NetClips: A Framework for Video Analytics in Sports Broadcast

Masoumeh Izadi, Aiden Chia, Bernard Cheng, Shangjing Wu
Television Content Analytics, R&D department
Singapore
masoumeh.izadi@tvconal.com, aidenchia95@gmail.com, bernardcheng95@hotmail.com, leoveesg@gmail.com

***Abstract*—This work presents an early stage big data application, NetClips, for automation in broadcast production in the area of content management, delivery, and consumption. NetClips acts as a cognitive engine to automatically annotate video clips with contextual sports content during match production and on archival contents. Ensemble of deep learning architectures and advanced machine vision techniques are used in NetClips to accurately recognize players, objects, actions, and events in every clip created during a sport broadcast. This application also has a great impact on production workflow, as it will lift up the burden of concurrent tagging from operators list of duties.**

Keywords-unstructured data, automatic annotation, deep learning architecture, player detection, sports game events, cricket

I. INTRODUCTION

The ability to quickly repeat and slow down sport's finest and most contentious moments is a paramount feature in every sports broadcast production. Although there are multiple technologies that offer the hardware and the systems in instant replay for clipping, cutting, tagging, editing, ingesting, saving, and streaming video contents from multiple camera systems, there is still a need for many experienced operators to manually do those tasks in a very fast pace. Thousands of clips of different lengths are created during a live match. Untagged or minimally tagged piles of clips can quickly become a logistical nightmare when it comes to retrieving the right content. In a live event, broadcast operators have to be extremely dedicated and focused on their individual tasks to instantly find and play what the producers ask. While an operator can tediously remember the content of clips he created at the production time, the situation gets much harder if there's a need to retrieve a clip from archived contents.

Video analytics in sports technology has been progressing rapidly in the area of tracking [1,2]. Recent advances in such technologies have gained a great amount of success in detection of balls and athletes in the outdoor environments, which used to be very challenging problems. However, such technologies have been developed with the objective of identifying the coordinates of the ball and players, which is substantially different than understanding what game event goes on in a video clips. In addition, the video input that are used for tracking purposes in sports mainly require multiple specialised cameras installed around the game venue, that are different from broadcast cameras and introduce a higher cost to sports production. Therefore, these technologies are not easily expandable to any live coverage, let alone the archived ones.

In this work, we try to address the task of contextual retrieval of sports contents through thousands of unstructured video clips. We introduce NetClips for cognitively processing and tagging videos during the fast-paced rhythm of a live sports production. Currently available video clips tagging solutions such as Google's Cloud Video Intelligence are not easily applicable to this problem because they are too general to understand the game semantics and identify all players. NetClips is trained with a large sets of broadcast video clips form the corresponding sport matches and libraries of players, player actions, game events, and relevant objects. This application will be equipped with thousands of built-in search terms for consuming archived content and can be integrated in in an easy to use format in other platforms such as OTTs.

The most fundamental annotations by NetClips include team names, player names and their jersey number, player actions, and type of sports-specific game events. To the best of our knowledge, the only similar video analytics solution to NetClips is STATS-Edge developed for soccer [3]. However, Edge is only suitable for archived videos and mainly is targeted at teams and leagues use.

In this paper, we focus on the professional sport of cricket, which is the second most watched sport after soccer with over a billion fans [4]. Cricket matches are long in duration compared to other sports. Cricket is played in three formats according to the maximum number of plays or deliveries allowed for each team. In the shortest format, T20, a live match can take more than twice the length of a soccer match. Consequently, the volume of the unstructured data such as video clips and commentary text produced is much larger than other sports.

In televised cricket international games and in some popular leagues, there are about forty broadcast camera systems involved in the match coverage. Therefore, to manage the original video feeds from each camera system a big team of operators and sophisticated equipment are required. Currently, the content tagging is performed manually in a fragmented manner only on a small percentage of the large set of data created for each match. NetClips will be integrated in video servers and media management

systems to reduce the number of human operators required for a live broadcast and for making the content available for instant and future use.

Deep Learning and Convolutional Neural Network (CNN) has recently gained a great success in many big data applications, especially in large-scale image and video recognition [5]. These models usually need a lot of data samples in the learning phase to converge. It is a natural way to use these models for detection tasks in our application. While we have huge number of video clips in our disposal taken from years of games played in international and domestic levels, annotating these samples is labor intensive and time consuming. For this reason, in the proof of concept phase of NetClips we have focused on a selected number detection tasks to gain confidence in our methodology. Nonetheless, the results will be generalizable.

II. METHODS

We evaluate several deep neural network architectures to combine image information across a video clip. We distinguish the task of object detection, player detection, and identification of players from the task of game-event detection due to the temporal information related to the game events. We need to model the motion in the video as an ordered sequence of frames in order to define of an event that takes sometimes up to four seconds of the footage.

