TenniSet: A Dataset for Dense Fine-Grained Event Recognition, Localisation and Description

Hayden Faulkner
School of Computer Science
The University of Adelaide
Adelaide, South Australia
Email: hayden.faulkner@adelaide.edu.au

Anthony Dick
School of Computer Science
The University of Adelaide
Adelaide, South Australia
Email: anthony.dick@adelaide.edu.au

***Abstract*—This paper introduces a new video understanding dataset which can be utilised for the related problems of event recognition, localisation and description in video. Our dataset consists of dense, well structured event annotations in untrimmed video of tennis matches. We also include highly detailed commentary style descriptions, which are heavily dependent on both the occurrence as well as the sequence of particular events. We use general deep learning techniques to acquire some initial baseline results on our dataset, without the need for explicit domain-specific assumptions.**

I. Introduction

Video understanding includes several important problems in computer vision research, such as the detection and recognition of events, and video description or captioning. Following trends in image understanding, video understanding problems have recently been advanced with the use of deep learning approaches which are inherently data driven. However, compared to image based problems, progress has been slow, mostly due to the difficultly in collecting, annotating and processing video data.

As these deep approaches get more complex, the requirement for video data increases. Furthermore as the performance of these approaches improve, researchers are beginning to cover multiple facets of the video understanding problem with unified or very similar frameworks. Motivated by this research direction we introduce a tennis dataset which is able to be utilised for event recognition, localisation and description within videos (Figure 1).

Most current description methods describe videos by recognising the objects that are present, presenting only a vague summary of the action that is taking place. This has several limitations. Firstly, for many types of video, such as sports and surveillance, much of the salient information lies in the *fine-grained* detail of the action that is depicted. Secondly, in the case of sports and in many other domains too, it is much more useful to describe a video with domain related context and knowledge. Both of these characteristics are missing from recent video description datasets. Our dataset contains 746 commentary style descriptions which are not only heavily reliant on the particular actions and events that take place, but also the order in which they take place.

In accordance with our detailed descriptions, our dataset also contains over $4,000$ event annotations to frame-level precision. These events enable our dataset to be utilised for action / event recognition and detection evaluations. Our annotations are at multiple levels of conceptual and temporal abstraction, from entire games to individual hits, making our dataset useful for higher level event modelling. Also, compared to current datasets for action / event recognition and detection our lowest level events are not only short ($\tilde{3}0$ frames), but also dense, with less than 5 frames separating most events. This requires online models to be able to make decisions about their current event state very quickly.

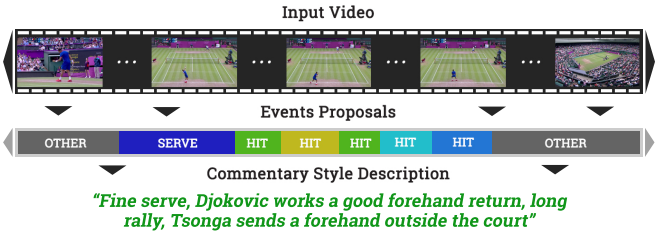

Fig. 1. Our dataset is annotated with temporally dense events based on tennis actions, as well as contextually relevant event based descriptions.

We utilise some recent general deep learning techniques to form baselines for our dataset. Briefly, we employ Convolutional Neural Networks (CNN) for visual processing and Recurrent Neural Networks (RNN) temporal and sequence modelling. We show that these modern techniques achieve impressive results on our dataset, but believe further improvement should be possible by focusing more on small but highly influential spatial and temporal regions within the video.

II. Related Work

We review works related to action recognition and detection, as well as those related to video description. We also review some current datasets related to these problems and highlight the necessity of our dataset.

A. Action Recognition & Video Classification

The problem of action recognition is to classify a trimmed video clip into one of a pre-determined set of classes. Although this problem has a long research history, only very recently has deep learning been applied to solve it. Early approaches involved extracting video descriptors using hand-crafted features [15], [22], [55]–[57]. Early datasets such as KTH [41], Weizmann [13], Hollywood-2 [25], HMDB [16], [22] and UCF-101

[48] were focused on short clip classification, where a clip was trimmed around a single action. The ACT dataset [59] follows the same premise, however it introduces hierarchical super-classes.

More recent methods have begun to utilise deep data driven architectures. [17] introduce 3D CNNs by extending 2D convolutional operators into the time dimension, allowing the network to capture motion information encoded in multiple adjacent frames. [20] experiment with 3D CNNs and different styles of temporal fusion. [51] introduce C3D, a much deeper 3D net than that of [17], [20]. Distinct from 3D CNNs, [10], [45] introduce a two-stream approach where a raw frame CNN and an optical flow CNN are trained separately and fused together. [28] explore late-style pooling techniques, and also an RNN for learning longer range temporal dependencies.

Along with rise of data driven deep learning approaches and with the aid of online video services such as YouTube, datasets have become much larger both in terms of number of classes as well as number of samples. These large sets, Sports-1M [20], YouTube-8M [1] and Kinectis [21] have focused more on video classification, where actions may be relevant to class decisions, but the classes aren't explicitly actions. Also, forming such large datasets is difficult and with some annotations being automatic, annotations are noisier than is usual for image datasets.

B. Action Detection & Localisation

The task of event (or action) recognition involves finding and classifying an event in an untrimmed video clip which can contain many different events. Initial approaches [19], [27], [29], [43], [58] employed an exhaustive sliding window approach to generate fixed clips, which were then treated as trimmed clips. Similar to the earlier action recognition works [19], [58], [29] utilise hand crafted and CNN features for encoding a window. [27] average out probabilities across sub-slips encoded with a 3D CNN and RNN over sub-clips across a video. [43] take a more complex approach adopting 3D CNNs in a three stage proposal, classification and localisation framework. [9], [24], [47], [64] replace the exhaustive sliding window step with RNNs which label every time step. Most use regression to locate boundary proposals at individual points in time.

