Murtaza, F., Haroon Yousaf, M., Velastin, S. A. y Qian, Y. (2020). Vectors of Temporally Correlated Snippets for Temporal Action Detection. *Computers and Electrical Engineering*, Vol 85.

DOI: https://doi.org/10.1016/j.compeleceng.2020.106654

Vectors of Temporally Correlated Snippets for Temporal Action Detection

Fiza Murtaza[a], Muhammad Haroon Yousaf[a,b,*], Sergio A. Velastin[c,d,e], Yu Qian[c]

[a]*Department of Computer Engineering, University of Engineering and Technology Taxila, Taxila 47080, Pakistan*
[b]*Swarm Robotics Lab, National Centre for Robotics and Automation, Pakistan*
[c]*Zebra Technologies Corporation, London SE1 9LQ, UK*
[d]*School of Electronic Engineering and Computer Science, Queen Mary University of London, London E1 4NS, UK*
[e]*Department of Computer Science, Applied Artificial Intelligence Research Group, University Carlos III de Madrid, Madrid 28270, Spain*

Abstract

Detection of human actions in long untrimmed videos is an important but challenging task due to the unconstrained nature of actions present in untrimmed videos. We argue that untrimmed videos contain multiple snippets from actions and the background classes having significant correlation with each other, which results in imprecise detection of start-end times for action regions. In this work, we propose Vectors of Temporally Correlated Snippets (VTCS) which addresses this problem by finding the snippet-centroids from each class which are discriminant for their own class. For each untrimmed video, non-overlapping snippets are temporally correlated with the snippet-centroids using VTCS encoding to find the action proposals. We evaluate the performance of VTCS on the Thumos14 and ActivityNet datasets. For Thumos14, VTCS achieves a significant gain in mean Average Precision (mAP) at temporal Intersection over Union (tIoU) threshold 0.5, improving from 41.5% to 44.3%. For the sports-subset of ActivityNet dataset, VTCS obtains 38.5% mAP @0.5 tIoU threshold.

Keywords: temporal action detection, action proposals, 3D-Convolutional network (C3D), bag of words, k-means clustering

1. Introduction

With the advancement in information and communication technologies, sensing devices have now become pervasive. The pervasiveness of camera devices has enabled recording of video data at any time and

[*] Corresponding author: Tel.: 92-51-9047574; fax: +92-51-9047420;
Email address: haroon.yousaf@uettaxila.edu.pk (Muhammad Haroon Yousaf)

anywhere. That gives rise to massive amounts of untrimmed video data being produced, which typically consist of several human-related activities and actions including some background activities as well. Therefore, there is a need for computer vision based methods to detect the actions of interest in such long and untrimmed videos so that they can be further processed for recognition purposes. Temporal action detection attempts to address this problem by detecting the start and end of an action of interest in an untrimmed video sequence. Having a long untrimmed video as an input to the process, temporal action detection entails answering, 'when does an action instance start and end and to which action class it belongs to?'. Other than actions of interest, all other actions and background activities comprise the background class. In contrast to temporal action detection approaches, human action recognition approaches [1, 2] only identify the action class in already trimmed video sequences. In contrast, temporal action detection also processes the untrimmed videos and provides information about the starting and ending points of all instances of actions present in the video. Due to the unrestricted nature of long untrimmed videos in both space and time, the task of temporal action detection is more challenging than conventional action recognition. Existing methods for temporal action detection have unsatisfactory performance as they are still unable to find the precise start-end times of each action instance present in long untrimmed videos. Temporal action detection has become a significant area of research due to its abundant applications in security and surveillance, health monitoring, video analytics and other domains.

Existing methods for temporal action detection have several disadvantages. First, these methods tend not to make use of the discriminative power of the snippet-centroids, which gives rise to incorrect detections as there exist several small clips having multiple frames (called snippets) that have high correlation with different actions as well as with the background snippets. Hence, these methods fail to distinguish an action of interest from other actions of interest and from the background class. In our previous work, we proposed the Bag of Discriminant Snippets (BoDS) [3] approach. BoDS utilizes the discriminant importance of key-snippets in the encoding process by learning the weights of key-snippets. BoDS classifies the candidate regions into action or background in an unsupervised manner which resulted in many incomplete action proposals of small duration. To overcome this problem requires the mapping of snippets to highly correlated snippet-centroids, extracted per-class, during encoding. For each snippet-centroid, the sum of maximum correlation with the best matching snippets per proposal is found, called Vectors of Temporally

Correlated Snippets (VTCS). For the incomplete proposals, the resulting VTCS will have less contribution of each action class therefore it will be filtered out. Second, many existing approaches [4–9] entail two-stage models, i.e., ‘proposal generation + classification’, which need a separate features extraction step for each stage which makes them computationally expensive.

This work also proposes a two-stage paradigm to detect multiple actions from untrimmed videos, which uses the same visual encoder for both stages hence it is more computationally efficient than [4–9]. In the first stage, the proposed method detects multiple action regions while rejecting the background regions. In the second stage, it classifies each action region into one of the predefined action classes. In both stages, the proposed VTCS based visual encoder is used to discriminate snippets belonging to multiple actions and background classes. VTCS is integrated with features from a 3D-Convolutional network (C3D) [10].

The proposed VTCS based temporal action detection framework has the following contributions:

1. A visual encoding method, called it VTCS, which discriminates multiple actions from each other and from the background actions. An efficient method is proposed to aggregate multiple video snippets into candidate proposals of varying length at each time step of the untrimmed video.

2. A simple yet effective two-stage temporal action detection framework that unrolls over an untrimmed video to encode and recognize multiple actions present in the given video. It can be learned using trimmed as well as untrimmed videos (utilizing the ground truth information for starting and ending times and the category label).

3. VTCS based temporal action detection framework attains state-of-the-art temporal action detection performance on benchmark datasets.

The rest of the paper is organized as follows: Section 2 presents the review of related temporal action detection methods; Section 3 introduces the proposed VTCS based temporal action detection method. Experimental results are discussed in Section 4 and the conclusions are given in Section 5.

2. Related Work

Recently, Convolutional Neural Networks (CNNs) have succeeded to improve the recognition accuracy for the task of human action recognition [1, 10]. However, temporal action detection approaches [11, 12] still need improvement. The conventional approaches for action detection make use of snippet-based classifiers [13–15] for detecting actions. Segment-CNN (SCNN) [4] uses 3D convolution to integrate temporal structures, but this method depends on the architecture beneath [10], which provides for only 16 frames. An action detection method, which is based on the Pyramid of Score Distribution Feature (PSDF) is presented in [11]. The computational complexity of PSDF is very high since it draws motion information at multiple scales. In [5], a Structured Segment Network (SSN) is proposed for modeling human activities in long untrimmed video sequences, based on a structured temporal pyramid.

The Temporal Actionness Grouping (TAG) method is discussed in [6] to find variable sized action proposals. This method is less feasible in practice for real-time applications and scenarios, as it depends on two thresholds for differentiating the action regions from the background as well as incomplete regions. Motion History Images (MHIs) are used in [16] for generating action proposals from untrimmed videos in an unsupervised way. However, this approach is dependent on the extraction of precise silhouettes, which is very hard and challenging for real-time videos and moving cameras. Convolutional De-Convolutional (CDC) Networks [7] detect action boundaries with the help of dense prediction at each frame. However, CDC depends on other action proposal approaches, e.g. [4], to generate proposals. Cascaded Boundary Regression (CBR) [9], Regional C3D (R-C3D) [17] and Temporal Unit Regression Network (TURN) [8] methods provide boundary refinement based on temporal boundary regression. However, these methods do not perform well for high tIoU thresholds.