A. Data

In practice, video clips are created based on raw feeds from individual camera systems or from a broadcast feed. However, in order to generalize the use of NetClips for existing archived broadcast contents, which normally is stored by television networks, we only used data extracted from broadcast feed. For simplicity, we considered limited number of teams, players, leagues, and game events in the first phase of our developments. In particular, we used the full cricket broadcast footage of Cricket World Cup 2015 matches, limited to the players from three countries: Australia, Sri Lanka, and West Indies, and four categories of the most fundamental cricket game events.

Selected footages were manually clipped into 5-second long video clips and manually annotated into four main classes: Players with names from the team squads, Wicket, Boundary score 4, and Boundary score 6. Naturally, the number of clips per player is different as most famous players play more games, stay longer in the field, and are captured in more footage. In addition to the video clips, we used about 400 images per individual players that are manually annotated with the player names. These images in combination with the videos are used for player identification.

B. Preprocessing

In extracting the video clips we chose clips where the majority of frames show the player as at least one quarter of the foreground. Jittered sampling was performed on frames of each clip to remove frames that are too similar with adjacent temporal frames. Jittered Sampling is implemented as follows: To select x frames from the total number of frames of a clip, x_{total}, the set of video frames are divided into x contiguous sections, with the *i-th* section consists of frames from frame $\left\lfloor \frac{x_{total}}{x} \right\rfloor * (i-1)$ to frame $\left\lfloor \frac{x_{total}}{x} \right\rfloor * (i)$. From each section, one frame is then selected at random.

The bottom regions of all video frames are uniformly cropped out as it contains distracting information such as the game scoreboard. Each frame is then resized to 224 pixels by 224 pixels in order to input the image through the Convolutional Neural Network (CNN) model.

C. Models

In the development of our player identification model, we employed three of state-of-the-art CNNs: ResNet-50 [6], Inception-ResNet-v2 [7], and Xception [8]. We implemented the fore-mentioned models using Keras library as pre-trained models excluded the final default output layer in these CNN models in favor of a customized output layer specifically for our multiclass classification algorithm. Changing various hyperparameters of each of these three models, a total of 28 models were used for player identification and the model with highest test accuracy was selected as our player recognition model. The majority of layers for these models were frozen to reduce the potential overfitting problems arising from tuning the sheer number of parameters associated with these extremely deep networks. Early stopping policy was implemented as another way of preventing overfitting.

The game-event recognition task falls under the domain of a f class of learning problems: $\langle x_1, x_2, \dots, x_t \rangle \rightarrow y$. For a sequence of t frames as input, we want to predict a single class label y. For this purpose we chose a recurrent neural network that uses Long Short-Term Memory (LSTM) [9]. LSTM networks operate on frame-level CNN activations, and can learn how to integrate information on relationship between the frames over time. Similar to Guadarrama et al, [10] we used a Long-term Recurrent Convolutional Network (LRCN) model combining a CNN as a deep hierarchical visual feature extractor with a LSTM that can exploit the temporal dependencies between the frames. More specifically, we used the CNN to produce a fixed-length vector representation of the individual frames, and the vectors are subsequently fed into an LSTM model to predict the class label, determining if the frames belong to any of the three game events of Boundary 4, Boundary 6, and Wicket, or they belong to other action or events labeled as Others. For the CNN model we employed the ResNet-50 with pre-trained weights. We decided to use ResNet-50 as it outperformed other models in our player identification task.

Similarly, we removed the original 1,000-class SoftMax layer and replaced it with a fully connected dense layer. We fine-tuned the weights of our CNN over twenty epochs. Since we used a large number of parameters in the network, an effective regularization mechanism is essential to combat overfitting. We used a powerful regularization approach of dropout to reduce the potential overfitting. Once the CNN model was trained, we removed the last dense layer of the CNN so that the final output of the CNN is now a 1,028-dimensional 1D vector. This 1,028-dimensional 1D vector is then reshaped and passed as an input to the LSTM in batch sizes of 50 frames, which corresponds to exactly 1 video clip. The LSTM was trained over 100 epochs. We simply took the output vectors corresponding to the 50th frame of each clip and passed them through an argmax function to find out which class each clip belonged to.

III. EXPERIMENTS

There are two sets of evaluation experiments for this paper: player identification, and event/activity recognition. We trained and tested the models for player identification over players that have the most number of clips in our data set. This includes only 11 players of the three squads. We have considered player names as the labels for those selected players and the rest of the players in a category labeled as *Others*.

TABLE 1-Player Recognition Evaluation

Model	Resnet50	Inception ResNet v2	Xception
Epoch	20	20	20
No. Of trainable layers	150	50	131
Learning rate	1.00E-05	1.00E-05	1.00E-05
Accuracy	**84.46%**	43.5%	65.2%

Table1 listed the best average results achieved by different models. We observed that the ResNet-50 outperformed other models in terms of accuracy. Thus ResNet-50 will be chosen as the architecture of choice in NetClips for player identification.

Post-classification analysis of the misclassified images showed that those images with a bright and colorful background were more likely to be categorized wrongly, suggesting that reducing the noise of the background might achieve a higher accuracy. Therefore, we developed a foreground object extraction algorithm, which bounded the player with a box and recolored the background area outside of the bounding box with the average color of the background area. However, the foreground object extraction algorithm still hasn't shown a satisfactory improvement on the performance of player recognition algorithm. In future, we will invest more time on the development of the foreground object extraction algorithm.