Action detection datasets have events marked up in untrimmed videos. ActivityNet [3], THUMOS [12], [18] and MultiTHUMOS [63] use videos sourced from YouTube, AVA [14] uses movie clips, while Charades [44] uses crowdsourced actors. The aforementioned datasets contain actions across visually different scenes, allowing classifiers to use factors other than the explicit actions to make decisions. This contrasts with *fine-grained* datasets, MPII Cooking [38] and Cooking 2 [39] which focus on cooking videos, Basketball [34] which uses broadcast footage of basketball games, and MERL [47] which uses a static overhead camera to record what people shop for from a set of shelves. Our dataset is in the same vein as these fine-grained sets, with different actions occurring in the same scene with the same objects at different times.

C. Video Captioning & Description

The task of event description is to generate a sentence, describing and summarising an event, or set of events. Compared with the aforementioned problems, video captioning is a relatively young problem. [54] combine a CNN with an RNN to process framewise image data from a video and generate a description. This was later extended [53], [60] by including optical flow, multi-scale features, and a sequence-to-sequence RNN model to capture temporal patterns more effectively. [62] utilise a 3D CNN to capture short term temporal relations. [30] jointly learn both 2D and 3D CNNs which are mean pooled over time before an RNN is used for sentence generation. Alternative video features such as dense trajectories, object detectors, and scene CNNs have been used to help find key concepts such as verbs, objects and scenes [65]. [42] use frame and video level features, before using an RNN for sentence generation, while [32] uses bi-directional RNNs.

With the exception of the Charades dataset which includes multiple descriptions per clip, video description datasets have been disjoint from action based datasets. Description datasets TACoS [35] and TACoS M-L [36] use the video data from MPII Cooking 2, however they aren't brought together as one dataset. The most widely used video description dataset, MSVD [4], contains general videos from YouTube with multiple descriptions per clip marked up by AMT workers. Recently, MSR-VTT [61] also uses YouTube but is over 4 times the size of MSVD. MPII-MD [37] and M-VAD [50] use movie clips and use Descriptive Video Services (DVS) data for description annotation. In comparison to our dataset, these datasets use videos where descriptions are heavily based on scene contents rather than action sequences.

III. The Tennis Dataset

Tennis data is ideal for the problems of action recognition, detection, and description for a number of reasons. Firstly this is one of the few datasets which can be considered fine-grained, where different actions occur in the same scene at a fine level of detail. Secondly, the actions are short and dense in time, with event boundaries often occurring only a few frames apart, meaning the model has a very small window of opportunity to make a decision. Thirdly, most description datasets focus on general scenes and therefore have very *noun* driven descriptions which lack the detail we aim to recover. Lastly, compared to many event occurrences in the real world, the game of tennis has a very set structure and sequence of events, giving our dataset potential to be used for higher order logic models related to event sequence modelling and prediction. We also believe this is the first dataset with these properties to cover all three problems of event recognition, detection and description.

A. Outline

Following the lead of [49], we form a tennis dataset consisting of five singles matches from the 2012 London Olympics. Obtained from YouTube (youtube.com) each video is 1280×720 at 25 fps. Each match is marked up with a number of sequences relating to particular tennis related events, each event belongs to a particular event type and has its own particular attributes. In our case we utilise six event types shown in Table I.

The `Serve` and `Hit` types are the finest level classes representing specific actions, with `Point` events containing

one or more Serve and Hit events, Game events containing multiple Point events, Set events containing multiple Game events, and Match events containing multiple Set events. This hierarchical structure of more refined actions being subsets of higher level events is similar to the real world where short-term actions are just parts of a longer-term event or goal. Compared to the real world however, the structure and sequence of events in a tennis match are much more constrained and structured.

TABLE I. DATASET EVENT STATISTICS FOR DIFFERENT EVENT TYPES AND ATTRIBUTES

Type	Attributes	Events	Frames	Frames/Event
Match	Winner	5	786455	157291
Set	Winner & Score	11	765738	69613
Game	Winner & Score & Server	118	588759	4989
Point	Winner & Score	746	159494	214
Serve	Near|Far & In|Fault|Let	1017	68385	67
	Near & In	345	22920	66
	Near & Let	12	817	68
	Near & Fault	128	8902	70
	Far & In	382	25205	66
	Far & Let	26	1777	68
	Far & Fault	124	8764	71
Hit	Near|Far & Left|Right	2551	73564	29
	Near & Left	670	18083	27
	Near & Right	625	17790	28
	Far & Left	600	18587	31
	Far & Right	656	19104	29

B. Splits

We manually split the dataset into training, validation, and test sets (Table II). We split videos between Game events to ensure all Hit, Serve, Point and Game events fall exclusively into either the training, validation or testing splits. We split in this fashion to allow the same training data used on a frame, or lower hierarchical event type, basis to be used in a higher event type, *ie.* frames within sub-events from a Point in the training set will also fall in the training set for frames.

TABLE II. DATASET SPLIT STATISTICS

Event	train		val		test	
	#	%	#	%	#	%
Game	85	72	8	7	25	21
Point	550	74	42	6	154	20
Serve	750	74	57	6	210	20
Hit	1868	74	140	5	543	21
Frames	571280	73	44463	6	171317	21

C. Action Classes & Set Balancing

To address the problems of action recognition and detection we employ the following classes across temporal space: Hit Near Right (HNR), Hit Near Left (HNL), Hit Far Right (HFR), Hit Far Left (HFL), Serve Near (SN), Serve Far (SF), and Other (O). These categories reflect the hit and serve event types labelled in the dataset. Serves, no matter whether they are faults, lets, or land in, are labelled as a serve event. The "Other" (O) label is used when the frame or clip represents none of the other six classes.