Temporal Action Localization Network (TAL-Net) is proposed in [18], based upon the Faster RCNN [19] object detection framework. TAL-Net produced improved temporal action localization results for the Thumos14 and ActivityNet dataset, yet it is computationally expensive as it requires separate stages for proposal generation and classification. Complementary Temporal Action Proposal (CTAP) is put forward in [20] to generate action proposals. CTAP initially takes the proposals generated by TAG [6] and sliding windows. Then it applies a complementary filter to obtain precise action proposals. CTAP requires multiple stages to obtain precise action proposals which are then classified using the SCNN [4] action detector.

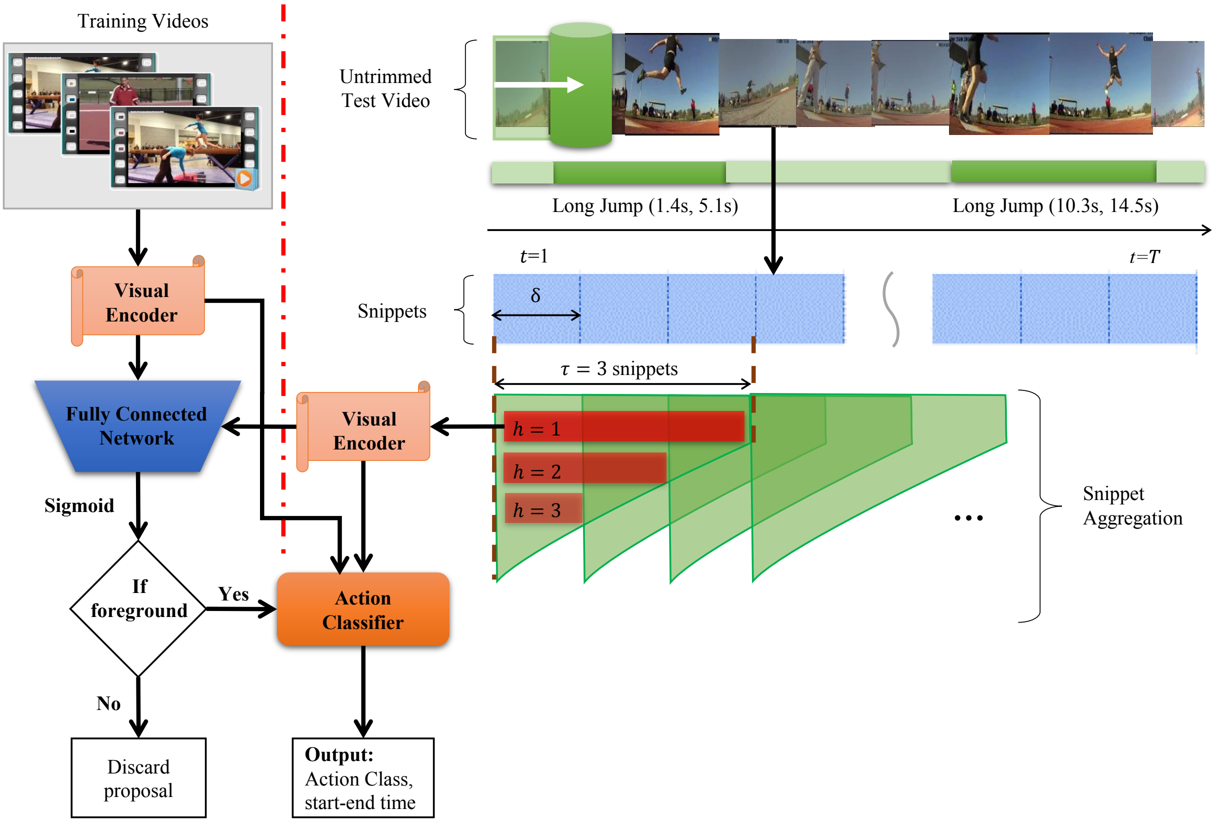


Figure 1: Block diagram of the proposed VTCS based temporal action detection method. For a given untrimmed video, VTCS based temporal action detection unrolls over each time step t and detects the foreground/background regions using a Fully Connected (FC) network. Proposals classified as foreground are further passed to SVM classifiers (learned on the trimmed action classes) which gives the final classification label and probability for each action proposal.

3. Temporal Action Detection System Overview

Given an untrimmed video $V = \{f_n\}_{n=1}^{N}$ with a total of N frames, the goal is to localize all the occurrences of actions of interest present in the video. Each video V has m action occurrences in the ground truth $\mathcal{G} = \{(g_m, g_m^{'}, l_m)\}_{m=1}^{G}$, where g_m, $g_m^{'}$ and l_m are the start time and the end time of the occurrence m respectively. l_m represents the action label and $l_m \in \{a_1, \cdots, a_C\}$, with a total of C action classes. The regions beyond g_m and $g_m^{'}$ are considered as background regions which are represented as the a_{C+1} class i.e. the background class.

The proposed framework generates a set of candidate proposals in a sliding-window manner and categorizes each of them into $C + 1$ classes (where the extra class corresponds to the background class) through three main modules: *visual encoder*, *snippet aggregation* and *proposal classification* module. The visual encoder module is used to extract the discriminant representation from the temporal regions of the test and training videos. It does that by first computing the snippet-based features at each time step and then by extracting class-specific clusters, which we call them snippet-centroids, from snippet set of $C + 1$ classes of the training data. These snippet-centroids are used for the encoding of candidate proposals and the training data. The snippet aggregation module aggregates multiple snippets of the untrimmed video into candidate proposals that are likely to contain actions of interest. The proposal classification module has two stages. In the first stage, the candidate proposal is classified into an action or a background proposal using an FC (fully connected) layer followed by a sigmoid function. In the second stage, the candidate proposal classified as foreground is further passed into class-specific SVM classifiers which predict the action category of the foreground proposal. Both stages share the same visual encoder which makes it computationally fast.

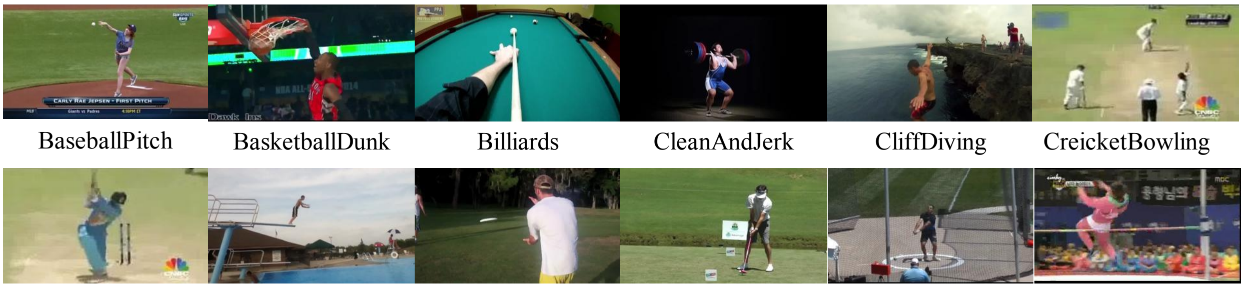
An encoding strategy is proposed to represent every candidate region which not only discriminates the action instances from each other but also from the background regions. An overview of the proposed action detection system is shown in Figure 1. The remainder of this section first provides the details of the visual encoder module (Section 3.1) which is the building block for the other two modules. Then, the snippet aggregation (Section 3.2) and the proposal classification (Section 3.3) modules are discussed in detail.

