IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received January 9, 2022, accepted March 9, 2022, date of publication April 4, 2022, date of current version April 13, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3164730

TEDdet: Temporal Feature Exchange and Difference Network for Online Real-Time Action Detection

YU LIU, FAN YANG, AND DOMINIQUE GINHAC
ImViA EA7535, University Burgundy Franche-Comté, 21078 Dijon, France

Corresponding author: Yu Liu (yu_liu@etu.u-bourgogne.fr)

This work was supported by the European Horizon 2020 Innovative Training Network (H2020-MSCA-ITN-2017) under Grant 765866.

ABSTRACT Localizing and interpreting human actions in videos require understanding the spatial and temporal context of the scenes. Aside from accurate detection, vast sensing scenarios in the real-world also mandate incremental, instantaneous processing of scenes under restricted computational budgets. However, state-of-the-art detectors fail to meet the above criteria. The main challenge lies in their heavy architectural designs and detection pipelines to reason pertinent spatiotemporal information, such as incorporating 3D Convolutoinal Neural Networks (CNN) or extracting optical flow. With this insight, we propose a lightweight action tubelet detector coined TEDdet which unifies complementary feature aggregation and motion modeling modules. Specifically, our Temporal Feature Exchange module induces feature interaction by adaptively aggregating 2D CNN features over successive frames. To address actors' location shift in the sequence, our Temporal Feature Difference module accumulates approximated pair-wise motion among target frames as trajectory cues. These modules can be easily integrated with an existing anchor-free detector to cooperatively model action instances' categories, sizes, and movement for precise tubelet generation. TEDdet exploits larger temporal strides to efficiently infer actions in a coarse-to-fine and online manner. Without relying on 3D CNN or optical flow models, our detector demonstrates competitive accuracy at an unprecedented speed (89 FPS) that is more compliant with realistic applications. Codes will be available at https://github.com/alphadadajuju/TEDdet.

INDEX TERMS Action detection, action localization, convolutional neural network, deep learning, real-time computation, video understanding.

I. INTRODUCTION

In recent years, human action detection has become an active area of research driven by numerous vision applications and industries such as autonomous driving, security monitoring, transport and human-computer interaction systems, etc. The task (also referred to as action localization) concerns creating spatiotemporal action proposals to locate individual actors in space and time from a video, as well as classifying their undergoing action categories [1]. Inherently, action detection imposes more challenges in general when compared to action recognition which seeks only the global label of the video. Its complexity is further extended when detection is required to take place in real-time and incremental manner, which are crucial criteria in a host of practical scenarios coping with online video streams.

Following the success of Convolutional Neural Network (CNN) throughout various computer vision applications, recent works in action detection mainly adopt CNN-based detectors [2]–[4] to localize action instances, either at the frame-level [5]–[7] or in the form of tubelets (i.e., a sequence of bounding boxes over consecutive frames) [8]–[11]. Alongside spatial information, effective temporal modeling plays an imperative role for identifying actions. Simonyan *et al.* [12] pioneered the two-stream CNN framework, extracting RGB and optical flow features from two separate CNN backbones and then fusing the complementary information. This line of method achieves state-of-the-art performance and drives many subsequent research in the field of action recognition/detection. Another

The associate editor coordinating the review of this manuscript and approving it for publication was Byung-Gyu Kim.

approach to capture spatiotemporal information is to employ 3D CNN [13], [14]. Hierarchically stacking 3D convolutional filters enables simultaneous modeling of spatial and temporal variations over consecutive frames. In this fashion, highly abstracted video representation specific to different actions can be learned and inferred directly from RGB images [15]–[19].

In spite of the aforementioned advancements, existing methods are often challenged to be deployed in real-world applications with limited computing budget. Firstly, acquiring optical flow for two-stream CNN is time-consuming and computationally expensive. Hence in most existing methods, optical flow is pre-computed and stored on disk, inevitably excluding two-stream CNN from online tasks or those with real-time specifications. Alternatively, 3D CNN is capable of encoding video representations directly from RGB sequences. Nevertheless, common practices with 3D CNN process 8 or 16 frames in parallel as 3D convolutions favor dense sampling in time to effectively model spatiotemporal evolution [20]. This workflow is sub-optimal for online videos in terms of efficiency, implying that every attempt to detect potential actions on a few incoming frames requires feed-forwarding a complete sequence to the 3D CNN model. At the same time, 3D CNN introduces more parameters, computational cost and training difficulty than its 2D counterpart by an order of magnitude. Despite recent emergence of lightweight architectures [21]–[26], these models have been mainly experimented for offline action recognition and not yet explored for the detection task.

Aiming to meet the criteria of realistic applications, we propose the **T**emporal feature **E**xchange and **D**ifference action **det**ector (termed TEDdet), a unified action tubelet detector on top of 2D CNN. Specifically, TEDdet's temporal modeling makes use of our Temporal Feature Exchange (TE) and Temporal Feature Difference (TD) modules. The TE module performs partial feature exchange among neighboring frames, adaptively aggregating contextual information from adjacent frames to the target one on which actions will be detected. While TE helps to induce interaction among successive frames, TD approximates relative motion based on feature-level displacement to facilitate tracking actors' location shift over time.

We integrate the two complementary modules with the anchor-free CenterNet detector [27]. Given an input video sequence, TEDdet leverages the spatiotemporal information aggregated by TE to categorize and localize action instances by their centers on the target frame. To model movement of actions throughout the sequence, we exploit stacked pair-wise motions produced by TD to estimate the global trajectory and refine locations of action centers over the entire input sequence. Finally, the bounding boxes of each action instance is deduced at the predicted action centers. The above sub-tasks take place cooperatively at three respective detector branches. Combining a coarse-detection scheme and an online tube generation algorithm, TEDet is capable of proposing long-range action tubes from online video streams, both at high localization precision and in real-time.

To the best of our knowledge, TEDdet has one of the most compact architecture for action detection while equipping spatiotemporal and motion modeling capability. The main contributions of our work can be summarized as follows:

- We present two lightweight temporal modeling modules: Temporal Feature Exchange (TE) and Temporal Feature Difference (TD) to facilitate learning action-specific spatiotemporal pattern and trajectory.
- We propose TEDdet, an integrated action tubelet detector on top of 2D CenterNet and TE-TD plug-in. Our detector operates in a coarse-to-fine manner. Alongside the online tube generation algorithm, TEDdet's detection speed well exceeds real-time requirement (89 FPS).
- Comprehensive analysis in terms of TEDdet's accuracy, robustness, and efficiency are conducted on public UCF-24 and JHMDB-21 datasets.

II. RELATED WORK

Recent approaches to address action detection largely build upon the techniques from CNN-based object detection and action recognition. In this section, we briefly review these relevant topics.

Mainstream **Object detection** methods make use of anchors (i.e., pre-defined proposals) to expedite locating objects anywhere in the image. These include the R-CNN series [4], [28], which designs a two-stage pipeline to first extract potential regions of interest using a convolutional region proposal network. In the second stage, each proposal is further classified and its bounding box is refined. This line of method achieves state-of-the-art accuracy. However, such a sequential pipeline imposes a bottleneck to real-time inference. Alternatively, detectors such as YOLO [3] and SSD [2] bypass the intermediate region proposal, directly performing bounding box regression and classification across every image feature location in a single forward-pass. Generally, one-stage detectors are capable of real-time inference without compromising much accuracy, hence are widely considered when speed is of priority.

Even though obtaining balanced performance in speed and accuracy, single-stage detectors heavily rely on pre-defined proposals. As a result, their detection workflow revolves classifying and regressing over numerous anchors densely sampled across every location of the image feature. Such an approach not only incurs complicated IoU calculation during training when matching anchors and groundtruth objects, but also generates an enormous amount of redundant bounding boxes to be filtered by the non-maximal suppression (NMS) algorithm. Often, achieving precise localization on specific datasets also hinges on heuristic anchor design and placement, hampering the generalization ability of the detectors. Recently, anchor-free detectors tackle the detection problem as keypoint estimation and demonstrate comparable accuracy [27], [29], [30].

