.

[12] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. Mask R-CNN. In *ICCV*, 2017.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[14] Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *CVPR*, 2015.

[15] Y.-G. Jiang, J. Liu, A. Roshan Zamir, G. Toderici, I. Laptev, M. Shah, and R. Sukthankar. THUMOS challenge: Action recognition with a large number of classes. http://crcv.ucf.edu/THUMOS14/, 2014.

[16] Svebor Karaman, Lorenzo Seidenari, and Alberto Del Bimbo. Fast saliency based pooling of fisher encoded dense trajectories. In *ECCV Workshop*, 2014.

[17] Wonhee Lee, Joonil Na, and Gunhee Kim. Multi-task self-supervised object detection via recycling of bounding box annotations. In *CVPR*, 2019.

[18] Peng Lei and Sinisa Todorovic. Temporal deformable residual networks for action segmentation in videos. In *CVPR*, 2018.

[19] Tianwei Lin, Xiao Liu, Xin Li, Errui Ding, and Shilei Wen. BMN: Boundary-Matching Network for Temporal Action Proposal Generation. In *ICCV*, 2019.

[20] Tianwei Lin, Xu Zhao, Haisheng Su, Chongjing Wang, and Ming Yang. BSN: Boundary Sensitive Network for Temporal Action Proposal Generation. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *ECCV*, 2018.

[21] Qinying Liu and Zilei Wang. Progressive boundary refinement network for temporal action detection. In *AAAI*, 2020.

[22] Yuan Liu, Lin Ma, Yifeng Zhang, Wei Liu, and Shih-Fu Chang. Multi-granularity generator for temporal action proposal. In *CVPR*, 2019.

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

[24] Shugao Ma, Leonid Sigal, and Stan Sclaroff. Learning activity progression in lstms for activity detection and early detection. In *CVPR*, 2016.

[25] Ishan Misra, Abhinav Shrivastava, Abhinav Gupta, and Martial Hebert. Cross-stitch networks for multi-task learning. In *CVPR*, 2016.

[26] Dan Oneata, Jakob Verbeek, and Cordelia Schmid. The lear submission at thumos 2014. In *ECCV Workshop*, 2014.

[27] Mihai Marian Puscas, Enver Sangineto, Dubravko Culibrk, and Nicu Sebe. Unsupervised tube extraction using transductive learning and dense trajectories. In *ICCV*, 2015.

[28] Rajeev Ranjan, Vishal M. Patel, and Rama Chellappa. Hyperface: A deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 41(1):121–135, 2019.

[29] Sebastian Ruder, Joachim Bingel, Isabelle Augenstein, and Anders Søgaard. Sluice networks: Learning what to share between loosely related tasks. *CoRR*, abs/1705.08142, 2017.

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

[31] Zheng Shou, Jonathan Chan, Alireza Zareian, Kazuyuki Miyazawa, and Shih-Fu Chang. CDC: convolutional-de-convolutional networks for precise temporal action localization in untrimmed videos. In *CVPR*, 2017.

[32] Zheng Shou, Hang Gao, Lei Zhang, Kazuyuki Miyazawa, and Shih-Fu Chang. Autoloc: Weakly-supervised temporal action localization in untrimmed videos. In *ECCV*, 2018.

[33] Zheng Shou, Dongang Wang, and Shih-Fu Chang. Temporal action localization in untrimmed videos via multi-stage cnns. In *CVPR*, 2016.

[34] Gunnar A. Sigurdsson, Santosh Kumar Divvala, Ali Farhadi, and Abhinav Gupta. Asynchronous temporal fields for action recognition. In *CVPR*, 2017.

[35] Gunnar A. Sigurdsson, Gül Varol, Xiaolong Wang, Ali Farhadi, Ivan Laptev, and Abhinav Gupta. Hollywood in Homes: Crowdsourcing Data Collection for Activity Understanding. In *ECCV*, 2016.

[36] Chi Su, Fan Yang, Shiliang Zhang, Qi Tian, Larry S. Davis, and Wen Gao. Multi-task learning with low rank attribute embedding for person re-identification. In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*, pages 3739–3747, 2015.

[37] Limin Wang, Yu Qiao, and Xiaoou Tang. Action recognition and detection by combining motion and appearance features. In *ECCV Workshop*, 2014.

[38] Limin Wang, Yuanjun Xiong, Dahua Lin, and Luc Van Gool. Untrimmednets for weakly supervised action recognition and detection. In *CVPR*, 2017.

[39] Philippe Weinzaepfel, Zaïd Harchaoui, and Cordelia Schmid. Learning to track for spatio-temporal action localization. In *ICCV*, 2015.

[40] Dan Xu, Wanli Ouyang, Xiaogang Wang, and Nicu Sebe. Pad-net: Multi-tasks guided prediction-and-distillation network for simultaneous depth estimation and scene parsing. In *CVPR*, 2018.

[41] Huijuan Xu, Abir Das, and Kate Saenko. R-C3D: region convolutional 3d network for temporal activity detection. In *ICCV*, 2017.

[42] Mengmeng Xu, Chen Zhao, David S. Rojas, Ali K. Thabet, and Bernard Ghanem. G-TAD: sub-graph localization for temporal action detection. In *CVPR*, 2020.

[43] Serena Yeung, Olga Russakovsky, Greg Mori, and Li Fei-Fei. End-to-end learning of action detection from frame glimpses in videos. In *CVPR*, 2016.

[44] Di Yuan, Xiaojun Chang, Po-Yao Huang, Qiao Liu, and Zhenyu He. Self-supervised deep correlation tracking. *IEEE Trans. Image Process.*, 30:976–985, 2021.

[45] Runhao Zeng, Wenbing Huang, Chuang Gan, Mingkui Tan, Yu Rong, Peilin Zhao, and Junzhou Huang. Graph convolutional networks for temporal action localization. In *ICCV*, 2019.

[46] Hongyi Zhang, Moustapha Cissé, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *ICLR*, 2018.

[47] Yue Zhao, Yuanjun Xiong, Limin Wang, Zhirong Wu, Xiaoou Tang, and Dahua Lin. Temporal action detection with structured segment networks. In *ICCV*, 2017.

[48] Fengda Zhu, Yi Zhu, Xiaojun Chang, and Xiaodan Liang. Vision-language navigation with self-supervised auxiliary reasoning tasks. In *CVPR*, 2020.

[49] Hongyuan Zhu, Romain Vial, and Shijian Lu. TORNADO: A spatio-temporal convolutional regression network for video action proposal. In *ICCV*, 2017.