3.1. Visual Encoder

To extract the low-level spatiotemporal information from the video frames, C3D [10] is used. C3D has been chosen for feature extraction as it has been effectively used by other temporal action detection methods [4, 13, 14] to extract unit-level features, but the method can be used with other features. The unit is a snippet containing δ frames. Each video V is first segmented into $T = N \ / \ \delta$ non-overlapping snippets. Following standard practice [4, 10, 13, 14], δ is set to 16 frames to extract C3D features. For each video V, the snippet-level C3D features $\{s_t\}_{t=1}^{T}$ are extracted. A C3D model pre-trained on the publicly available Sports1M dataset is used [10]. The output of the *fc6* layer is used as the snippet-level C3D features.

3.1.1. Extracting snippet-centroids

Given a set of training videos, each training video is first divided into snippets to obtain their corresponding D-dimensional C3D feature representation. The snippet-centroids from each class are then extracted. It is important to extract the snippet-centroids from each class because the snippets within a specific action



BaseballPitch BasketballDunk Billiards CleanAndJerk CliffDiving CreicketBowling

CreicketShot Diving FrisbeeCatch GolfSwing HammerThrow HighJump

Figure 2: An example of snippet-centroids (one per class) from the 12 action classes of Thumos14 dataset. Although each snippet has a duration of 16 frames, for better visualization only one image is shown from each snippet.

class are not in general fully independent, but instead they can have similarity with the snippets of the other classes including the background class. To find a discriminant representation for the action and background regions, we need to find snippet-centroids from each class. To achieve this, class-specific clustering is performed on the training data to find the K snippet-centroids from $C + 1$ classes. For the class a_i, all snippets are compiled into a snippets matrix $X_i \in \mathbb{R}^{D \times n_i}$ $(i \in \{1, \cdots, C + 1\})$ where n_i and D are the total number of snippets for class a_i and the dimension of the feature vector respectively. For each snippet matrix X_i, class-specific clustering using K-means is performed using Euclidean distance to extract K snippet-centroids. k-means clustering starts with K randomly selected snippet-centroids, which act as the initial points for every cluster. Each snippet in the training data is assigned to the nearest snippet-centroid and then the centroids of newly formed clusters are recomputed. This process is repeated until newly formed snippet-centroids do not change. However, random initialization of snippet-centroids may lead to different cluster centres since k-means might get stuck in a local optimum and may not converge to the global optimum. To address this issue, k-means is run 20 times using different initializations of snippet-centroids and the final snippet-centroids of the run that resulted in the lowest sum of *sumd* is selected. *sumd* is a K-dimensional vector, where its i^{th} element represents the sum of point-to-centroid distances within cluster i. In this way, the newly formed clusters do not change. For $C + 1$ classes, $K \times (C + 1)$ snippet-centroids $\{S_j\}_{j=1}^{K \times (C+1)}$ are obtained from the training data. An example of some snippet-centroids is given in Figure 2.

3.1.2. VTCS encoding

After extracting snippet-centroids from the training data, they are used to encode a video clip containing a varying number of snippets into a fixed-length representation. For each snippet at time step t, its corresponding C3D feature representation s_t is first extracted. Then, the correlation of each snippet s_t with $K \times (C+1)$ snippet-centroids is calculated and the index of the snippet-centroid having a maximum correlation is obtained using Equation 1 below. For each snippet s_t, its corresponding encoded vector v_t is obtained using a hard-voting scheme to vote for the best matching snippet-centroid using Equations 1-3:

$$j = \underset{1 \leq r \leq K \times (C+1)}{\arg\max} \; corr(s_t, S_r) \tag{1}$$

$$v_t[j] = corr(s_t, S_j) \tag{2}$$

$$corr(s_t, S_r) = \frac{1}{D-1} \sum_{i=1}^{D} \left(\frac{s_t[i] - \mu_{s_t}}{\sigma_{s_t}}\right)\left(\frac{S_r[i] - \mu_{S_r}}{\sigma_{S_r}}\right) \tag{3}$$

where σ and μ represent the standard deviation and mean values respectively and D represents the dimension of the feature vector. v_t is a vector of size $K \times (C+1)$ initialized with all zeros. Then, the correlation of s_t with best matching snippet-centroid S_j is added at the j-th index. Note that in v_t, indexes $1 \cdots K$ indicate snippet-centroids belong to the first class, indexes $(K+1) \cdots 2K$ indicate second class and so on. The dimension of v_t depends upon the number of targeted classes, therefore a simple hard voting scheme is preferred for encoding than other high dimensional super vector encoding approaches [21]. Finally, for any region h with o_h and $o_h^{'}$ as starting and ending times respectively, the aggregation of multiple snippets into a fixed-length encoded vector is done using:

$$\phi(h, o_h, o_h^{'}) = \sum_{i=o_h}^{o_h + o_h^{'} - 1} v_i \tag{4}$$

Note that the obtained representation ϕ also produces a $K \times (C+1)$ dimensional vector where each of its elements represents the sum of maximum correlation for each of the corresponding snippet-centroid.

Our previous work i.e. BoDS [3] was based upon Bag of Words (BoW) but with the difference that the discriminative power of the key-snippets (centroids) was integrated during the encoding process in terms of discriminative weights. BoDS encoding for multiple snippets is calculated based upon the sum of weighted histograms, where the weight for the specific cluster is calculated by computing the ratio between the within-

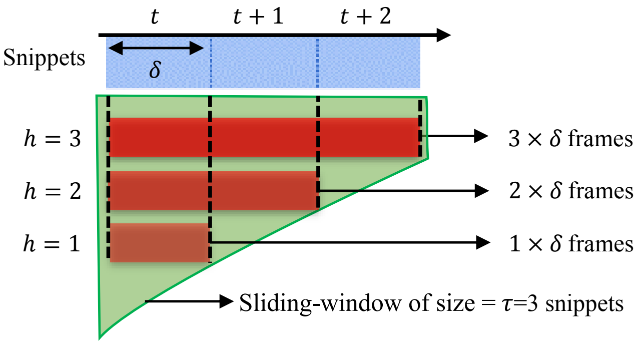


Figure 3: An example of τ=3 candidate proposals at time step t for a sliding-window of size τ=3 snippets. The size of each proposal is also shown in the form of a number of frames.

class assignments and the total assignments for that cluster. The proposed VTCS based representation has the following differences with the BoDS and BoW:

(i) VTCS does not incorporate the weights of the codewords during encoding process as done in BoDS.

(ii) During the encoding process, VTCS finds the correlation of each snippet with the $C+1$ snippet-centroids and assigns the snippet to the snippet-centroid with maximum correlation lets say j as given in Equation 1. Then the value of this maximum correlation is update at the jth location in encoding vector v as given in Equation 2. Neither BoDS nor BoW, incorporate correlation information in the encoding vector.