Action recognition is generally treated as a video classification problem. Early attempts include applying a shared 2D CNN on video frames independently, and then aggregating frame-wise scores for the final video-level prediction [31]. Such an approach lacks the ability to model intra-frame dynamics. To reason the temporal context across consecutive frames, the two-stream CNN frame proposed by Simonyan *et al.* [12] demonstrates an effective strategy modeling appearance-motion correspondences. It is constructed with two feed-forward pathways, where each CNN backbone learns and extracts spatial or temporal features separately (typically from RGB and optical flow stream); results of the two streams are then combined by different fusion strategies. This fundamental framework has been expanded in many subsequent research [32], [33]. Even though the two-stream approach can seamlessly exploit 2D CNN backbones, it depends on the separate motion flow stream which itself is computationally costly to acquire on-site.

Alternative to separately modeling temporal information from motion stream, 3D CNN can jointly learn spatiotemporal pattern from a stack of RGB frames. Recently, the emergence of large-scale action dataset (e.g., Kinectics [14]) gives rise to pre-trained models more suitable for video-related tasks, allowing 3D CNN to demonstrate comparable and even superior temporal modeling capacity than two-stream CNN. On the downside, 3D CNN incurs a massive amount of extra parameters and computational cost than their 2D counterparts, inevitably making powerful computing hardware (e,g., high-end GPUs) a necessity. To reduce computational cost associated with 3D convolution, architectures such as P3D [23] and R(2 + 1)D [24] decouple every 3D convolution into a 2D-spatial and a 1D-temporal convolution in sequence. Other attempts include directly applying 3D convolutional layers on top of spatial cues of higher semantics (i.e., 2D CNN features instead of raw video frames) [26]. In a different direction, several works propose performing partial channel-shifting across 2D CNN features in time, demonstrating an efficient workaround of spatiotemporal modeling via 2D filters [21], [22].

Spatiotemporal action localization simultaneously localizes and classifies action instances in time and space from videos by forming action proposals. Leading approaches leverage CNN-based detectors to locate individual action instances on each frame. They also adopt the two-stream CNN framework, combining complementary appearance and motion features or directly fusing their bounding-box predictions at test time (e.g., union of bounding boxes) to augment frame-level detection [5]–[7]. To better leverage the temporal continuity of consecutive frames, several works perform tubelet detection by taking a short clip of successive frames at once and jointly inferring actions via 3D cuboid regression & classification [8]–[10], [17]. In the common practice, frame-wise detection are linked over time by the Viterbi algorithm based on IoU matching, yielding long-range tubes for localizing actions in trimmed/untrimmed videos. Following the prevalence of adopting 3D CNN for action recognition, more recent studies in spatiotemporal localization incorporate these architectures as the backbone feature extractors [15], [16], [18], [19], [34] [35].

III. OUR METHOD

We describe technical details of the Temporal feature Exchange and Difference action detector (coined TEDdet). The proposed action detector builds upon 2D CNN and performs spatiotemporal action localization over multiple frames as tubelets. Our method incorporates CenterNet as the detector block, which will be briefly reviewed in this section. Following the review, we first introduce the two spatiotemporal modeling modules: Temporal Feature Exchange (TE) and Temporal Feature Difference (TD). The former aims to adaptatively aggregate abstracted visual cues over successive video frames to induce multi-frame feature interaction and learning, while the latter captures implicit motion information for tracking action centers in time. Finally, we unfold TEDdet's full architecture integrating the above components as well as our online action tube generation algorithm.

A. CenterNet

Unlike mainstream CNN-based detectors which infer objects from pre-defined anchors, CenterNet represents objects by their bounding boxes' center points, converting the detection problem to keypoint estimation. Concretely, given an input RGB frame $I \in \mathbb{R}^{3\times H\times W}$ (H and W denote the height and width of the frame), CenterNet extracts high-resolution feature maps from the designated 2D CNN backbone followed by an up-scaling decoder block. The resulted feature $F \in \mathbb{R}^{C\times \frac{H}{R}\times \frac{W}{R}}$, where C and R denote the feature channel and downsampling ratio, is used to predict object centers in the form of a multi-channel keypoint heatmap. Within this heatmap of dimension $cls \times \frac{H}{R} \times \frac{W}{R}$ where cls corresponds to the number of object classes, each peak represents a potential object center of a certain class. Meanwhile, detected centers allow objects' bounding boxes to be regressed directly from corresponding locations in image feature F. In practice, keypoint estimation and size regression are handled by two separate branches (typically recognized as **Center** and **Box** branch, respectively). As each feature location is no longer manifested by overlapping anchors, CenterNet's detection pipeline mainly involves the feed-forward inference without NMS post-processing to remove redundant detection.

B. TEMPORAL FEATURE EXCHANGE: MULTI-FRAME FEATURE AGGREGATION

Beyond frame-level object detection, we adapt the keypoint estimation paradigm of CenterNet for video action detection. As pointed out earlier, even though 3D CNN excels to capture rich spatiotemporal context over successive frames, it tremendously raises computational cost and training complexity. With this insight, we present the Temporal Feature

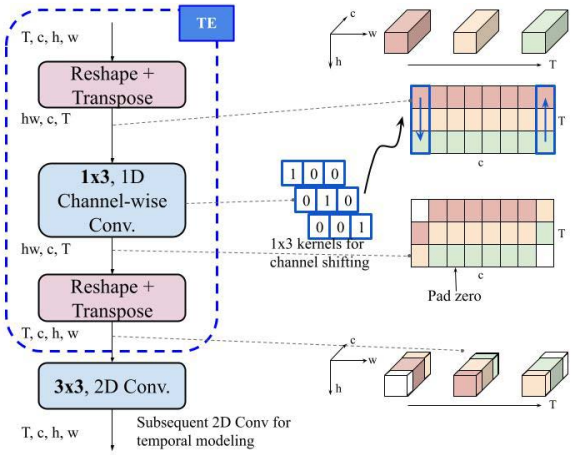


FIGURE 1. Architecture of Temporal Feature Exchange. The left displays the workflow of TE, while its effect on features is illustrated on the right. The grids in white represent features filled with zeros (due to zero-padding at temporal borders).

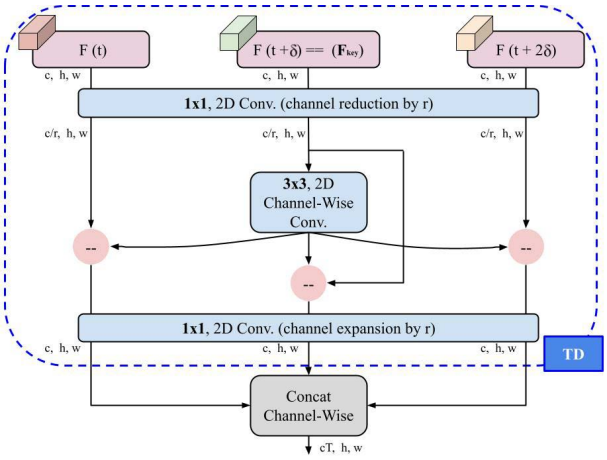


FIGURE 2. Architecture of Temporal Feature Difference.

Exchange (TE) module to facilitate modeling action-specific pattern on top of 2D CNN. Given a set of successive action frames, the TE module serves to aggregate supportive context among nearby frames by partially exchanging their spatial features in a channel-wise manner. As the outputted features have now interacted with those at different time steps, any subsequent 2D convolution can implicitly reason spatiotemporal information from such temporal-aware 2D features.