Within the dataset most frames and clips belong to the category 'Other' (O), so to prevent bias in training, we randomly under-sample to achieve approximately equal numbers for each category. We chose random under-sampling for its simplicity and speed-up factors in model training and testing. This sampling results in approximately 13k frames and 350 clips per class for training, and 950 frames and 30 clips per class for validation. We use all samples for testing however we disregard the class O in our metric calculations.

We annotate actions to begin when a player starts their upswing for a serve, or backswing for a hit, and finish when the follow-through of the racquet is complete. For the event detection setting, events can be of any length, however for event recognition we use clips of a constant 25 frames centred on the middle frame of the untrimmed event.

D. Image Preprocessing

We crop the 1280×720 frames to 1200×700 centrally to maximise the court area, and then resize them to a square 512×512 image. We perform mean subtraction where the mean is calculated across the training set. Similar to past works [8], [11], [53] we also process frames adjacent frames into optical flow frames. That is, we first generate traditional flow features, and then stack the x, y, and magnitude into a three dimensional image.

E. Descriptions

Similar to [49], for Point events we also obtain one sentence of commentary describing the point scraped from the web (tennisearth.com). We parse the commentary to remove player names, replacing with them with np and fp for near and far player respectively. The near and far is based on the viewpoint from the main camera which looks over the court from one end. We also replace forehand and backhand with ls or rs, for left or right side respectively, overcoming the 'handedness' of a particular player. Again, left and right is based on the viewpoint from the main camera, and is based on whether the shot is on the left or right side of the hitting players body (not a players court position).

Upon inspection of the raw commentary data we find that 582 out of the 746 scraped descriptions (78%) are either missing or incorrect, so we revise them to match the video. For example, in one case the original commentary reads *"High kick serve, fp returns a ls return, short rally, np cross-court ls lands out-side the court"*, where the video shows a double fault, and hence is altered to read *"Double Fault"*.We will make our revised commentary available upon acceptance so others can verify our alterations are indeed correct. Post-correction we end up with one description for each Point event, with an average sentence length of 15 words, and a vocabulary size of 223.

IV. EXPERIMENTS

We implement some standard deep learning methodologies to generate some baselines for our new dataset for the tasks of event recognition, event localisation, and event description. An overview of the three pipelines can be seen in Figure 2, which will be described in further detail in the following sections.

Similar to our work, [49] attempted to perform video description on the same data as that seen in our dataset. They detect court and player positions using dense trajectories and learn phrase classifier SVMs. The phrase predictions are smoothed across frames with a MRF model and then lexically matched to a commentary line. A few other works also address tennis video understanding such as retrieval [26], annotation [7], and ball motion statistics [33]. All of these works use domain specific processes whereas we employ a very general purpose neural network pipeline.

A. Frame Classification

We utilise a 2D CNN for processing individual frames, particularly we use the VGG16 network [46] as a base network for all of our approaches. We train the network for frame classification, with no temporal modelling. This gives us a starting baseline against which to measure the effect of temporal modelling, and also gives us a trained CNN which we can use as a feature extractor for more complicated pipelines. Following other work in the area of classification, action recognition, and action detection we measure performance with class-wise average precision (AP), and cross-class mean average precision (mAP). We exclude the AP of the class `Other` (`O`) in our mAP results as we consider it background.

1) Input Size: Unlike image and video classification, including action recognition, where the objects of interest occupy a large proportion of the frame, for tennis and many other video applications this is not the case. We therefore experimented with different input frame sizes to test the ability of the CNN to operate with decreasing amounts of discriminative information. We found that using the 224×224 input size expected by the VGG network performed similarly with other sizes 128 and 512.

2) Dense Size: Unlike ImageNet [40], which the VGG networks were designed for, our problem only consists of 7 classes rather than 1000. Also, most ($\sim 90\%$) of the parameters of the original VGG models are in the two dense layers of the networks. For these reasons we experiment with decreasing the dimensions of the dense layers. We found decreasing the number of dense neurons to 256 and only having 1 dense layer rather than 2 had no negative effects on the models classification performance. We therefore use these values for our experiments.

3) Optical Flow: Our dataset contains *fine-grained* actions which take place in very localised spatial and temporal areas, so we expect low level motion information to be important. So like other works [10], [45], [53] we experiment with optical flow as an input into the CNN. We input optical flow as (1) an individual input instead of RGB; and (2) as a combined input into a separate optical flow and RGB CNNs akin to the two-stream approach of [45].

Table III presents the mAP results of our framewise CNN pipelines, with sole RGB, sole optical flow and the two-stream model merged at different layers. We find that the sole optical flow CNN outperforms its RGB counterpart highlighting the importance of low level motion in our dataset. However, the two-stream approaches attain the highest performances, with merging at the last layer (`fc1`) being the most beneficial.

TABLE III. FRAME CLASSIFICATION: PER-FRAME MAP OF RGB V FLOW V TWO-STREAM CNN MODELS

CNN Model	Merged At	mAP
RGB Only		0.6748
Optical Flow Only		0.7607
Two-Stream	`pool4`	0.7978
Two-Stream	`pool5`	0.8134
Two-Stream	`fc1`	**0.8157**

B. Event / Action Recognition

The task of event (or action) recognition involves classifying a trimmed clip into one class. The simplest extension of our network to perform clip classification is to simply mean pool the frame-wise classification scores from the Softmax (`sm`) layer.