(iii) VTCS is a two-stage framework which filters the background proposals in the first stage and then in the second stage it classifies these proposals in one of the action classes using one-vs-all SVM classifiers. On the other hand, BoDS is a single stage framework and it classifies the proposals in one of the action classes by finding the maximum sum of weighted votes from the action classes.

3.2. Snippet Aggregation

To obtain the action proposals from the untrimmed videos, a sliding-window strategy is adopted of length equal to τ snippets from [3] to generate temporal proposals of multiple durations. At each time
step t, the window is slid while producing left-aligned proposals set $P_t = \{p_h\}_{h=1}^{\tau}$. For each snippet in the sliding-window, snippet-level C3D features s_t are first extracted which are then encoded into a vector v_t using Equation 2-3. For each proposal h in the proposals set P_t, h snippets are accumulated to find a

fixed-length representation $p_h = \phi(h, o_h, o_h^{'})$ using Equation 4, where $o_h = t$ and $o_h^{'} = h + t - 1$ represent the start and end times respectively for the proposal h in the sliding-window. Note that at each time step t, the starting time for each h-th proposal is the same, therefore the aggregated features for the proposal h can also be extracted in an efficient way using the previously extracted representation p_{h-1} as given by:

$$p_h = p_{h-1} + v_{t+h-1}, \qquad \forall\, h \in \{1 \cdots \tau\}, p_0 = 0 \tag{5}$$

As the proposals overlap in time, it is therefore more efficient to aggregate different sized proposals using sum pooling as the v_t is obtained only once for each non-overlapping snippet. Note that for the fixed sized window, τ proposals of duration equal to $1 \times \delta$, $2 \times \delta$, $3 \times \delta$, $\cdots$, $\delta \times \tau$ frames are obtained, as shown in Figure 3.

3.3. Proposal Classification

After extracting VTCS representation for the candidate proposals, they are categorized into one of the $C + 1$ classes using two stages. In the first stage, the candidate proposals are classified into two classes, i.e. action and background, by passing them to the FC network followed by a sigmoid activation (as shown in Figure 1). In the second stage, all the proposals classified as action or foreground are passed to the *action classifier* trained for C action classes. In this way, any candidate proposal will be classified into one of the $C + 1$ classes using two stages. Section 4.4, also presents the results for skipping the first stage i.e. the *FC network* stage and passing the candidate proposals directly to the *action classifier* stage trained at $C + 1$ (including background) classes.

For training an FC network, all ground truth occurrences in $\mathcal{G}$ of training videos are taken as a positive class and the remaining regions which are not the part of the annotated data are treated as a negative class. For each training region, its corresponding VTCS representation is found using Equation 4. The negative data is randomly sampled to have a 1:1 ratio with the positive data. During testing, each candidate proposal i is passed through the trained FC network and classified into either foreground or background via sigmoid activation. If the candidate proposals are classified as the foreground (action), they are then classified into one of the C action classes using a multi-class classifier based upon the same VTCS representation p_i. Since every action has different temporal structures, linear class-specific SVMs are trained using the ground truth occurrences from C action classes of the training videos. After passing p_i to the trained SVM, the class label $l_i \in \{a_1, \cdots, a_C\}$ and the classification scores for each class are obtained.

4. Experiments

4.1. Datasets

For the evaluation of the proposed temporal action detection framework, two publicly available datasets are used. First, the challenging Thumos14 dataset [22] is used. This dataset is comprised of untrimmed YouTube videos with a total of about 20 hours duration with 20 sports actions. Each untrimmed video has a duration of about 3.9 minutes containing multiple instances of an action. Each action instance is confined to a small portion of the entire video and the remaining part contains the background segments. Thumos14 is one of the most challenging datasets as it has large intra-class variability in action classes. For training, 200 untrimmed validation videos and for testing 213 untrimmed test videos are used.

Second, the ActivityNet dataset [23] is used. Similar to previous work [12, 8, 23], the proposed method was evaluated on the 'Sports' subset of ActivityNet-v1.1 dataset. Although both datasets share some sports activities, the average action duration in Thumos14 and ActivityNet datasets is 5 and 50 seconds respectively. Moreover, on average, Thumos14 videos are two times longer than those of ActivityNet videos i.e. 233 vs. 114 seconds. This difference in action and video length reflects the diverse nature of both datasets in terms of coarseness and temporal structures. The results on five top-level subsets of ActivityNet-v1.3 dataset are also reported.

Mean average precision (mAP) is used as a standard evaluation measure for both datasets. mAP is calculated at temporal Intersection over Union (tIoU) thresholds from 0.3 to 0.7 with a step of 0.1. The publicly available toolkit, provided by Thumos14 [22], is used to calculate mAP.

4.2. Implementation Details

During the testing phase, a sliding window of length equal to τ snippets is used to generate the candidate proposals of variable lengths. The value of τ is set by utilizing information from the training set on the maximum possible duration of actions. τ=64 snippets (1024/16=64, 16 being the snippet duration) are used, as the maximum duration of actions in Thumos14 dataset is about 1024 frames. The maximum action duration in ActivityNet dataset is about 1600 frames, therefore τ is set to 100 snippets (1600/16). In this way

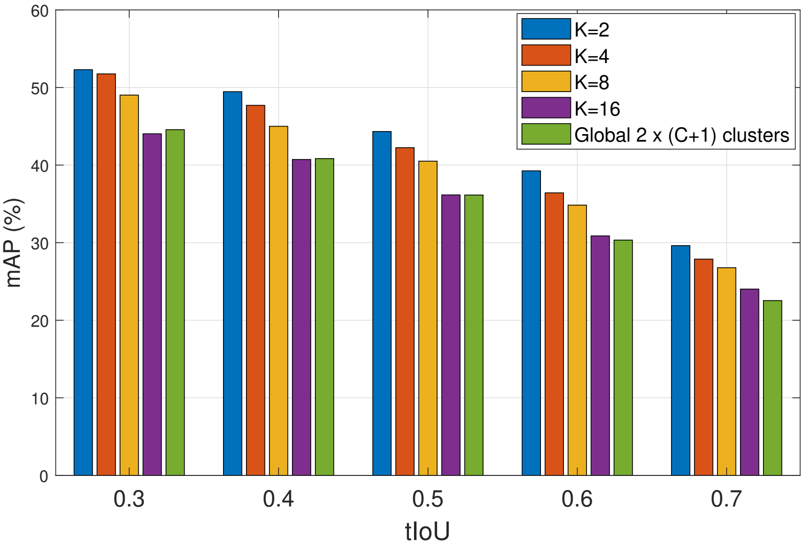


Figure 4: Evaluation of the effect of snippet-centroids on test videos of THUMOS-14 dataset.

at each time step *t*, proposals of multiple durations are obtained, which may overlap with each other. For the classification of obtained action proposal into one of the *C* classes, linear SVMs in one-vs-all fashion are used with cost=100, where cost is used as a parameter to smooth the decision boundary between data points of different classes. After classifying action proposals into *C* classes, non-maximal suppression (NMS) is performed to remove the overlapping proposals having an overlap ratio greater than 0.7 with each other, as done in other temporal action detection methods [4, 14] .

4.3. How snippet-centroids affect mAP?

This experiment is carried out to assess the effectiveness of using different snippet-centroids *K* extracted per-class on the system performance in terms of mAP.