Figure 1 illustrates the architecture of our TE module. Formally, given a sequence of T RGB frames, they can be fed to a standard 2D CNN in parallel (concatenated in the batch axis) and transformed to abstract feature tensor $F \in \mathbb{R}^{T\times c\times h\times w}$, where c, h, and w denote the number of channels, height, and width of the feature. The TE module takes such a tensor as input and operates as follows. Firstly, F is reshaped and transposed to $F' \in \mathbb{R}^{hw\times c\times T}$, where spatial dimensions h and w are collapsed into one. Feature exchange between adjacent frames is then carried out by 1D channel-wise temporal convolution defined by kernel $K \in \mathbb{R}^{c\times 1\times 3}$. Here, each 1×3 kernel of K convolves with a feature channel of F' independently. The weight of these kernels are specifically initialized as: [1, 0, 0], [0, 0, 1] or [0, 1, 0], each corresponding to a temporal forward-shift, backward-shift, and no-shift operation, respectively. The proportion of the three shift operators dictates the extent of interaction among nearby features. The above operation can be summarized as follows:

$$\begin{aligned} F'_x &= K * F', \quad K \in \mathbb{R}^{c\times 1\times 3}, \quad F' \in \mathbb{R}^{hw\times c\times T} \\ K[c_f, 1, :] &= [1, 0, 0], \quad forward - shift \\ K[c_b, 1, :] &= [0, 0, 1], \quad backward - shift \\ K[c_n, 1, :] &= [0, 1, 0], \quad no - shift \\ 1 &\le c_f < c/div, \\ c - c/div &\le c_b < c, \\ c_n &\in c - \{c_f, c_b\}, \end{aligned} \tag{1}$$

where * and *div* denote the convolution operation and feature exchange ratio. In Equation (1), c_f and c_b indicate the channel indices where forward- and backward-direction feature exchange take place. The rest of the channels (c_n) does not exchange with those at neighboring features. When taking $div = 4$ as an example, (1) depicts $1/2$ of the total feature channels interacting with those at adjacent frames ($1/4$ forward and $1/4$ backward).

During the exchange, we perform zero-padding accordingly to fill up feature channels at the temporal borders. Finally, the resulted F'_x is transformed back to the original shape as F (i.e., $T \times c \times h \times w$). Additional convolution can be seamlessly appended after F'_x to further extract visual cues of higher semantics. As each of the T features in F'_x has partially interacted with others in adjacent times, this expands the temporal receptive field of subsequent 2D convolution by 3 for spatiotemporal modeling.

C. TEMPORAL FEATURE DIFFERENCE: PAIR-WISE DISPLACEMENT AS MOTION

In most existing action detectors, action tubelets are regressed from 3D cuboids (frame-level anchors expanded in time). Any positive cuboid sample for training is determined by the mean IoU between all its enclosed anchors and groundtruth action tubes. Similarly at test time, NMS post-processing needs to compute tubelets' mean IoU for removing low-confidence detection that are densely overlapped. On the other hand, anchor-free detector is conceptually easier when adapted for tubelet detection.

Following the notion of keypoint estimation [27], we predict action instances by their centers on the target frame only (i.e., key frame) while modeling their movement to other frames via an additional regressor. To achieve this, we propose the Temporal Feature Difference (TD) module for encoding relative motion between any two designated frames. Specifically, the relative motion between frames is approximated by the displacement in their abstract feature space.

Figure 2 depicts the workflow of TD, whose objective is to model action instances' offset with respect to the key frame. Given a feature tensor $F \in \mathbb{R}^{T\times c\times h\times w}$ extracted from T frames, a 1×1 2D convolutional layer is firstly applied to transform F into a compressed latent space for efficiency ($F^r \in \mathbb{R}^{T\times \frac{c}{r}\times h\times w}$). Next, we slice F^r into T portions along the temporal axis, resulting in T features of dimension $\frac{c}{r} \times h \times w$. For F^r_{key} and any F^r_t, which denote the key and t^{th} frame features, we compute their spatial displacement by element-wise subtraction. Note that prior to the subtraction, we follow the standard practice of applying a 2D channel-wise convolution on all F^r_t. This additional convolution has been proven useful for spatially aligning high-level instances acquired at different time steps [22], [25]. The above operation is described in Equation 2:

$$F^r_{dsp}(t) = conv_{trans} * F^r(t) - F^r_{key}, \quad 1 \le t \le T, \qquad (2)$$

where $conv_{trans}$ denotes the 3×3 2D channel-wise convolution. Here, $F^r_{dsp} \in \mathbb{R}^{T\times \frac{c}{r}\times h\times w}$ corresponds to all the displacement features between the key frame and others. Finally, we apply another 1×1 convolution to restore the channel number of F^r_{dsp} back to c. We will describe how the displacement feature is utilized for modeling the trajectory of input sequence in the following section.

D. TEDdet

The full architecture of TEDdet, which integrates CenterNet and two complementary modules (TE and TD), is summarized in Figure 3. To keep TEDdet effective yet compact, we employ the most lightweight ResNet variant, ResNet18 as the feature backbone. The input to TEDdet is T successive frames: $[I_t, I_{t+\delta}, \ldots, I_{t+(T-1)\delta}]$, where δ denotes the temporal stride between any two sampled frames. Unlike precedent methods such as [8], [11], our detector does not restrict δ to 1. In TEDdet, we select the middle frame of an input sequence as the key frame (I_{key}), which is the target for keypoint heatmap estimation.

1) TE AND CENTER BRANCH

Theoretically, the TE module can be inserted prior to any 3×3 convolutional layer in ResNet18. This would enable the following 2D filters to respond to specific spatiotemporal pattern while still able to exploit ImageNet pre-trained weights. However, unlike previous studies focusing on video-level action recognition [21], [22], performing temporal exchange in early layers of CNN risks distorting well-learnt spatial features that could be essential for the localization task in hand. Inspired by [26] which applies spatiotemporal modeling only towards deeper features of higher semantics, the TE module is mainly inserted right before Center branch to aggregate abstract multi-frame context for the key frame.

In practice, we implement two variants of temporal exchange, namely TE_{bi} and stacked TE_{uni}. The former conducts bi-directional exchange to simultaneously collect $1/2$ adjacent-frame features for the key frame (i.e., $1/4$ forward and $1/4$ backward by setting $div = 4$ in (1)). The latter performs two uni-directional exchange to separately aggregate $1/2$ forward/backward information into two key frame features ($div = 2$), which we then fuse by stacking along the channel dimension. Note that while a single TE module increases the temporal receptive field by 3, stacking multiple TEs can further enlarge the receptive filed for longer-range feature aggregation, as illustrated in Figure 4.

Our TE module is composed of learnable shift-operators that adaptively accumulate multi-frame visual context upon which Center branch uses for keypoint heatmap estimation. The design of Center branch follows that of CenterNet with minor adjustment. It consists of a 3×3 and 1×1 convolutional layer interleaved with ReLU non-linearity. The number of filters is set to 256 and number of action classes for the 3×3 and 1×1 convolution, respectively. The training objective for Center branch (l_{Center}) utilizes the same focal loss as in [27]. At test time, the obtained heatmap is further filtered to keep local peaks that are greater than their 8-connected neighbors. The top N peaks across all classes are considered candidate action centers where N is fixed to 100 in our study.

2) TD AND TRAJECTORY BRANCH

The predicted keypoint heatmap encodes centers of action instances in the key frame; it does not guarantee actions' locations in the rest of input frames. To address precise localization over the input sequence, we introduce Trajectory branch to track the movement of action instances with respect to the key frame. Prior to this branch, we insert our TD module which generates T displacement features $F_{dsp} \in \mathbb{R}^{T\times C\times \frac{H}{R}\times \frac{W}{R}}$. Each displacement feature estimates pair-wise offset between the corresponding frame from the key frame. To model the trajectory of the whole sequence, we stack every pair-wise offset along the channel dimension ($CT \times \frac{H}{R} \times \frac{W}{R}$) and feed as input to Trajectory branch.

The design of Trajectory branch follows that of our Center branch, consisting of a 3×3 and 1×1 convolutional layer interleaved with ReLU non-linearity. The output of Trajectory branch is the movement map $\hat{m}^{I_{key}} \in \mathbb{R}^{2T\times \frac{H}{R}\times \frac{W}{R}}$, where $2T$ denotes the center offsets between action instances at $[I_t, I_{t+\delta}, \ldots, I_{t+(T-1)\delta}]$ in sequence and those at I_{key} (in X and Y directions). Inherently, each grid in the movement map encodes horizontal and vertical offsets used for repositioning action centers on non-key frames.

To train the branch, groundtruth action centers on each input frame are first acquired the same way as in Center branch. Then, the groundtruth movement ($m^{I_{key}}$) of any action instance with respect to I_{key} is simply the offset between its center at I_{key} and those at other frames. Finally, Trajectory branch is optimized based on L1 loss as follows:

$$l_{Trajectory} = \frac{1}{n}\sum_{i=1}^{n} |\hat{m}_i^{I_{key}} - m_i^{I_{key}}|, \qquad (3)$$

where i indicates the i^{th} out of n action instances.