A different approach for capturing temporal information is to use an RNN, which is known for capturing longer-term dependencies. The RNN simply uses activations from a particular layer in the CNN as inputs, we use either the last pooling layer (`pool5`) or the last fully connected layer (`fc1`). Table IV shows our results for event recognition from trimmed clips. These results show that: 1) Simple mean pooling is very effective, we believe this is because the CNN has high mAP without temporal information; 2) The two-stream architecture doesn't provide the same performance boost that it did for frame classification, which is likely a result of such high mAP where better performance is extremely difficult; 3) Using a layer which still contains spatial information such as the last pooling layer (`pool5`) performs better than an input without any spatial structure (`fc1`); and 4) Using a Bi-directional RNN (Bi-RNN) is more effective than a forward direction (One-way) RNN.

TABLE IV. EVENT RECOGNITION: CLIP MAP USING TEMPORAL MEAN POOLING OR RNN

CNN Model	Input Layer	mAP
Mean Pooling		
Sole RGB	`sm`	0.9533
Sole Optical Flow	`sm`	0.9676
Two-Stream	`sm`	**0.9798**
One-way RNN		
Sole RGB	`pool5`	0.9436
	`fc1`	0.9095
Sole Optical Flow	`pool5`	**0.9602**
	`fc1`	0.9288
Two-Stream	`fc1`	0.9412
Bi-directional RNN		
Sole RGB	`pool5`	**0.9797**
	`fc1`	0.9528
Sole Optical Flow	`pool5`	0.9787
	`fc1`	0.9732
Two-Stream	`fc1`	0.9705

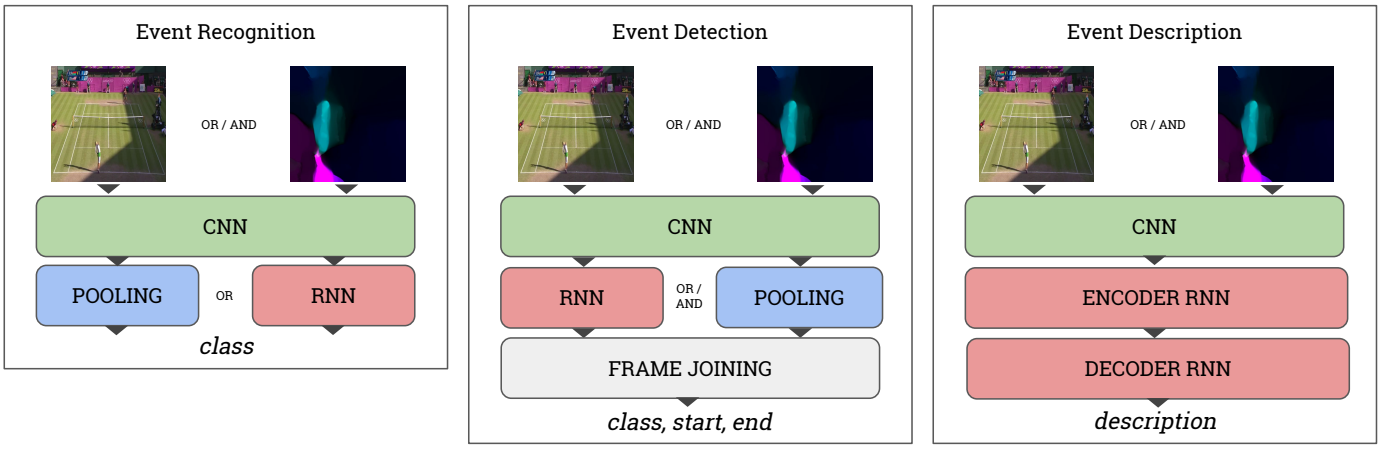

Fig. 2. Overview of the three pipelines for event recognition, detection and description. All of our pipelines rely on a CNN framework trained on individual frames. We use RNN and mean pooling to get clip classifications in event recognition, and to get frame classifications which are then joined for event detection. We use an RNN encoder-decoder framework for event description.

C. Event / Action Detection

The task of event (or action) detection involves detecting and classifying an event in an untrimmed clip. Since we aren't explicitly trying to find events of the class `Other`, but rather finding instances of all of the non-`Other` classes, we disregard the `Other` classes' AP in our cross-class mAP evaluations.

Using our CNN's frame classifications as a base, the simplest way to generate action proposals is to join adjacent frames of the same class into an event. However as our CNN isn't perfectly accurate, a few noisy frames within an event will lead to many short events being proposed rather than one longer event. As seen for event recognition in Section IV-B, including temporal information using pooling or an RNN can improve the performance by removing much of the noise. We therefore utilise a temporal sliding window which performs mean pooling across all frames within the window to get a result at a single time point t. Deciding on window boundaries is an important consideration, for a particular frame of interest t, the window could include frames on either side or just one side of t. For the framework to be strictly online, the window can only look into the past. However for our dataset, as the events are so short and dense, looking a few frames into the future is very beneficial, at the cost of introducing a short lag. Therefore our window is centred on t, where a window size w_p of 1 equates to no pooling at all. Table V shows the results for different intersection-over-union (IoU) thresholds α, between proposed events and ground truth events. We find that any pooling is better than none ($w_p = 1$) with increases in mAP for all α. Comparing rows (2) and (3) in Figure 3, which shows event proposals over one of the `Point` events in the test split, highlights the 'noise-removing' effects of pooling.