From the training data, for each class *K* snippet-centroids are generated by using the class-specific k-means clustering. The effect on the method's performance of using different number of snippet-centroids per class is assessed with $K \in \{2, 4, 8, 16\}$. Figure 4 plots the mAP values achieved for a different number of snippet-centroids *K*. From Figure 4 it can be seen that there is a small difference between mAP values obtained for a different number of snippet-centroids. It can also be observed that by extracting only few snippet-centroids, e.g. $K = 2$, the proposed method achieves higher mAP value as compared to other values of *K*. A drop in mAP can be seen when snippet-centroids are increased from 2 to 16. This is because

Table 1: Impact of using two-stage vs. single-stage VTCS based temporal action detection framework on the mAP for Thumos14 dataset.

IoU threshold	0.3	0.4	0.5	0.6	0.7
VTCS-Single-stage framework	44.3	43.2	39.0	36.6	**30.2**
VTCS-Two-stage framework	**52.3**	**49.5**	**44.3**	**39.3**	30.0

by increasing the number of snippet-centroids, the distance between them will be smaller which causes incorrect assignment of snippets to these snippet-centroids.

An experiment is also performed to use global clustering instead of per-class clustering to extract snippet-centroids. Per-class clustering with $K = 2$ resulted in $2 \times (C + 1)$ clusters. Therefore, to have a fair comparison with the results obtained by per-class clustering using $K = 2$, a total of $2 \times (C + 1)$ clusters are extracted globally. Results in Figure 4 indicate a significant difference (of about 8% in mAP for tIoU thresholds from 0.3 to 0.7) between global clustering and per-class clustering with $K = 2$ although both have the same dimensions for the proposal representation. This reveals that VTCS performs best with per-class clustering because of equal contribution of each class during the encoding process. For both datasets, $K = 2$ will be used for the rest of experiments to extract per-class snippet-centroids as it achieves higher mAP.

4.4. Two-stage vs. single-stage framework

Section 3.3 pointed out that a two-stage temporal action detection framework is used for the classification of candidate proposals, where the first stage is used to filter the background proposals and the second stage is used to predict the class label of each proposal. This section reports an ablation study to see the impact of using only a single-stage to categorize the candidate proposals directly into $C + 1$ classes. For this, we directly train multi-class linear SVMs over VTCS representation of training data from $C + 1$ classes while skipping the FC network. Results in Table 1 show that the proposed two-stage frame performs better than a single-stage framework. It can be seen from Table 1 that the difference in mAP for both frameworks decreases from 0.3 to 0.6 tIoU thresholds. This is because, the background regions having less overlap with the ground truth regions may classify into action regions resulting in high false positives that will decrease the mAP values for lower tIoU thresholds. The two-stage framework filters out the background proposals in the first stage. Therefore, in the second stage the detected action proposals are categorized into

Table 2: Comparison of run-time in terms of FPS calculated for a video taken from Thumos14 dataset of duration 3 minutes (with 30FPS).

Methods	Framework	FPS
Sparse-prop [24]	Two-stage	10.2
SCNN [4]	Multi-stage	60.0
DAPS [14]	Two-stage	134.3
CDC [7]	Single-stage	500.0
TURN [8]	Two-stage	880.8
R-C3D [17]	Two-stage	1030.0
BoDS [3]	Single-stage	**1279.0**
VTCS (Ours)	Two-stage	1141.0

C action classes more accurately. For the rest of experiments, the two-stage framework based on VTCS representation will be used.

4.5. Run-time comparison

Table 2 reports the run-time comparison of the proposed VTCS based method with other methods [3, 4, 7, 8, 14, 17, 24]. The proposed VTCS based temporal action detection approach runs at 1141 Frames Per Second (FPS) on a single Nvidia TITAN Xp GPU. All methods given in Table 2 are based upon C3D based unit-level feature extraction except Sparse-prop [24] and CDC [7] which use Improved Dense Trajectories (IDT) [25] and CDC features respectively. Note that VTCS is faster than the other two-stage methods, e.g. it is about 1.3 times faster than TURN [8] and 1.1 times faster than R-C3D [17] which also use C3D features to represent video units. BoDS [3] was tested on an Nvidia TITAN Xp GPU and SCNN [4] on a GTX 980 GPU. DAPS [14], R-C3D [17], TURN [8] and CDC [7] were tested on a single Nvidia TITAN X GPU. SCNN [4], DAPS [14], R-C3D [17], CDC [7] and TURN [8] require multi-scale sliding windows during training stage, therefore these methods are computationally expensive. As the proposed VTCS-based approach has the ability to reuse the same visual encoding module for both stages (proposal generation and classification) and it can be trained with trimed positive and negative example videos, it is faster than other two-stage action detection methods. However, BoDS [3] is about 1.1 times faster than VTCS as it is based upon a single-stage temporal action detection framework. However, Section 4.6.1 points out that VTCS achieves high mAP value at multiple tIoU thresholds.

Table 3: Temporal action detection performance on Thumos14 dataset in terms of mAP (%) @ different tIoU thresholds. Results are sorted in ascending order at 0.5 tIoU where - indicates that results are not available.

IoU threshold	0.3	0.4	0.5	0.6	0.7
Sparse-prop [24]	-	-	13.5	-	-
DAPs [14]	-	-	13.9	-	-
SST [13]	-	-	13.9	-	-
FG [12]	36.0	-	17.1	-	-
PSDF [11]	33.6	26.2	18.8	-	-
SCNN [4]	36.3	28.7	19.0	10.3	5.3
CDC [7]	40.1	29.4	23.3	13.1	7.9
TURN [8]	44.1	34.9	25.6	-	-
TPN [26]	44.1	37.1	28.2	20.6	12.7
TAG [6]	48.7	39.8	28.2	-	-
R-C3D [17]	44.9	35.6	28.9	-	-
SS-TAD [15]	45.7	-	29.2	-	9.6
SSN [5]	51.9	41	29.8	-	-
CTAP [20]	-	-	29.9	-	-
ETP [27]	48.2	42.4	34.2	23.4	13.9
CBR [9]	50.1	41.3	31	19.1	9.9
TAL-Net [18]	53.2	48.5	42.8	33.8	20.8
BoDS [3]	**54.9**	47.2	41.5	37.5	**31.6**
VTCS [Ours]	52.3	**49.5**	**44.3**	**39.3**	30.0

4.6. Comparisons with the state-of-the-art

4.6.1. Thumos14 dataset

Table 3 provides comparative results between the proposed VTCS-based method with other temporal action detection methods for the Thumos14 dataset. In this case, the proposed method outperforms the state-of-the-art approaches including SCNN [4], Single-Stream Temporal Action Detection (SS-TAD) [15], PSDF [11], SST [13], TAG [6], action detection from Frame Glimpses (FG) [12], DAPs [14], SSN [5], CDC [7], TURN [8], CBR [9], Temporal Preservation Networks (TPN) [26], R-C3D [17], Evolving Temporal Proposals (ETP) [27], TAL-net [18], CTAP [20] and BoDS [3].