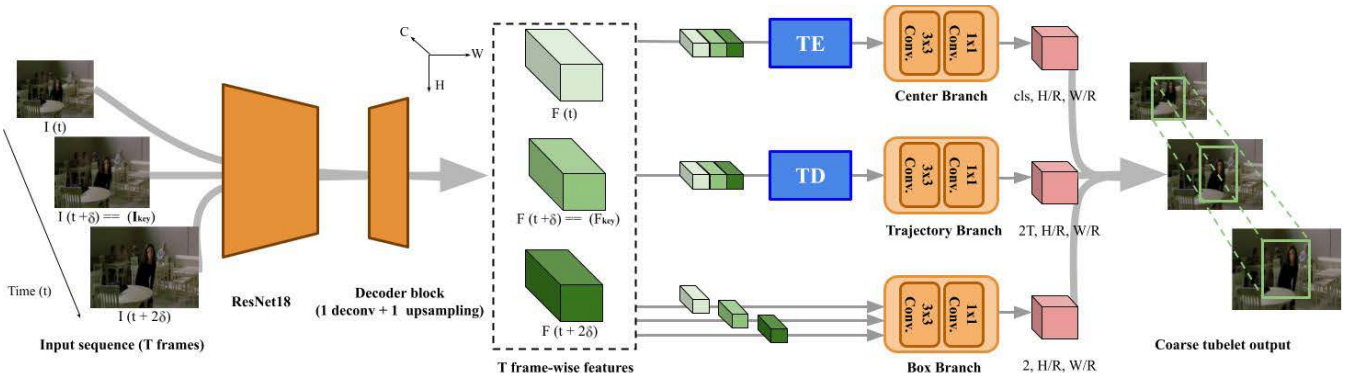


FIGURE 3. Architecture of TEDdet. Input to the model is a sequence of T RGB frames with temporal stride δ. Similar to CenterNet, TEDdet's backbone consists of ResNet18 followed by a decoder block (for adaptive up-sampling). TEDdet's detector head comprises three branches. The TE and TD modules are inserted prior to Center and Trajectory branch, respectively for spatiotemporal feature aggregation and trajectory modeling. The resulting outputs are coarse tubelets (determined by δ). Note that we also insert an additional TE module between ResNet18 and the decoder block (omitted in this figure) to endow feature interaction upon up-sampling.

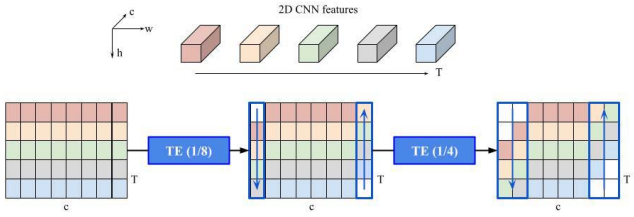


FIGURE 4. Stacking multiple TE modules. The above example applies two consecutive TE to enlarge the temporal receptive field of key frame feature (third row) to 5 frames. Note that in this figure, we directly display the "reshaped" tensor for clarity (height and width collapsed into one dimension); in practice, the reshape operation takes place inside the TE module.

3) BOX BRANCH

In TEDdet, we extend CenterNet's Box branch to regress the spatial extent of action instances from their centers. We assume that class-agnostic bounding box generation does not benefit from temporal modeling. Consequently, Box branch inputs the single frame's feature and regresses actions' bounding boxes independently for each frame. Specifically, it generates a size map of dimension $2 \times \frac{H}{R} \times \frac{W}{R}$. In the size map, each location encodes the height and width of a potential action instance whose center is identified there. We optimize this branch with L1 loss following [27]. At test time, action centers are first deduced by either Center or Trajectory branch.

The overall training objective of TEDdet is shown in Equation 4, where hyperparameter a, b, and c are set to 1, 1, and 0.1 respectively in accordance with [27].

$$l_{TEDdet} = al_{Center} + bl_{Trajectory} + cl_{Box} \tag{4}$$

4) INFERENCE

Given an action video of sufficient length, TEDdet predicts coarse action tubelets for every sequence of T frames (i.e., $[I_t, I_{t+\delta}, \ldots I_{t+(T-1)\delta}]$). The next sequence of length T frames to detect begins at $I_{t+\delta}$ where there exist $T-1$ overlapped frames between the two sequences. As each frame is propagated through the shared 2D CNN backbone independently, the computation of overlapping tubelets can be performed at an extremely low cost by caching $T-1$ previously obtained features in the buffer while only extracting those of the current (latest) frame. Retrieved and newly captured visual cues are simply fed to the detector branches together for action keypoint heatmap, trajectory, and size estimation. Once the resulting tubelets are acquired, TEDdet's feature buffer updates the cache to keep the latest $T-1$ features while dequeueing the oldest one. Conducting action inference in such a way well conforms to online detection where input video frames are streamed continuously.

E. ONLINE ACTION TUBE GENERATION

We adopt the online linking algorithm similarly employed by Kalogeiton *et al.* [8]. Given an input video stream, TEDdet detects N initial tubelets from which the top 10 (in terms of confidence scores) are initialized as active action links. As the video continues to be streamed and new tubelet candidates are detected, TEDdet enumerates through active links in descending order of their confidence scores and associates them with new tubelet candidates when matched.

In detail, whenever a collection of new tubelet candidates temporally overlaps with an active link, we associate the best-matched tubelet to that link in accordance with the mean IoU of their bounding boxes on overlapped frames. Additionally, two conditions are respected during the linking process. First, the best-matched tubelet should have a mean-IoU exceeding threshold $\tau = 0.5$. Second, each candidate tubelet can only be assigned to an active link. After adding a new tubelet, each link's confidence score is updated as the average score of all associated tubelets. An active link stops extending and is terminated (''inactive'') either when there no longer exist temporal overlaps with the newly detected tubelets, or videos stop being streamed.

The final action tubes are constructed from all the inactive action links. The temporal extent of any action tube is

determined by the starting frame of the initialized tubelet and the end frame of the lastly linked tubelet. Finally, we discard any resulting action tube having either a low confidence score or a short temporal duration. Action tube generation is carried out independently for each class.

Bounding Box Interpolation: When temporal stride (δ) of the input sequence exceeds 1, a fully linked action tube starting at frame 1 comprises coarse detection across $[I_1, I_{1+\delta}, I_{1+2\delta}, \ldots]$. To obtain dense frame-wise detection, we calculate bounding box results for intermediate frames using a simple coordinate-wise linear interpolation between any two available detection. Such a simple approach reasonably assumes that actions are smooth and continuous in videos.

IV. EXPERIMENTAL RESULTS

In this section, various aspects of TEDdet are evaluated on public action detection benchmarks. Specifically, we examine different architectural configurations and hyperparameters alongside their impact on multiple detection metrics. Finally, our method is validated by comparing against state-of-the-art action detectors.

A. DATASETS

We evaluate TEDdet on two challenging action datasets: UCF-24 and JHMDB-21. The former [36] is composed of 3207 temporally untrimmed videos of 24 action classes under different sporting categories. Following previous practices, we take 2290 of these video clips for training and test on the rest. The latter [37] consists of 928 short videos (maximum of 40 frames per clip) divided into three splits, with 21 action categories towards instantaneous actions such as *sit*, *stand*, and *walk*, etc. Each video is well trimmed and has a single action instance. For JHMDB-21, the experimental results are reported over the mean of three splits.

B. EVALUATION METRICS

We employ the standard frame-mAP and video-mAP (mean Average Precision) to measure the accuracy of the proposed action detector. The former metric examines the IoU between detected and groundtruth boxes for each frame separately; it does not depend on our online linking strategy. For frame-mAP, the IoU threshold is fixed at 0.5 throughout all experiments. On the other hand, video-mAP inspects spatiotemporal overlaps between linked action tubes and groundtruth tubes at multiple IoU thresholds: $0.2, 0.5, 0.75, [0.5 : 0.05 : 0.95]$. Besides accuracy, we also measure TEDdet's detection efficiency in terms of its model size (number of trainable parameters), FLOPs (number of floating-point operations), and runtime (FPS: frame-per-second).