Going beyond just pooling, we again use bi-directional RNNs to generate event proposals which are less affected by errors made by the CNN. Similar to the sliding window used for pooling, the RNNs take CNN `fc1` layer activations as input over a centred window of size w_{rnn}. As shown in Table V we find that the RNNs perform better than simply mean pooling, except for when α is high. This indicates for some few event proposals which are the easiest to detect, mean pooling results in more precise temporal event boundaries. Applying mean pooling after the RNNs doesn't increase event detection performance, and as seen in Table VI, has different effects on different classes. As expected the longer duration events benefit most from longer mean pooling while events from shorter classes are negatively affected by surrounding events when using larger window sizes. These findings are also reflected in Figure 3, where the pooling (row 5) has both positive and negative effects on the RNN proposals (row 4) depending on the particular event.

TABLE V. EVENT DETECTION: MAP OVER DIFFERENT IOU THRESHOLDS (α) USING TEMPORAL WINDOW POOLING OF DIFFERENT LENGTHS (w_p)

w_p	α 0.1	0.3	0.5	0.7	0.9
	Mean Pooling `sm`				
1	0.812	0.760	0.641	0.414	0.043
5	0.895	0.867	0.784	0.525	**0.048**
10	0.895	0.878	0.809	0.523	0.046
20	0.885	0.871	0.794	0.521	0.044
40	0.856	0.842	0.734	0.439	0.035
	Bi-Directional RNN `fc1` ($w_{rnn} = 25$)				
1	0.898	0.877	0.798	**0.528**	0.032
40	0.854	0.840	0.763	0.449	0.026
	Bi-Directional RNN `fc1` ($w_{rnn} = 40$)				
1	**0.905**	**0.886**	**0.814**	0.481	0.028
40	0.864	0.846	0.759	0.387	0.020

D. Event Description

We implement a sequence-to-sequence RNN framework, similar to that in [53], to generate commentary descriptions. Our sequence-to-sequence RNN consists of 4 one-way layers (2 encoding, 2 decoding) each with 256 GRU units. We manually split the sequences into `Points` as they are marked-up in the dataset. Each `Point` is represented by a sequence of CNN `fc1` activations, and each possesses a sequence of

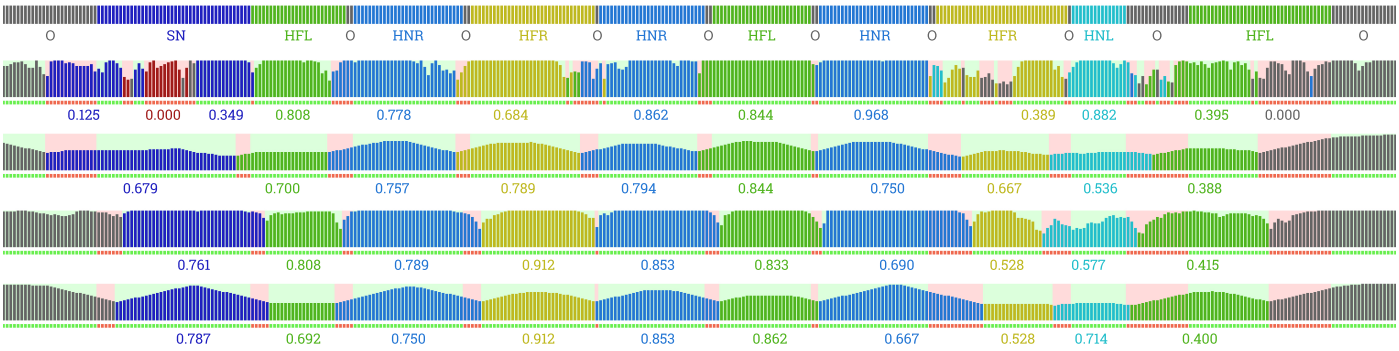

Fig. 3. Event timelines across a single `Point` in untrimmed video. Each vertical bar is a frame, bar height is confidence, numbers are IoU %. Rows from top: (1) Ground truth; (2) Two-Stream Framewise CNN (3) Two-Stream Pooled $w_p = 40$ (4) Bi-RNN `fc1` ($w_{rnn} = 40$); (5) Bi-RNN `fc1` ($w_{rnn} = 40$) $w_p = 40$.

TABLE VI. AVERAGE PRECISION (AP) FOR DIFFERENT CLASSES USING TEMPORAL WINDOW POOLING OF DIFFERENT LENGTHS (w_p) WITH IOU THRESHOLD ($\alpha = 0.5$).

class	w 1	5	10	20	40
	Mean Pooling `sm`				
O	0.121	0.286	0.357	0.398	**0.402**
HNR	0.806	0.900	0.910	**0.925**	0.846
HNL	0.681	**0.748**	0.732	0.692	0.605
HFR	0.657	0.771	**0.811**	0.808	0.789
HFL	0.567	0.679	0.747	**0.765**	0.698
SN	0.629	**0.813**	0.805	0.749	0.728
SF	0.508	0.795	**0.850**	0.827	0.738
	Bi-Directional RNN `fc1` ($w_{rnn} = 40$)				
O	0.419	0.419	0.420	0.422	**0.425**
HNR	**0.881**	0.877	0.871	0.866	0.797
HNL	0.750	**0.759**	0.749	0.744	0.674
HFR	0.820	0.842	**0.853**	0.852	0.748
HFL	**0.739**	0.705	0.684	0.677	0.601
SN	0.898	0.916	0.921	**0.938**	0.906
SF	0.796	0.814	0.849	**0.860**	0.829

words making up the commentary sentence. For this network it is necessary to build a vocabulary of all possible words (all those found in the dataset) and represent each with a different vector. Due to the low number of words in our vocabulary we simply use one-hot vectors for each word, compared to other works which use an embedding for their larger vocabularies.