Similar to the proposed method, SS-TAD [15] and SST [13] also produce right-aligned times. However, these methods require multiple sliding windows to produce dense training data during the training stage. Instead, here only the ground truth annotations are used for training the SVM classifier, therefore the method proposed here is computationally efficient. SCNN [4], DAPS [14] and TURN [8] produce action proposals using multi-scale sliding windows, therefore these methods are computationally expensive. The state-of-

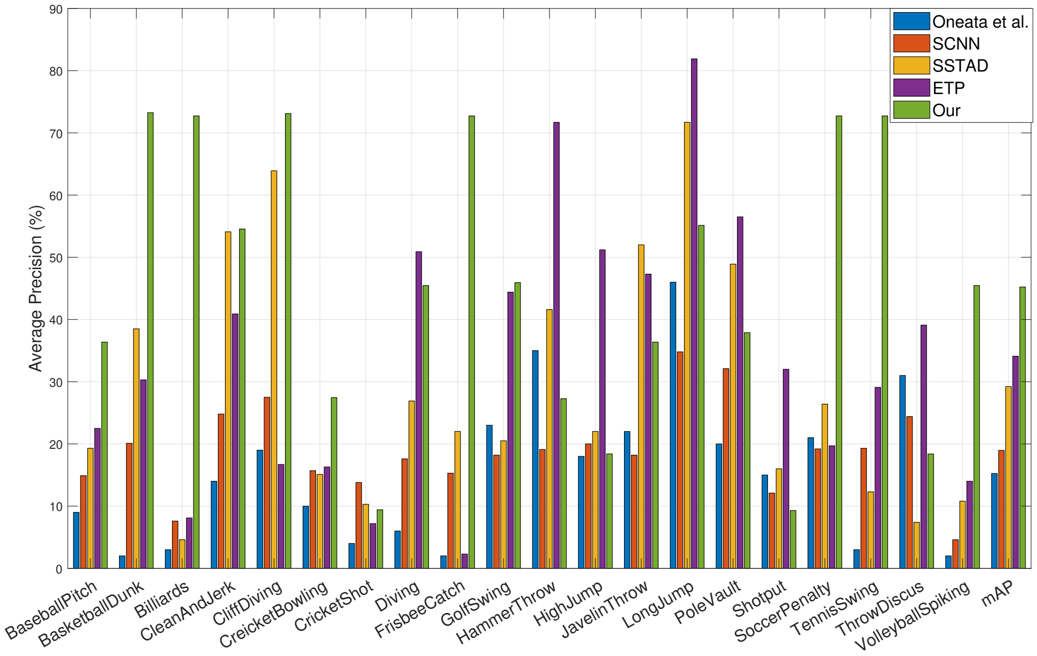


Figure 5: Comparison of per-class average precision (%) for Thumos14 dataset at 0.5 tIoU threshold.

the-art methods i.e. [18], [9], [27] and [5] require multiple networks for producing action proposals, for their refinement and for their classification tasks. However, a two-stage framework is proposed in this paper, in which both stages share the same encoded features which makes it computationally fast.

The proposed VTCS based temporal action detection approach respectively achieves 2.3%, 2.8% and 1.8% gain in mAP at 0.4, 0.5 and 0.6 tIoU threshold than BoDS [3]. The significant improvement in mAP over the previous methods [4–9, 11–15, 17, 18, 20, 26, 27] for high tIoU thresholds and comparable results with [3] shows that VTCS can detect actions from untrimmed videos more precisely. The average precision of each action class is also provided in Figure 5. It can be observed that the proposed VTCS based temporal action detection achieves better average precision for 11 out of 20 action classes with large margins as compared to other methods and mAP for the Thumos14 dataset, higher than other methods.

Average number of retrieved proposals. For the Thumos14 dataset, the performance of the proposed method is assessed in terms of mAP by extracting different numbers of action proposals per video at tIoU threshold equal to 0.5. Table 4 presents a comparison with other state-of-the-art methods for the Thumos14 dataset at 0.5 tIoU overlaps, showing that the proposed method attains better performance compared to other action

Table 4: Temporal action detection performance in terms of mAP (%) at different number of proposals using 0.5 tIoU threshold for Thumos14.

Methods	@50	@100	@200	@500	@1000
Sparse-prop [24]	5.7	6.3	7.6	8.2	8
SCNN [4]	5.6	7.7	10.5	13.5	13.5
DAPs [14]	8.4	12.1	13.9	12.5	12.0
SST [13]	10.9	13.2	13.9	13.1	12.0
TURN [8]	-	-	-	-	25.6
VTCS [Ours]	**19.1**	**29.2**	**36.5**	**42.9**	**44.3**

Table 5: Temporal action detection performance on ActivityNet-v1.1 dataset in terms of mAP (%) @ different tIoU thresholds. Results are sorted in ascending order at 0.5 tIoU where - indicates that results are not available.

IoU threshold	0.3	0.4	0.5	0.6	0.7
[23]	-	-	33.2	-	-
FG [12]	-	-	36.7	-	-
TURN [8]	-	-	37.1	-	-
BoDS [3]	**51.1**	45.0	38.1	**34.2**	**29.0**
VTCS [Ours]	50.7	**45.0**	**38.5**	32.0	28.1

proposals obtained by computationally expensive methods such as SCNN [4] and Sparse-prop [24]. Furthermore, the results also show that the approach provides a satisfactory mAP using only a few number of temporal proposals.

4.6.2. *ActivityNet dataset*

For the Sports subset of ActivityNet-v1.1 dataset, the proposed VTCS based method is compared with [23], FG [12], TURN [8] and BoDS [3] as shown in Table 5. The Sports subset contains 21 actions named: ‘Archery’, ‘Bowling’, ‘Bungee’, ‘jumping’, ‘Clean and jerk’, ‘Cricket’, ‘Curling’, ‘Discus throw’, ‘Dodgeball’, ‘Doing motocross’, ‘Hammer throw’, ‘High jump’, ‘Javelin throw’, ‘Long jump’, ‘Paintball’, ‘Playing kickball’, ‘Pole vault’, ‘Shot put’, ‘Skateboarding’, ‘Starting a campfire’, ‘Triple jump’ and ‘Volleyball’. Table 5 reports the mAP results of the proposed method at different tIoU thresholds while the methods proposed in [12, 8, 23] only provide results at 0.5 tIoU threshold. It can be seen that the proposed VTCS method resulted in higher mAP value at tIoU thresholds of 0.4 and 0.5 and comparable results at other tIoU thresholds. For this dataset, the number of per-class snippet-centroids, $K = 2$ was adopted, as for Thumos14, illustrating the generalization capability of VTCS to another action detection dataset as well.