C. IMPLEMENTATION DETAILS

1) NETWORK ARCHITECTURE

The original CenterNet attaches a decoder block of three deconvolution layers at the final convolution output of ResNet [38]. This serves to adaptively up-scale highly abstracted feature maps by 8 times (each deconv layer up-scales the feature by 2) to better detect small/overlapped objects. Different from object detection, we assume the likelihood of small actors emerging densely in a scene is low. Aiming to conduct highly accelerated and efficient detection, TEDdet's backbone re-uses ResNet18 but reduces the decoder block to one deconvolution layer followed by a bilinear upsampling layer. We also insert an extra TE module before the decoder block to introduce additional temporal modeling upon feature up-scaling.

Each RGB frame of an input sequence is resized to 288×288. Propagating the input sequence of T frames (input tensor: $3T \times 288 \times 288$) through ResNet18 and our reduced decoder block transforms the input to its video representation (of dimension $256T \times 36 \times 36$). Prior to the detector branches, we apply a 1×1 convolutional layer to reduce the number of channels by a factor of 4 for efficiency gain.

2) TRAINING DETAILS

TEDdet is trained with the Adam optimizer. We set the initial learning rate to $2.5e^{-4}$ for both datasets while initializing ResNet18 with COCO pre-trained weights. On JHMDB-21, we train our model for 10 epochs, during which we reduce the learning rate by a factor of 10 at the 5^{th} epoch. Likewise, UCF-24 is trained for 10 epochs; the learning rate is reduced by half at the end of each epoch since the second one. During training, we freeze ResNet18's first convolutional layer (to reduce chances of overfitting) and apply the same data augmentation as in [8]: photometric transformation, scale jittering, random cropping/expansion and location jittering, etc. In our experiments, all training has been conducted on an NVIDIA Titan V5 GPU with a mini-batch size of 16.

D. ABLATION STUDY

1) EFFECT OF SPATIOTEMPORAL FEATURE AGGREGATION AND TRACKING

We first conduct ablation study to validate the addition of TE and Trajectory branch (including TD). A baseline tubelet detector with no feature aggregation nor action center refinement is firstly established. In other words, given T frames as input, the baseline directly predicts the keypoint heatmap from the key frame and assumes that action centers remain at the same location in time. In this ablation study, we report frame-mAP as the evaluation metric on JHMDB-21; temporal stride δ and T are fixed to 5 and 3, respectively.

Table 1 summarizes varied configurations and performances of TEDdet. It can be observed that all configurations benefit from Trajectory branch by approximately 2.5 frame-mAP. As Trajectory branch does not concern any aspect of classification, these results highlight the importance of refining action centers in time to cope with moving actors in videos. Notably, the increases in GFLOPs and model parameters are fairly minimal when adding TD and Trajectory branch as they only execute a few layers of convolutional operations on low-resolution feature maps.

TABLE 1. Accuracy, FLOPs, and model size comparison over variants of TEDdet (JHMDB-21). TEDdet performs best in terms of accuracy when incorporating stacked TE_{uni} and Trajectory branch.

CenterNet	✓	✓	✓	✓	✓	✓
TE_{bi}			✓	✓		
stacked TE_{uni}					✓	✓
TD+Traj. branch		✓		✓		✓
Frame-mAP	51.33	53.98	55.82	58.15	58.34	**61.15**
GFLOPs	4.34	4.53	4.34	4.53	4.40	4.60
Param. (M)	13.73	14.18	13.73	14.18	13.88	14.33

While equipping TD and Trajectory branch, TEDdet's accuracy boosts significantly from the baseline when incorporating feature exchange (nearly 7 and 10 mAP in TE_{bi} and stacked TE_{uni} respectively). This phenomenon is expected. When $T = 3$, a bidirectional exchange essentially aggregates $1/4$ features each from the forward and backward direction into the key frame. On the other hand, stacked TE_{uni} is able to aggregate and retain more information from all three frames, endowing more context to differentiate actions. Since TE only introduces the shift-operators implemented by 1D convolutions, it is extremely efficient and barely increases computation and model size.

2) EFFECT OF SEQUENCE COVERAGE

Intuitively, video spanning a longer duration embeds richer and more discriminating spatiotemporal context. However, longer sequences could potentially introduce irrelevant background cues, as well as possibly raising difficulty to track actions' trajectories. To investigate how video duration affects accuracy of the proposed detector, we conduct experiments on both JHMDB-21 and UCF-24 by varying the input sequence length T and temporal stride δ.

Figure 5 summarizes results of the above experiments. From there we observe that when $T = 3$, TEDdet's accuracy continues to arise on JHMDB-21 when the temporal stride expands ($\delta = [3, 5, 7, 10]$). This not only reflects the advantage of accumulating more diverse context, but also validates Trajectory branch's ability to track action centers further away from the key frame. We also experiment increasing T to 5 (while keeping δ at 5). Even though this configuration essentially has the same temporal coverage as $(T, \delta) = (3, 10)$, it slightly improves the accuracy due to incorporating more intermediate frame features. To verify whether TEDdet correctly models action's temporal structure (rather than naively gathering contextual information from multiple frames), we repeat the configuration for $(T, \delta) = (5, 5)$ while reversing the order of the input video at test time. The resulting frame-mAP of each action class is displayed in Figure 6. It can be noted clearly that actions that depend more on static visual cues (e.g., *brush_hair*, *climb_stairs*, *golf*, and *shoot_bow*, etc.) remain unaffected. On the other hand, those relying on strict temporal modeling suffers when the video sequence is reversed (e.g., *sit*, *stand*, *pick*, and *shoot_ball*, etc.), confirming that TEDdet is able to learn the temporal relation in actions.

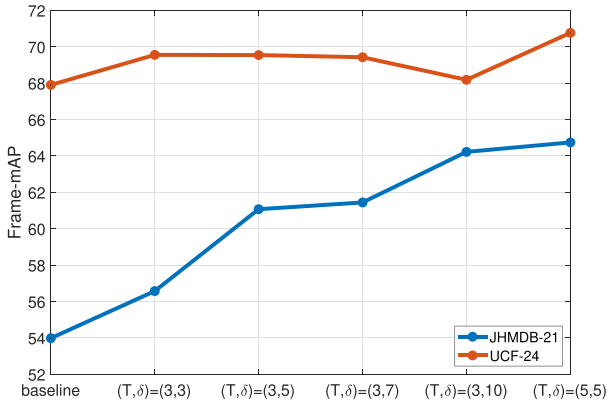


FIGURE 5. Accuracy comparison over varied input length (T) and temporal stride (δ).

For UCF-24, our detector also improves from the baseline method via spatiotemporal feature aggregation. In opposition to JHMDB-21, increasing the temporal stride for UCF-24 exhibits little influence in terms of accuracy, suggesting that strong spatial context around the key frame more or less suffices to determine the underlying actions. As pointed out in [25], UCF-24 is recognized having strong scene-related cues where background information highly correlates with an action category. Hence, the efficacy of temporal modeling saturates quickly. Lastly, we observe that setting (T, δ) as (5, 5) significantly outperforms (3, 10) by 2.5 mAP even though the two configurations share the same temporal coverage. We suspect that in the former case, the TD module and Trajectory branch manage to track action centers' movement more precisely from the extra intermediate frame-features.

3) EFFECT OF VARYING SEQUENCE COVERAGE AT TRAIN/TEST TIME

In the previous experiment, we train and evaluate target models with matching δ. To assess the generalization ability of the trained models during inference, we examine them (i.e., models trained by $\delta_{tr} = [3, 5, 10]$) on JHMDB-21 by varying δ at test time ($\delta_{te} = [3, 5, 7, 10]$). JHMDB-21 is selected as it more prominently reflects the effect of temporal modeling according to our previous experiments.