To measure commentary generation performance we utilise commonly used metrics for sentence comparison seen in previous video description works BLEU [31] , METEOR [2], CIDEr [52] and ROUGE-L [23]. We use the Microsoft Evaluation Server [5] to generate these statistics. Table VII presents the results of our event description framework in terms of these metrics. We investigate if there is a link between the classification performance of our CNN models and the sentence generation performance. Although the two-stream CNN based pipeline achieves the best scores across all metrics, there appears to be no correlation between a CNNs mAP and commentary generation performance for our particular models and data. This could be because all CNNs are already accurate enough, especially after encoder processing, for the description RNNs learning capacity with our data. All models perform better than random retrieval of descriptions from the test set.

TABLE VII. EVENT DESCRIPTION: BLEU4 (B4), METEOR (M), CIDER (C) AND ROUGE-L (RL) RESULTS WITH INPUT FEATURES FROM DIFFERENT PERFORMING CNN MODELS.

CNN Model	mAP	B4	M	C	RL
Rdm. Retrieves		0.0593	0.1493	0.2713	0.3147
RGB Only	0.6748	0.1038	0.2014	0.5729	0.4078
Optical Flow Only	0.7607	0.0839	0.1905	0.4486	0.4053
Two-Stream	0.8157	**0.1284**	**0.2223**	**0.6777**	**0.4518**

We believe the standard metrics are not a reliable indication of how 'good' a description is as they all rely on word matching. Particularly for our task where the vocabulary is small, a single word difference, for example *near* player v *far* player, might completely change the meaning of the description. With this in mind we provide some example results in Table VIII which we believe provides a better indication of how well our model is able to generate sentences. In the table we highlight 'concept' errors in red, and sentence generation errors in blue. Empirically, sentences are mostly correct however they can suffer from repeated or misplaced words, as well as the occasional concept error.

E. Implementation Details

We use the Keras library [6] to train and test our models. All CNN models in our experiments are trained in similar circumstances, using stochastic gradient descent over 5 epochs, with a learning rate of 0.01, a decay of 1×10^{-6} and momentum of 0.5. The loss function employed is categorical cross-entropy. We sample the validation accuracy every epoch, and take the highest scoring model for testing. Similarly for our RNN models we train all with equal parameters, using RMSProp with learning rate of 0.001 and $\rho = 0.9$, and again categorical cross-entropy loss. The RNNs are made up of layers of Gated Recurrent Units (GRU), which have been shown to achieve similar results to Long-Short Term Memory Units (LSTM) while using less parameters. For the sentence generation encoder-decoder RNN we use the same learning rate of 0.001 and $\rho = 0.9$ but employ mean squared error as the loss function.

TABLE VIII. EVENT DESCRIPTION: EXAMPLES WITH TWO-STREAM CNN. GREEN: CORRECT, RED: CONCEPT INCORRECT, BLUE: SENTENCE GENERATION ERROR.

Point ID	Ground Truth Sentence	Predicted Sentence
P00000628	good serve aimed in the corner np only reaches to it	*sharp angled slice serve is an ace over the net*
P00000627	good serve in the middle np crafts a rs return a brief rally fp rs cross-court is a winner	*good serve in the middle np returns a rs return short rally np produces a rs cross-court winner the line*
P00000743	quick serve fp returns a rs return rally np cross-court rs fails to clear the net in the middle	*heavy serve serve fp crafts a rs return couple of shots exchanged np rs a rs a rs the the net*
P00000597	fp hits a flat bodyline serve np struggles to put it back	*fp arrows a serve serve np return it over net*
P00000746	lightening serve down t is an ace	*np serves out ace wide wide wide*
P00000606	good serve in the middle fp returns a rs np hits a rs winner	*np aims a bodyline serve at t fp an no*
P00000600	fp serves a good one np delivers a ls return good rally fp hits a ls cross-court drop-shot winner	*fp serves a good one np delivers a ls return brief rally np hits to rs net the the net*

V. CONCLUSION

This work introduces a unified dataset for the problems of event recognition, detection and description in video. The dataset focuses on dense fine-grained events enabling the generation of event dependent and contextually relevant descriptions. We form some initial baseline results using recent deep learning approaches finding that, although the framework is relatively simple and generalised, it is still able to achieve impressive results across all tasks.

We see this work as a first step towards the extraction of more detailed and useful information from video. By placing greater focus on movement, actions and events we are able to produce detailed descriptions of long and complex video sequences. In future, we believe this will support deeper video understanding, such as the ability to learn the rules of a game by watching it being played for a period of time.

We recognise a few shortcomings of our dataset. Firstly, in relation to recent datasets utilised for deep methodologies, our is relatively small and therefore vulnerable to over-fitting with these deep models. Like most cases acquiring such detailed and domain specific information is not a straighfoward task. Labelling and temporal alignment of events manually is not feasible at large scale. Secondly, although the actions in our dataset are fine-grained, we believe our CNN framework finds player court positioning, movement and pose the most discriminative cues for deciding on our action classes. This could be alleviated to some degree by making the classes even more specific, such as topspin, slice, backspin, *etc*.

REFERENCES

[1] S. Abu-El-Haija, N. Kothari, J. Lee, P. Natsev, G. Toderici, B. Varadarajan, and S. Vijayanarasimhan. YouTube-8M: A Large-Scale video classification benchmark. 27 Sept. 2016.

[2] S. Banerjee and A. Lavie. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, volume 29, pages 65–72. aclweb.org, 2005.