Table 6: Comparison results in terms of mAP(%) for five subsets of ActivityNet-v1.3. Numbers in brackets shows the total number of actions present in each subset.

tIoU threshold	0.3		0.4		0.5		0.6		0.7	
Subset	BoDS	VTCS	BoDS	VTCS	BoDS	VTCS	BoDS	VTCS	BoDS	VTCS
Household (45)	**23.6**	21.8	**18.2**	16.6	**15.2**	13.3	11.7	**11.8**	10.1	**10.3**
Personal care (19)	14.4	**16.2**	12.0	**13.8**	8.0	**10.9**	6.7	**8.9**	5.7	**7.5**
Eating and drinking (11)	17.4	**18.8**	13.2	**13.6**	12.0	**11.2**	6.6	**7.9**	5.0	**7.0**
Socializing and leisure (37)	35.6	**35.9**	30.0	**30.0**	**26.0**	25.6	22.6	**22.7**	18.9	**19.0**
Sports and exercises (88)	49.6	**49.9**	**43.5**	43.2	37.4	**37.6**	32.6	**32.7**	28.1	**28.3**
Average	28.1	**28.5**	23.3	**23.4**	19.7	**19.7**	16.4	**16.8**	13.6	**14.4**

Subsets of ActivityNet. Table 6 reports comparative results between the proposed VTCS based method with BoDS [3] for the 5 top-level subsets: 'household','personal care','eating and drinking','socializing and leisure' and 'sports and exercises' of the ActivityNet-v1.3 dataset. From the results of different subsets, it can be seen that the 'sports and exercises' attains the highest mAP because the activities present in this subset have well-defined temporal ordering. Other subsets, e.g. 'personal care','eating and drinking' attains less mAP values as compared to 'sports and exercises' because the activites present in these subsets do not have well-defined temporal ordering and have an unstructured nature. From Table 6, it can be seen that the proposed VTCS based method achieves higher mAP as compared to BoDS [3] at different tIoU thresholds.

4.7. *Qualitative Results*

Figure 6 shows qualitative results for three untrimmed videos from the Thumos14 dataset. Figure 6(a), shows the detected vs. ground truth regions for an untrimmed video containing three occurrences of the 'Billiards' action. It can be seen that all occurrences of Billiards action are accurately detected with correct class label with tIoU equal to ≥ 0.7 with the ground truth regions. Figure 6(b) also provides qualitative results for the example video containing multiple instances of 'Long Jump' action. The proposed VTCS based method failed to detect correct action labels therefore it is considered as negative detection. Although VTCS detects the regions with tIoU equal to 0.88 for both instances, it fails to detect the correct class label because, in a single action instance, viewpoint changes for multiple times. Figure 6(c) provides an example for 'Volleyball Spiking' action, where the method predicted the correct class label, but it fails to detect precise boundaries of action. The method was not able to correctly detect the start-end time because this

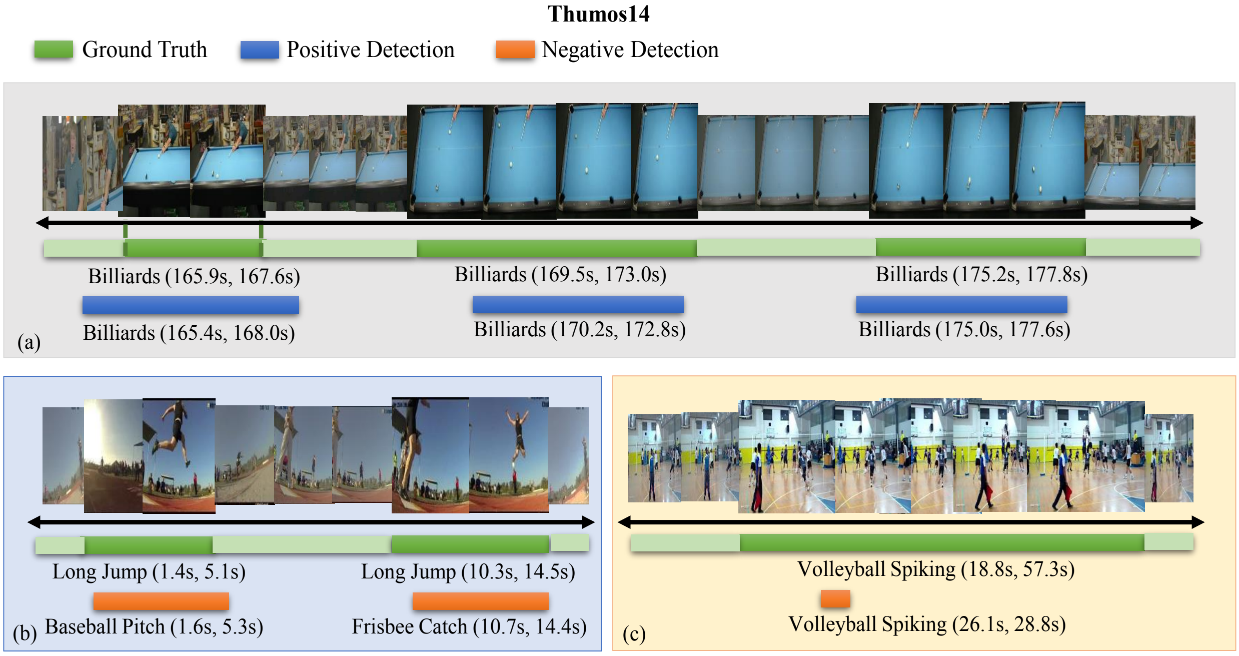

Thumos14
Ground Truth
Positive Detection
Negative Detection
Billiards (165.9s, 167.6s)
Billiards (169.5s, 173.0s)
Billiards (175.2s, 177.8s)
Billiards (165.4s, 168.0s)
Billiards (170.2s, 172.8s)
Billiards (175.0s, 177.6s)
(a)
Long Jump (1.4s, 5.1s)
Long Jump (10.3s, 14.5s)
Baseball Pitch (1.6s, 5.3s)
Frisbee Catch (10.7s, 14.4s)
(b)
Volleyball Spiking (18.8s, 57.3s)
Volleyball Spiking (26.1s, 28.8s)
(c)

Figure 6: Qualitative results for the example test videos from Thumos14 dataset. Action labels for each action instance are shown and their corresponding start-end times (in seconds) are shown inside brackets. If the tIoU of the detected region with any of the ground truth region is ≥ 0.5 and if it is classified into the correct action class, then it is considered as a true positive (best seen in color).

video has a very slow transition.

Although the proposed method obtained improved performance as compared to other methods, it has some limitations. 1) The dimension of VTCS features is $K \times (C+1)$, i.e. it depends on the number of classes. Therefore, for large number of classes VTCS will have large dimensions hence it will be computationally expensive. 2) It requires a separate stage for proposal generation due to which it cannot operate in an end-to-end manner to detect the action of interest from the untrimmed videos. Nevertheless, Table 1 presents a single-stage version of the proposed VTCS, and it can be seen that the two-stage VTCS framework obtained a higher mAP compared to single-stage VTCS. In future work we intend to address these limitations by seeking ways of making the dimension of VTCS features independent of the number of classes and by using single-stage action detection.

5. Conclusion

In this work, we have proposed a new encoding scheme, VTCS, which extracts the discriminative representation for non-overlapping snippets of the untrimmed videos. The proposed approach generates few snippet-centroids from each class which is used for the encoding process. It uses the same encoding representation for generating the initial proposals which are then passed to SVM classifiers already learnt at positive classes. This two-stage temporal action detection framework can run on long untrimmed videos at high processing speed. Compared to other methods, it does not require sliding windows during the training stage, as it only utilizes the annotated actions from positive as well as background classes which makes it computationally more efficient. Experiments show that the proposed two-stage VTCS based approach attains improved performance for the two datasets used in the experiments and that it outperforms other top methods.

We plan to use this framework in other video understanding tasks, such as anomaly detection in crowd scenes, video summarization and sports video analysis and to extend the method to situations with simultaneously different actions that might need to be first localised in space and then treated separately.