The robustness experiments are reported in Figure 7. We observe that models trained at larger δ_{tr} consistently perform comparably or better than others when tested on different δ_{te}. This remains true even when δ_{tr} and δ_{te} are far apart, e.g., $(\delta_{tr}, \delta_{te}) = (10, 3)$. It can also be seen that at $\delta_{tr} = 5$ and $\delta_{tr} = 3$, our models manage to adapt at first when tested at slightly larger temporal strides, but soon degrade in accuracy. These results imply that training with sparsely annotated frames (especially at higher temporal strides) potentially introduces a higher variety of visual pattern for TEDdet to learn and robustly discriminate actions. Such an attribute not only helps to reduce training complexity over long video sequences, but also relaxes our detector's reliance on densely annotated groundtruth boxes.

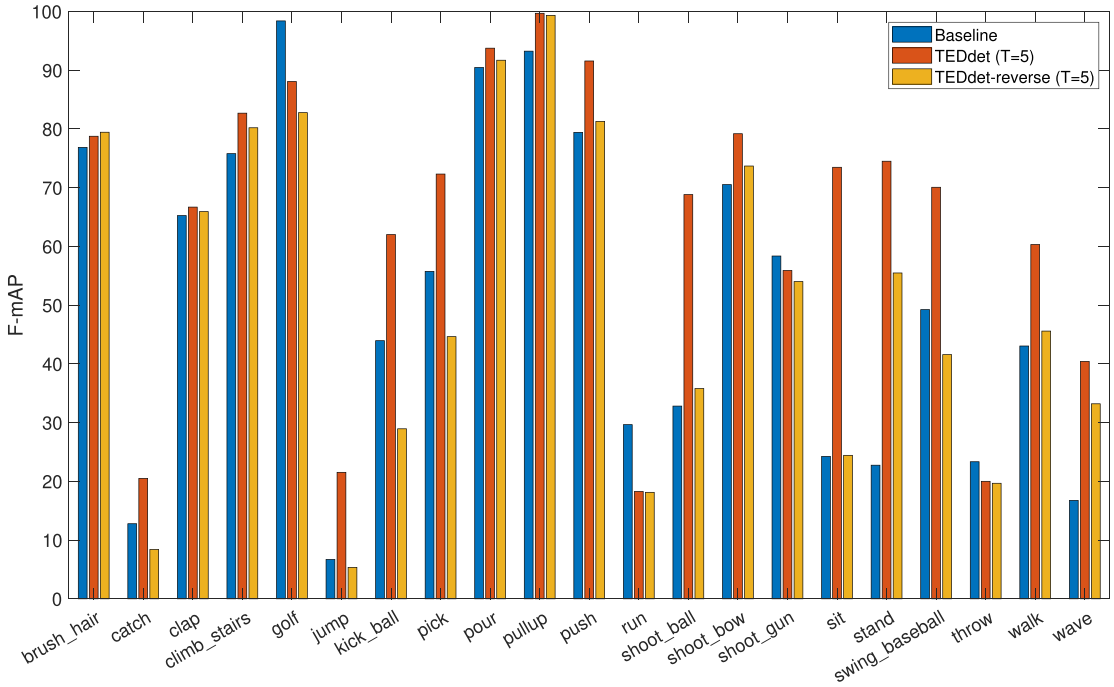


FIGURE 6. Per-class frame-mAP performance on forward (correct) and reversed input sequence at test time (JHMDB-21). The baseline performance is also included for easy comparison.

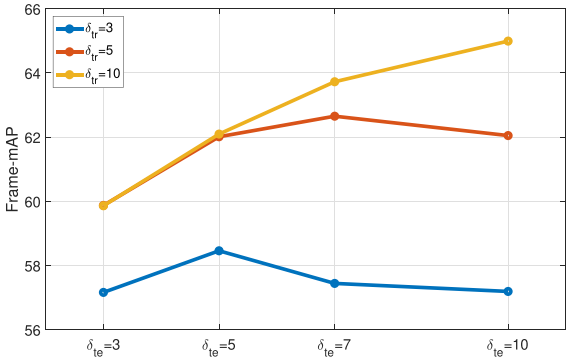


FIGURE 7. Accuracy comparison (JHMDB-21) over trained models ($\delta_{tr} = [3, 5, 10]$) tested on various temporal strides ($\delta_{tr} = [3, 5, 7, 10]$).

4) ACTION TUBE GENERATION AND RUNTIME

To evaluate TEDdet's spatiotemporal localization capability, we apply the online tube generation algorithm and compute video-mAP based on our top-performing configuration (i.e., T and δ as 5). We also measure runtime (FPS) to support our claim of real-time detection. These results along with frame-mAP are summarized in Table 2. Specifically, detection runtime is recorded over the complete time span to obtain action tubes (including data loading, tubelet inference, detection linking, and intra-frame interpolation) and then divided by the number of frames. We set the testing batch size as 1 to simulate processing a continuous video stream.

TEDdet significantly accelerates tubelet prediction and linking with its coarse-detection strategy. It achieves an overall inference speed greater than 85 FPS on both datasets. One may observe that even though the total runtime is similar between the two datasets, the time distribution is notably different. Specifically, data loading takes longer in JHMDB-21 as this dataset comprises shorter videos where TEDdet can not fully exploit feature caching-dequeuing. Instead, it will need to frequently clear its feature buffer and await for a new sequence of T frames upon any new video. On the other hand, tube generation takes perceivably longer in UCF-24. This is attributed to the fact that tubelet linking (which includes intra-frame interpolation) handles each action class independently. Unlike previous methods leveraging multi-thread CPU to complete such a task [8], our current implementation uses only a single CPU thread, thus taking longer to process UCF-24 than JHMDB-21 (i.e., 24 vs. 21 classes). We provide more qualitative analysis on TEDdet's video-mAP in the following section.

TABLE 2. Runtime, frame-mAP, and video-mAP performance. The total duration of action tube detection is broken down into three phases (top three rows under "Runtime") and reported in ms (millisecond/frame).

	JHMDB-21	UCF-24
Runtime (ms)		
Data loading	3.97	1.17
Detection	3.38	4.20
Tube generation	3.53	6.21
Speed (FPS)	92	86
Accuracy		
Frame-mAP	64.74	70.76
Video-mAP	67.86 67.39 53.74 44.71	74.57 50.41 21.82 25.04

E. COMPARISON WITH STATE-OF-THE-ARTS

In this section, we compare TEDdet with state-of-the-art action detectors. As our work aims at highly efficient and

TABLE 3. State-of-the-art comparison. In the table, "AF", "RTF", and "TS" denote accurate flow, real-time flow and two-stream, respectively. "Input frames" describes the type of modality and number of frames used as input by the detection model. Methods with * output action tubelets over multiple frames.

Method	Backbone	Input frames	JHMDB-21					UCF-24					FPS
			F-mAP	Video-mAP				F-mAP	Video-mAP				
				0.2	0.5	0.75	0.5:0.95		0.2	0.5	0.75	0.5:0.95	
Peng et al. [6]	VGG16×2	1RGB+5OF	58.5	74.3	73.1	−−	−−	−−	73.5	32.1	2.70	7.30	4
Singh et al. [5]-AF	VGG16×2	1RGB+1OF	−−	73.8	72.0	44.5	41.6	−−	73.5	46.3	15.0	20.4	7
Singh et al. [5]-RTF	VGG16×2	1RGB+1OF	−−	67.5	65.0	36.7	38.8	−−	70.2	43.0	14.5	19.2	28
Kalogeiton et al. [8]*	VGG16×2	6RGB+30OF	65.7	74.2	73.7	52.1	44.8	69.5	76.5	49.2	19.7	23.4	30
Saha et al. [10]*	VGG16×2	2RGB+10OF	−−	73.5	72.8	59.7	48.1	−−	78.5	49.7	22.2	24.0	21
Zhao et al. [9]*	VGG16	6RGB	−−	−−	58.0	42.8	34.6	−−	75.5	48.3	22.1	23.9	25
Zhao et al. [9]-TS*	VGG16×2	6RGB+30OF	−−	−−	74.7	53.3	45.0	−−	78.5	50.3	22.2	24.5	12.5
Zhang et al. [7]	VGG16×2	3RGB	37.4	−−	−−	−−	−−	67.7	74.8	46.6	16.7	21.9	38
Köpüklü et al. [16]	Dark19+3DResNext101	16RGB	74.4	87.8	85.7	58.1	−−	−−	75.8	48.8	−−	−−	34
Li et al. [11]*	DLA34×2	7RGB+35OF	70.8	77.3	77.2	71.7	59.1	78	82.8	53.8	29.6	28.3	25
Li et al. [11]-light*	ResNet18	7RGB	57.3	59.8	61.6	55.4	44.7	68.8	76.3	49.1	23.7	25.1	24
TEDdet*	ResNet18	5RGB	64.7	67.9	67.4	53.7	44.7	70.8	74.6	50.4	21.8	25.0	89