[3] F. Caba Heilbron, V. Escorcia, B. Ghanem, and 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*, pages 961–970, 2015.

[4] D. L. Chen and W. B. Dolan. Collecting highly parallel data for paraphrase evaluation. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 190–200, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics.

[5] X. Chen, H. Fang, T.-Y. Lin, R. Vedantam, S. Gupta, P. Dollar, and C. Lawrence Zitnick. Microsoft COCO captions: Data collection and evaluation server. 1 Apr. 2015.

[6] F. Chollet et al. Keras. https://github.com/fchollet/keras, 2015.

[7] W. J. Christmas, A. Kostin, F. Yan, I. Kolonias, and J. Kittler. A system for the automatic annotation of tennis matches. In *Fourth international workshop on content-based multimedia indexing*. researchgate.net, 2005.

[8] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2625–2634. cv-foundation.org, 2015.

[9] V. Escorcia, F. C. Heilbron, J. C. Niebles, and B. Ghanem. DAPs: Deep action proposals for action understanding. In *Computer Vision – ECCV 2016*, pages 768–784. Springer, Cham, 8 Oct. 2016.

[10] C. Feichtenhofer, A. Pinz, and A. Zisserman. Convolutional two-stream network fusion for video action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1933–1941. cv-foundation.org, 2016.

[11] G. Gkioxari and J. Malik. Finding action tubes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 759–768. cv-foundation.org, 2015.

[12] A. Gorban, H. Idrees, Y. G. Jiang, A. R. Zamir, I. Laptev, M. Shah, and R. Sukthankar. THUMOS challenge: Action recognition with a large number of classes. In *CVPR workshop*, 2015.

[13] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri. Actions as Space-Time shapes. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(12):2247–2253, Dec. 2007.

[14] C. Gu, C. Sun, S. Vijayanarasimhan, C. Pantofaru, D. A. Ross, G. Toderici, Y. Li, S. Ricco, R. Sukthankar, C. Schmid, and J. Malik. AVA: A video dataset of spatio-temporally localized atomic visual actions. 23 May 2017.

[15] M. Jain, H. Jegou, and P. Bouthemy. Better exploiting motion for better action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2555–2562. cv-foundation.org, 2013.

[16] H. Jhuang, H. Garrote, E. Poggio, T. Serre, and T. Hmdb. A large video database for human motion recognition. In *Proc. of IEEE International Conference on Computer Vision*, volume 4, page 6, 2011.

[17] S. Ji, M. Yang, and K. Yu. 3D convolutional neural networks for human action recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(1):221–231, Jan. 2013.

[18] Y. G. Jiang, J. Liu, A. R. Zamir, G. Toderici, I. Laptev, M. Shah, and R. Sukthankar. THUMOS challenge: Action recognition with a large number of classes, 2014.

[19] S. Karaman, L. Seidenari, and A. Del Bimbo. Fast saliency based pooling of fisher encoded dense trajectories. In *ECCV THUMOS Workshop*, volume 1, page 5. crcv.ucf.edu, 2014.

[20] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1725–1732, 2014.

[21] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, M. Suleyman, and A. Zisserman. The kinetics human action video dataset. 19 May 2017.

[22] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. HMDB: A large video database for human motion recognition. In *2011*

International Conference on Computer Vision, pages 2556–2563. ieeexplore.ieee.org, Nov. 2011.

[23] C.-Y. Lin. ROUGE: A package for automatic evaluation of summaries. In S. S. Marie-Francine Moens, editor, *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*, pages 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics.

[24] S. Ma, L. Sigal, and S. Sclaroff. Learning activity progression in lstms for activity detection and early detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1942–1950. cv-foundation.org, 2016.

[25] M. Marszalek, I. Laptev, and C. Schmid. Actions in context. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2929–2936. ieeexplore.ieee.org, June 2009.

[26] H. Miyamori and S. I. Iisaku. Video annotation for content-based retrieval using human behavior analysis and domain knowledge. In *Proceedings Fourth IEEE International Conference on Automatic Face and Gesture Recognition (Cat. No. PR00580)*, pages 320–325. ieeexplore.ieee.org, 2000.

[27] A. Montes, A. Salvador, S. Pascual, and X. Giro-i Nieto. Temporal activity detection in untrimmed videos with recurrent neural networks. In *1st NIPS Workshop on Large Scale Computer Vision Systems*. arxiv.org, 29 Aug. 2016.

[28] J. Y.-H. Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici. Beyond short snippets: Deep networks for video classification. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4694–4702, June 2015.

[29] D. Oneata, J. Verbeek, and C. Schmid. The LEAR submission at thumos 2014. *ECCV THUMOS Workshop*, 2014.

[30] Y. Pan, T. Mei, T. Yao, H. Li, and Y. Rui. Jointly modeling embedding and translation to bridge video and language. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4594–4602, June 2016.

[31] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. BLEU: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 311–318, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics.

[32] Á. Peris, M. Bolaños, P. Radeva, and F. Casacuberta. Video description using bidirectional recurrent neural networks. In *Artificial Neural Networks and Machine Learning – ICANN 2016*, pages 3–11. Springer, Cham, 6 Sept. 2016.

[33] G. Pingali, A. Opalach, and Y. Jean. Ball tracking and virtual replays for innovative tennis broadcasts. In *Proceedings 15th International Conference on Pattern Recognition. ICPR-2000*, volume 4, pages 152–156 vol.4. ieeexplore.ieee.org, 2000.

[34] V. Ramanathan, J. Huang, S. Abu-El-Haija, A. Gorban, K. Murphy, and L. Fei-Fei. Detecting events and key actors in multi-person videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3043–3053, 2016.