Acknowledgements

We gratefully acknowledge the funding for Swarm Robotics Lab under National Centre for Robotics and Automation, Pakistan. We also acknowledge the support of NVIDIA Corporation with the donation of the TITAN Xp GPU used for this research. Sergio A Velastin acknowledges funding by the Universidad Carlos III de Madrid, the European Unions Seventh Framework Programme for research, technological development and demonstration under grant agreement no. 600371, el Ministerio de Econom´a y Competitividad (COFUND2013-51509) and Banco Santander.

Data Statement

The datasets used for the evaluation of the proposed framework are from previously reported studies, which have been cited.

References

[1] G. Cheng, Y. Wan, A. N. Saudagar, K. Namuduri, B. P. Buckles, Advances in human action recognition: A survey, arXiv preprint arXiv:1501.05964.

[2] F. Murtaza, M. HaroonYousaf, S. A. Velastin, Da-vlad: Discriminative action vector of locally aggregated descriptors for action recognition, in: 2018 25th IEEE International Conference on Image Processing (ICIP), IEEE, 2018, pp. 3993–3997.

[3] F. Murtaza, M. H. Yousaf, S. A. Velastin, Y. Qian, End-to-end temporal action detection using bag of discriminant snippets, IEEE Signal Processing Letters 26 (2) (2019) 272–276.

[4] Z. Shou, D. Wang, S.-F. Chang, Temporal action localization in untrimmed videos via multi-stage cnns, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 1049–1058.

[5] Y. Zhao, Y. Xiong, L. Wang, Z. Wu, X. Tang, D. Lin, Temporal action detection with structured segment networks, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 2914–2923.

[6] Y. Xiong, Y. Zhao, L. Wang, D. Lin, X. Tang, A pursuit of temporal accuracy in general activity detection, arXiv preprint arXiv:1703.02716.

[7] Z. Shou, J. Chan, A. Zareian, K. Miyazawa, S.-F. Chang, Cdc: Convolutional-de-convolutional networks for precise temporal action localization in untrimmed videos, in: Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on, IEEE, 2017, pp. 1417–1426.

[8] J. Gao, Z. Yang, K. Chen, C. Sun, R. Nevatia, Turn tap: Temporal unit regression network for temporal action proposals, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 3628–3636.

[9] J. Gao, Z. Yang, R. Nevatia, Cascaded boundary regression for temporal action detection, arXiv preprint arXiv:1705.01180.

[10] D. Tran, L. Bourdev, R. Fergus, L. Torresani, M. Paluri, Learning spatiotemporal features with 3d convolutional networks, in: Computer Vision (ICCV), 2015 IEEE International Conference on, IEEE, 2015, pp. 4489–4497.

[11] J. Yuan, B. Ni, X. Yang, A. A. Kassim, Temporal action localization with pyramid of score distribution features, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 3093–3102.

[12] S. Yeung, O. Russakovsky, G. Mori, L. Fei-Fei, End-to-end learning of action detection from frame glimpses in videos, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 2678–2687.

[13] S. Buch, V. Escorcia, C. Shen, B. Ghanem, J. C. Niebles, Sst: Single-stream temporal action proposals, in: Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on, IEEE, 2017, pp. 6373–6382.

[14] V. Escorcia, F. C. Heilbron, J. C. Niebles, B. Ghanem, Daps: Deep action proposals for action understanding, in: European Conference on Computer Vision, Springer, 2016, pp. 768–784.

[15] S. Buch, V. Escorcia, B. Ghanem, L. Fei-Fei, J. Niebles, End-to-end, single-stream temporal action detection in untrimmed videos, in: Proceedings of the British Machine Vision Conference (BMVC), 2017.

[16] F. Murtaza, M. H. Yousaf, S. A. Velastin, Pmhi: Proposals from motion history images for temporal segmentation of long uncut videos, IEEE Signal Processing Letters 25 (2) (2018) 179–183.

[17] H. Xu, A. Das, K. Saenko, R-c3d: region convolutional 3d network for temporal activity detection, in: IEEE Int. Conf. on

Computer Vision (ICCV), 2017, pp. 5794–5803.

[18] Y.-W. Chao, S. Vijayanarasimhan, B. Seybold, D. A. Ross, J. Deng, R. Sukthankar, Rethinking the faster r-cnn architecture for temporal action localization, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 1130–1139.

[19] S. Ren, K. He, R. Girshick, J. Sun, Faster r-cnn: Towards real-time object detection with region proposal networks, in: Advances in neural information processing systems, 2015, pp. 91–99.

[20] J. Gao, K. Chen, R. Nevatia, Ctap: Complementary temporal action proposal generation, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 68–83.

[21] F. Perronnin, J. Sánchez, T. Mensink, Improving the fisher kernel for large-scale image classification, Computer Vision–ECCV 2010 (2010) 143–156.

[22] Y. Jiang, J. Liu, A. Roshan Zamir, G. Toderici, I. Laptev, M. Shah, R. Sukthankar, Thumos challenge: action recognition with a large number of classes (2014) (2014).

[23] F. Caba Heilbron, V. Escorcia, B. Ghanem, J. Carlos Niebles, Activitynet: A large-scale video benchmark for human activity understanding, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 961–970.

[24] F. Caba Heilbron, J. Carlos Niebles, B. Ghanem, Fast temporal activity proposals for efficient detection of human actions in untrimmed videos, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 1914–1923.

[25] H. Wang, C. Schmid, Action recognition with improved trajectories, in: Computer Vision (ICCV), 2013 IEEE International Conference on, IEEE, 2013, pp. 3551–3558.

[26] K. Yang, P. Qiao, D. Li, S. Lv, Y. Dou, Exploring temporal preservation networks for precise temporal action localization, in: Thirty-Second AAAI Conference on Artificial Intelligence, 2018.

[27] H. Qiu, Y. Zheng, H. Ye, Y. Lu, F. Wang, L. He, Precise temporal action localization by evolving temporal proposals, in: Proceedings of the 2018 ACM on International Conference on Multimedia Retrieval, ACM, 2018, pp. 388–396.

Authors Biography

Fiza Murtaza is currently serving as an Assistant Professor with the Department of Computer Science, Women University Swabi, Pakistan. She received her Ph.D. in Computer Engineering from University of Engineering and Technology, Taxila, Pakistan in October 2019. Her research interests focus on Computer Vision, and machine learning, especially in Temporal Human Action detection.

Muhammad Haroon Yousaf is currently serving as an Associate Professor in Computer Engineering Department, University of Engineering and Technology Taxila, Pakistan. His research interests are image processing/computer vision. He is heading Swarm Robotics Lab under National Centre for Robotics and Automation. He has been the recipient of Best University Teacher Award (2012-2013) given by Higher

Education Commission (HEC) of Pakistan.

Sergio A Velastin has worked in industrial R&D and also at Kings College London (UK) and Kingston University London becoming a full professor of applied computer vision. He is a Fellow of the IET and currently Senior Research Scientist at Zebra Technologies Corp. and a visiting professor at Universidad Carlos III in Madrid and at Queen Mary University of London.

Yu Qian has educational background in electrical and electronics engineering and computer science with a bachelors and masters from Hefei University of Technology, China and Ph.D. from the school Computer Science of Middlesex University. She is currently working in Zebra Technologies Corp. as senior research scientist and used to work in MTRC University of Bath and CVSSP University of Surrey.