In the evaluation of our activity recognition experiment we distinguished four categories: *Boundary 4*, *Boundary 6*, *Wicket*, and *Others*. We sampled frames through Jittered sampling from the selected clips to extract 50 frames per clip.

TABLE 2-Activity Recognition Evaluation

Model	Accuracy
CNN only	0.473
LRCN (64)	**0.857**
LRCN (128)	0.857
LRCN (512)	0.809

In Table 2, we evaluate the performance of the models in recognizing actions in video based on accuracy. We observe that using a CNN alone to classify the frames is not enough as it only achieves an accuracy of 0.473. This is expected, since many of the frames (especially Boundary 4 and Boundary 6) are very similar and it is difficult for the CNN to distinguish between these frames. However, the LRCN model far outperforms the CNN-only model, achieving an accuracy of 0.857. Our tests also show that increasing the number of hidden states in the LSTM from 64 to 128 has negligible effect on accuracy, while increasing from 128 to 512 has a detrimental impact on performance efficiency. The runtime of this model is around 1.2 seconds on every 50 frames.

IV. CONCLUSION AND FUTURE WORK

NetClips automates proper tagging and classifies all the clips produced by any broadcast operator to be retrieved easily during or after production using built-in search functionality. NetClips has practical use across a wide variety of entertainment productions as well as sports, for which its use probably has more impact.

With increasing demand to consume sports content on OTT, these platforms will also need to generate consumable sports packages from live streams almost immediately after the live broadcast. AI-empowered tools such as NetClips that can process unstructured large-scale video data to automatically generate insights and provide tagging are of tremendous help because in a traditional setup, this process could involve dedicated resources to search through hours of broadcast videos.

In this research, we focused on exploring deep-learning based architectures for the detection of three main events in broadcast videos capturing the professional sport of cricket. We employed pre-trained convolutional neural networks for extracting temporal features and we investigated the use of ensemble architecture for fusion of the different networks. To validate the capabilities of the proposed architectures, we used a large dataset of broadcast footage from international cricket matches. Experimental results suggest that very low classification error can be achieved for both player identification and game event detection.

There are significant algorithmic and performance challenges we need to address in the future phases of development in NetClips. Exploiting scale in model size is central to the success of deep learning and since our datasets are sufficiently large, we can potentially increase the number of parameters in our networks that can give much better prediction accuracy. In terms of computational complexity, when pre-trained CNN are utilized, the most demanding operation is the extraction of the optical flow information. Future work will examine the impact of support vector machines in combination with our CNNs for achieving even a higher classification performance.

However, the recognition timeliness is of special importance in using NetClips in live sport coverage and should be considered as a tradeoff factor. There are a number of approaches recently introduced in the literature to overcome such computation-accuracy tradeoffs. In particular, Mixture-of-Experts layer (MoE) [11] has been proposed for significantly scaling up network architecture but using scarcity of neurons' activation to keep the computational resources in control. MoE architecture seems to perform very well in machine translation applications. We plan to examine this approach in NetClips.

REFERENCES

[1] Paul McIlroy, Hawk-Eye: Augmented reality in sports broadcasting and officiating. Proceedings of the 7th IEEE/ACM International Symposium on Mixed and Augmented Reality, 2008.

[2] Chyronhego Corporation, TRACAB Optical Tracking https://chyronhego.com/wp-content/uploads/2018/02/TRACAB-PI-sheet.pdf, 2018.

[3] SPORTTECHIE: Kevin Nardone, "STATS Launches New Artificial Intelligence Video and Analysis Tool Edge", https://www.sporttechie.com/stats-launches-artificial-intelligence-video-analysis-tool-edge/ , August 2018.

[4] International Cricket Council, ICC Global Survey 2018: Securing future growth, 2018.

[5] Qingchen Zhang, T.Yang, Zhikui Chenc, PengLi, "A survey on deep learning for big data", Journal of Information Fusion,Volume 42, Pages 146-157, July 2018.

[6] He, Kaiming, et al. "Deep residual learning for image recognition." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.

[7] Szegedy, Christian, et al. "Inception-v4, inception-resnet and the impact of residual connections on learning." AAAI. Vol. 4. 2017.

[8] Chollet, François. "Xception: Deep learning with depthwise separable convolutions." arXiv preprint (2017): 1610-02357.

[9] S. Hochreiter and J. Schmidhuber. Long short-term memory. Neural Computing, 9(8):1735–1780, Nov. 1997.

[10] Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, Trevor Darrell, Jeff Donahue, Lisa Anne Hendricks. Long-term recurrent convolutional networks for visual recognition and description.

[11] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, Jeff Dean, "OUTRAGEOUSLY LARGE NEURAL NETWORKS: THE SPARSELY-GATED MIXTURE-OF-EXPERTS LAYER", ICLR 2017.