[35] M. Regneri, M. Rohrbach, D. Wetzel, S. Thater, B. Schiele, and M. Pinkal. Grounding action descriptions in videos. *Transactions of the Association for Computational Linguistics*, 1(0):25–36, 31 Mar. 2013.

[36] A. Rohrbach, M. Rohrbach, W. Qiu, A. Friedrich, M. Pinkal, and B. Schiele. Coherent multi-sentence video description with variable level of detail. In *Pattern Recognition*, pages 184–195. Springer, Cham, 2 Sept. 2014.

[37] A. Rohrbach, M. Rohrbach, N. Tandon, and B. Schiele. A dataset for movie description. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3202–3212, June 2015.

[38] M. Rohrbach, S. Amin, M. Andriluka, and B. Schiele. A database for fine grained activity detection of cooking activities. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1194–1201, June 2012.

[39] M. Rohrbach, A. Rohrbach, M. Regneri, S. Amin, M. Andriluka, M. Pinkal, and B. Schiele. Recognizing Fine-Grained and composite activities using Hand-Centric features and script data. *Int. J. Comput. Vis.*, 119(3):346–373, 1 Sept. 2016.

[40] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet large scale visual recognition challenge. *Int. J. Comput. Vis.*, 115(3):211–252, 1 Dec. 2015.

[41] C. Schuldt, I. Laptev, and B. Caputo. Recognizing human actions: a local SVM approach. In *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, volume 3, pages 32–36 Vol.3, Aug. 2004.

[42] R. Shetty and J. Laaksonen. Video captioning with recurrent networks based on frame- and video-level features and visual content classification. 9 Dec. 2015.

[43] Z. Shou, D. Wang, and 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*, pages 1049–1058. cv-foundation.org, 2016.

[44] G. A. Sigurdsson, G. Varol, X. Wang, A. Farhadi, I. Laptev, and A. Gupta. Hollywood in homes: Crowdsourcing data collection for activity understanding. In *Computer Vision – ECCV 2016*, pages 510–526. Springer, Cham, 8 Oct. 2016.

[45] K. Simonyan and A. Zisserman. Two-Stream convolutional networks for action recognition in videos. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 568–576. Curran Associates, Inc., 2014.

[46] K. Simonyan and A. Zisserman. Very deep convolutional networks for Large-Scale image recognition. 4 Sept. 2014.

[47] B. Singh, T. K. Marks, M. Jones, O. Tuzel, and M. Shao. A multi-stream bi-directional recurrent neural network for fine-grained action detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1961–1970. cv-foundation.org, 2016.

[48] K. Soomro, A. R. Zamir, and M. Shah. UCF101: A dataset of 101 human actions classes from videos in the wild. 3 Dec. 2012.

[49] M. Sukhwani and C. V. Jawahar. TennisVid2Text: Fine-grained descriptions for domain specific videos. 26 Nov. 2015.

[50] A. Torabi, C. Pal, H. Larochelle, and A. Courville. Using descriptive video services to create a large data source for video annotation research. *arXiv preprint arXiv:1503. 01070*, 2015.

[51] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4489–4497, 2015.

[52] R. Vedantam, C. Lawrence Zitnick, and D. Parikh. Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4566–4575. cv-foundation.org, 2015.

[53] S. Venugopalan, M. Rohrbach, J. Donahue, R. Mooney, T. Darrell, and K. Saenko. Sequence to sequence – video to text. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 4534–4542, Dec. 2015.

[54] S. Venugopalan, H. Xu, J. Donahue, M. Rohrbach, R. Mooney, and K. Saenko. Translating videos to natural language using deep recurrent neural networks. 15 Dec. 2014.

[55] H. Wang, A. Kläser, C. Schmid, and C. L. Liu. Action recognition by dense trajectories. In *CVPR 2011*, pages 3169–3176, June 2011.

[56] H. Wang and C. Schmid. Action recognition with improved trajectories. In *Proceedings of the IEEE international conference on computer vision*, pages 3551–3558. cv-foundation.org, 2013.

[57] H. Wang, M. M. Ullah, A. Klaser, I. Laptev, and C. Schmid. Evaluation of local spatio-temporal features for action recognition. In *BMVC 2009-British Machine Vision Conference*, pages 124–121. hal.inria.fr, 2009.

[58] L. Wang, Y. Qiao, and X. Tang. Action recognition and detection by combining motion and appearance features. *THUMOS14 Action Recognition Challenge*, 1(2):2, 2014.

[59] X. Wang, A. Farhadi, and A. Gupta. Actions transformations. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[60] H. Xu, S. Venugopalan, V. Ramanishka, M. Rohrbach, and K. Saenko. A multi-scale multiple instance video description network. 21 May 2015.

[61] J. Xu, T. Mei, T. Yao, and Y. Rui. MSR-VTT: A large video description dataset for bridging video and language. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5288–5296, June 2016.

[62] L. Yao, A. Torabi, K. Cho, N. Ballas, C. Pal, H. Larochelle, and A. Courville. Describing videos by exploiting temporal structure. In *Proceedings of the IEEE international conference on computer vision*, pages 4507–4515, 2015.

[63] S. Yeung, O. Russakovsky, N. Jin, M. Andriluka, G. Mori, and L. Fei-Fei. Every moment counts: Dense detailed labeling of actions in complex videos. 21 July 2015.

[64] S. Yeung, O. Russakovsky, G. Mori, and 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*, pages 2678–2687. cv-foundation.org, 2016.

[65] M. Zanfir, E. Marinoiu, and C. Sminchisescu. Spatio-Temporal attention models for grounded video captioning. 17 Oct. 2016.