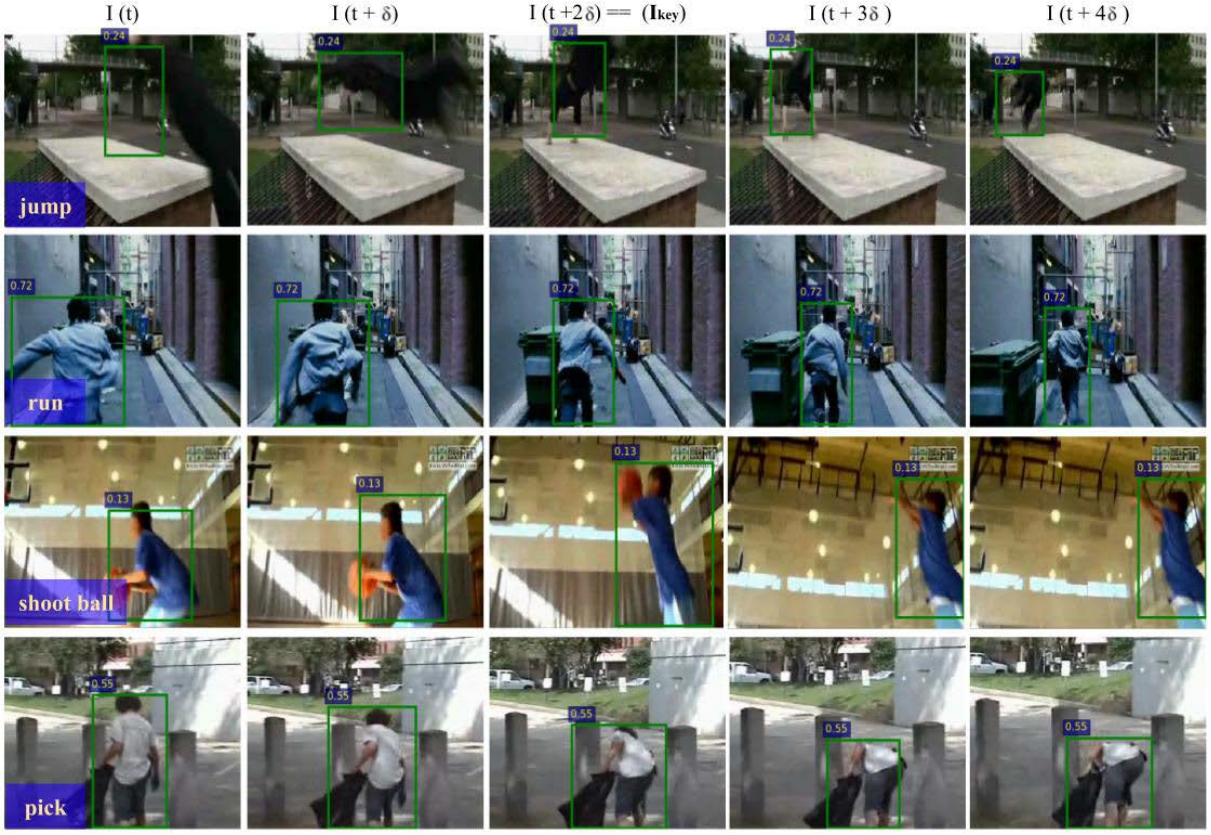


FIGURE 8. Examples of action sequences and detection where actors undergo significant location shift. Here, we only show predicted tubelets of the correct class (with confidence score above 0.1). In the first row, TEDdet could not precisely track the "jump" instance throughout the whole sequence due to drastic motion blur and indiscernible background cues. On the other hand, other actions (*run*, *shoot_ball*, and *pick*) with perceivable location shift are tracked properly. In the above examples, both T and δ are set to 5.

real-time detection solutions for realistic deployment, only methods which explicitly consider both accuracy and speed are taken into account. Table 3 presents a comprehensive list of top-performing action detectors and summaries of their methods (i.e., backbones and input types). Any method having repeated backbones adopts the two-stream CNN framework and makes use of optical flow (denoted by ''OF'' in the table).

It can be observed from Table 3 that TEDdet significantly outperforms others in terms of speed. This is mainly attributed to its coarse-to-fine detection paradigm which is able to accelerate action tube generation. Among all detectors listed in the table, TEDdet is also equipped with the most lightweight CNN backbone and does not depend on an additional model to extract optical flow features. Last but not least, our model leverages the least number of input frames

for action inference, inherently reducing computational cost (i.e., FLOPs) by many folds.

Even when prioritizing detection speed and efficiency in its design, TEDdet retains decent accuracy (more reflected in frame-mAP). Without relying on optical flow and two-stream CNN, our detector obtains comparable and even higher score than [7], [8]. The work of Li *et al.* [11] obtains impressive accuracy on both datasets but still counts on pre-computed optical flow as well as a stronger 2D backbone. When we adapt their pipeline closer to TEDdet's lightweight setting (i.e., to replace DLA34 by ResNet18, remove the optical flow stream, and keep its temporal stride δ as 1), the strength of TEDdet becomes evident. Köpüklü *et al.* [16] also demonstrate impressive spatiotemporal modeling capacity by fusing 2D and 3D CNN features. However, their detection on every target frame makes use of contextual information extracted from 16 nearby RGB frames via a very deep 3D CNN architecture.

We observe that the proposed action detector has more rooms for improvement in its video-mAP at lower IoUs (more prominent in JHMDB-21). The relatively low performance in this metric mainly results from TEDdet retaining a higher number of low-confidence action tubes after tubelet-linking (which contributes to more false-positive samples). Typically, low-confidence tubelets tend not to be consistent in time and can be easily discarded by dense-tubelet detectors [8], [11], [17] during the linking procedure. However, as TEDdet coarsely detects from an extended video sequence followed by intra-frame interpolation, it assumes the consistency of action class and confidence score for any action instance within this duration. Consequently, it becomes more difficult for TEDdet to suppress false-positive detection later through linking.

In addition to low-confidence detection, it comes to our attention that in a few extreme cases when actors undergo drastic spatial translation in time, TEDdet's TD and Trajectory branch are challenged to track action centers' location away from key frames (refer to the top-most example in Figure 8). In this case, the IoU-based linking strategy no longer guarantees the validity of linked action tubes, which negatively impacts video-mAP. In conclusion, our TD module and Trajectory branch may require a more sophisticated design to accommodate different degrees of movement. Overall, TEDdet still demonstrates high robustness coping with most other scenarios where highly perceivable shift in actors' locations is present. Along with its lightweight design and real-time inference speed, TEDdet is more compatible with computation-constrained devices and better appeals to deployment in real-world scenarios.

V. CONCLUSION AND FUTURE WORK

In this paper, we present a compact action tubelet detector termed TEDdet which builds upon 2D CNN and anchor-free detector. Equipped with our Temporal Feature Exchange module, TEDdet aggregates action-specific spatiotemporal cues over successive frames. In addition, our Temporal Feature Difference module alongside a tracking regressor facilitates modeling trajectories of action instances and refining tubelet localization. Complemented with the coarse-to-fine detection paradigm and online linking algorithm, TEDdet retains competitive accuracy against other top performers while being more lightweight and real-time capable by many folds (89 FPS). Without relying on 3D CNN or optical flow, the proposed detector is more feasible for edge device deployment in practical applications. Our future works include seeking a more precise approach to trace trajectories of moving action instances, as well as exploring ultra-efficient 3D CNN architectures for better video representations. We will also precisely customize TEDdet for embedding onto different edge devices (e.g., NVIDIA Jetson TX2 or Xavier GPU) to validate our detector's compatibility on resource-constrained vision systems.

REFERENCES

[1] M. S. Hutchinson and V. N. Gadepally, "Video action understanding: A tutorial," *IEEE Access*, vol. 9, pp. 134611–134637, 2021.
[2] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and C. AlexanderBerg, "SSD: Single shot multibox detector," in *Computer Vision*. Cham, Switzerland: Springer, 2016, pp. 21–37.
[3] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 779–788.
[4] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 91–99.
[5] G. Singh, S. Saha, M. Sapienza, P. Torr, and F. Cuzzolin, "Online real-time multiple spatiotemporal action localisation and prediction," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 3637–3646.
[6] X. Peng and C. Schmid, "Multi-region two-stream R-CNN for action detection," in *Computer Vision*. Cham, Switzerland: Springer, 2016, pp. 744–759.
[7] D. Zhang, L. He, Z. Tu, S. Zhang, F. Han, and B. Yang, "Learning motion representation for real-time spatio-temporal action localization," *Pattern Recognit.*, vol. 103, Dec. 2020, Art. no. 107312. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0031320320301163
[8] V. Kalogeiton, P. Weinzaepfel, V. Ferrari, and C. Schmid, "Action tubelet detector for spatio-temporal action localization," in *Proc. ICCV*, 2016, pp. 4405–4413.
[9] J. Zhao and C. G. M. Snoek, "Dance with flow: Two-in-one stream action detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 9935–9944.
[10] S. Saha, G. Singh, and F. Cuzzolin, "Two-stream AMTnet for action detection," 2020, *arXiv:2004.01494*.
[11] Y. Li, Z. Wang, L. Wang, and G. Wu, "Actions as moving points," in *Proc. ECCV*, 2020, pp. 68–84.
[12] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *Advances in Neural Information Processing Systems*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds. Red Hook, NY, USA: Curran Associates, 2014, pp. 568–576.
[13] H. Xu, A. Das, and K. Saenko, "R-C3D: Region convolutional 3D network for temporal activity detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 5783–5792.
[14] J. Carreira and A. Zisserman, "Quo vadis, action recognition? A new model and the kinetics dataset," in *Proc. IEEE Int. Conf. Comput. Vis.*, Jul. 2017, pp. 6299–6308.
[15] R. Hou, C. Chen, and M. Shah, "Tube convolutional neural network (T-CNN) for action detection in videos," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 5822–5831.
[16] X. Wei and G. Rigoll, "You only watch once: A unified CNN architecture for real-time spatiotemporal action localization," 2019, *arXiv:1911.06644*.
[17] X. Yang, X. Yang, M.-Y. Liu, F. Xiao, L. S. Davis, and J. Kautz, "STEP: Spatio-temporal progressive learning for video action detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 264–272.

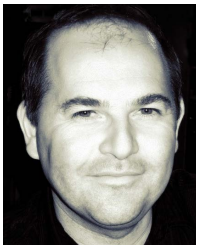
[18] J. Wei, H. Wang, Y. Yi, Q. Li, and D. Huang, "P3D-CTN: Pseudo-3D convolutional tube network for spatio-temporal action detection in videos," in *Proc. Int. Conf. Inf. Proc. (ICIP)*, 2019, pp. 300–304.
[19] J. Zhao, Y. Zhang, X. Li, H. Chen, S. Bing, M. Xu, C. Liu, K. Kundu, Y. Xiong, D. Modolo, I. Marsic, C. G. M. Snoek, and J. Tighe, "TubeR: Tubelet transformer for video action detection," 2021, *arXiv:2104.00969*.
[20] C.-Y. Wu, C. Feichtenhofer, H. Fan, K. He, P. Krahenbuhl, and R. Girshick, "Long-term feature banks for detailed video understanding," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019.
[21] J. Lin, C. Gan, and S. Han, "TSM: Temporal shift module for efficient video understanding," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 7083–7093.
[22] Y. Li, B. Ji, X. Shi, J. Zhang, B. Kang, and L. Wang, "Tea: Temporal excitation and aggregation for action recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2020, pp. 909–918.
[23] Z. Qiu, T. Yao, and T. Mei, "Learning spatio-temporal representation with pseudo-3D residual networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 5533–5541.
[24] D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun, and M. Paluri, "A closer look at spatiotemporal convolutions for action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Oct. 2018, pp. 6450–6459.
[25] B. Jiang, M. Wang, W. Gan, W. Wu, and J. Yan, "STM: Spatiotemporal and motion encoding for action recognition," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 2000–2009.
[26] M. Zolfaghari, K. Singh, and T. Brox, "Eco: Efficient convolutional network for online video understanding," in *Proc. ECCV*, Sep. 2018, pp. 695–712.
[27] X. Zhou, D. Wang, and P. Krähenbhl, "Objects as points," 2019, *arXiv:1904.07850*.
[28] J. Dai, Y. Li, K. He, and J. Sun, "R-FCN: Object detection via region-based fully convolutional networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 379–387.
[29] H. Law and J. Deng, "CornerNet: Detecting objects as paired keypoints," in *Proc. ECCV*, 2018, pp. 734–750.
[30] Z. Tian, C. Shen, H. Chen, and T. He, "FCOS: Fully convolutional one-stage object detection," in *Proc. IEEE Int. Conf. Comp. Vis. (ICCV)*, Oct. 2019, pp. 9627–9636.
[31] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," in *Proc. CVPR*, Sep. 2014, pp. 1725–1732.
[32] C. Feichtenhofer, A. Pinz, and A. Zisserman, "Convolutional two-stream network fusion for video action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Apr. 2016, pp. 1933–1941.
[33] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. V. Gool "Temporal segment networks: Towards good practices for deep action recognition," in *Proc. Eur. Conf. Comput. Vis.*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Cham, Switzerland: Springer, 2016, pp. 20–36.
[34] R. Su, W. Ouyang, L. Zhou, and D. Xu, "Improving action localization by progressive cross-stream cooperation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 12016–12025.
[35] K. Soomro, A. Zamir, and M. Shah, "UCF101: A dataset of 101 human actions classes from videos in the wild," Univ. Central Florida, Orlando, FL, USA, Tech. Rep. CRCV-TR-12-01, 2012.
[36] K. Soomro, A. Roshan Zamir, and M. Shah, "UCF101: A dataset of 101 human actions classes from videos in the wild," 2012, *arXiv:1212.0402*.
[37] H. Jhuang, J. Gall, S. Zuffi, C. Schmid, and M. J. Black, "Towards understanding action recognition," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2013, pp. 3192–3199.
[38] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Feb. 2016, pp. 770–778.

YU LIU received the B.S. degree in biomedical engineering from the University of Wisconsin—Madison, USA, in 2013, and the Erasmus Mundus M.S. degree in vision and robotics (VIBOT) conducted from the University of Burgundy, France, University of Girona, Spain, and Heriot-Watt University, Edinburgh, U.K., in 2018. He is currently pursuing the Ph.D. degree in computer vison with the University of Burgundy.

His research interests include computer vision, scene understanding, mobile robotics, and deep learning applications. Since 2018, his latest work has been focusing on efficient deep learning architectures applied to video and action analysis for edge devices/embedded systems.

FAN YANG received the B.S. degree in electrical engineering from the University of Lanzhou, China, in 1982, and the M.S. degree in computer science and the Ph.D. degree in image processing from the University of Burgundy, France, in 1994 and 1998, respectively.

She is currently a Full Professor and a member of the ImViA Laboratory, University of Burgundy. Since the beginning of her career, she has published 47 international peer-reviewed journals, four book chapters, and over 90 conference papers. Her research interests include pattern recognition, machine learning, multi-spectral imaging, parallelism and real-time implementation on embedded systems, and more specifically, biometric image processing algorithms, and architectures.



DOMINIQUE GINHAC received the M.S. degree in engineering and the Ph.D. degree in computer vision from the University of Clermont Auvergne, France, in 1995 and 1999, respectively.

He joined the University of Burgundy, France, as an Assistant Professor, in 2000, and was promoted to a Full Professor in computer vision, in 2009. He was the Head of the Le2i Laboratory, from 2016 to 2019. He has recognized expertise in embedded computer vision, computational imaging, and real-time image processing. He has authored 40 international peer-reviewed journals and over 100 conference proceedings. His research interest includes deep learning on the edge applied to the analysis of human activities.