Detection of Tennis Events from Acoustic Data

Aaron Baughman
baaron@us.ibm.com
IBM
Research Triangle Park, NC, USA

Eduardo Morales
eduardo.morales@ibm.com
IBM Research
Yorktown Heights, NY, USA

Gary Reiss
gwreiss@us.ibm.com
IBM
Atlanta, GA, USA

Nancy Greco
grecon@us.ibm.com
IBM Research
Yorktown Heights, NY, USA

Stephen Hammer
hammers@us.ibm.com
IBM
Atlanta, GA, USA

Shiqiang Wang
wangshiq@us.ibm.com
IBM Research
Yorktown Heights, NY, USA

ABSTRACT

Professional tennis is a fast-paced sport with serves and hits that can reach speeds of over 100 mph and matches lasting long in duration. For example, in 13 years of Grand Slam data, there were 454 matches with an average of 3 sets that lasted 40 minutes. The fast pace and long duration of tennis matches make tracking the time boundaries of each tennis point in a match challenging. The visual aspect of a tennis match is highly diverse because of its variety in angles, occlusions, resolutions, contrast and colors, but the sound component is relatively stable and consistent. In this paper, we present a system that detects events such as ball hits and point boundaries in a tennis match from sound data recorded in the match. We first describe the sound processing pipeline that includes preprocessing, feature extraction, basic (atomic) event detection, and point boundary detection. Then, we describe the overall cloud-based system architecture. Afterwards, we describe the user interface that includes a tool for data labeling to efficiently generate the training dataset, and a workbench for sound and model management. The performance of our system is evaluated in experiments with real-world tennis sound data. Our proposed pipeline can detect atomic tennis events with an F1-score of 92.39% and point boundaries with average precision and recall values of around 80%. This system can be very useful for tennis coaches and players to find and extract game highlights with specific characteristics, so that they can analyze these highlights and establish their play strategy.

CCS CONCEPTS

• **Computing methodologies** → **Machine learning approaches**; • **Applied computing** → **Sound and music computing**; • **Computer systems organization** → *Cloud computing*.

MMSports '19, October 25, 2019, Nice, France

ACM ISBN 978-1-4503-6911-4/19/10…$15.00
https://doi.org/10.1145/3347318.3355520

KEYWORDS

Acoustic classification, event detection, machine learning, multimedia system, sports

1 INTRODUCTION

Through all of the 2018 tennis Grand Slam tournaments that include the Australian Open Tennis Championships, Roland Garros, Wimbledon Championships, and the U.S. Open Tennis Championships, thousands of hours of tennis play was recorded. Practice facilities such as the United States Tennis Association Player Development (USTA-PD) collect additional thousands of hours of tennis practice. The majority of these videos *do not* contain gameplay annotations and only include high-level summary information such as play outcome, venue, players, and date. Even for the actual Grand Slam matches, only 60% of them have ball and player tracking information from a system such as Hawkeye [3] that can be used to create play annotations such as stroke, duration, player running distance, etc. Among the tennis practice facilities, very few of them have ball tracking data.

Before the tournaments begin, tennis competitors spend hundreds of hours practicing their forms and establishing their play strategy. The videos captured during the tournament and practice play are frequently used by coaches and players to refine their game. These videos provide multimedia tennis feedback that can be manually spliced, scrolled, and panned, but the visual aspect of the media is very diverse with numerous angles, occlusions, resolutions, contrast, and colors. The latter fact makes it very difficult for humans to find specific game patterns that deserve further analysis by coaches or players, particularly when considering that there are thousands of hours of video recordings.

To solve the above problem, Artificial Intelligence (AI) systems can be designed to detect useful game patterns from a long duration of tennis play recording. Machine learning models can be trained on video segments with pre-defined labels. Using these models, a tennis recording without gameplay annotations can be automatically annotated by the AI system. However, several challenges exist towards this goal: 1) The videos recorded in tennis games are usually very diverse (as explained above); therefore, training a good machine learning model to detect different video events would require a large number of labeled video segments, which is difficult to collect. 2) Processing long durations of video requires a large amount of processing power, which makes the system not scalable to a large number of users or long duration of recordings.

3) A particular game pattern usually consists of multiple smaller events, which makes the machine learning modeling more difficult than detecting each single event alone. 4) The human labeling of data for a large amount of tennis events is a time-consuming task.

In this paper, we address the above challenges by first observing that although the visual signatures are diverse, the *acoustic signatures* of actions during play are relatively stable and consistent throughout the tennis recordings. Based on this observation, we propose a system that efficiently detects game patterns from the audio that is recorded together with the video in tennis games. Since our system processes the audio, the amount of required training data and processing time is much smaller than what would be required for video processing. The start and end times of particular game patterns detected from the audio signal can then be annotated in the original video recording, which can be easily searched for and played back to users. For example, coaches and players can find video segments about certain strokes to analyze body position throughout a tennis ball hit. We focus on detecting atomic tennis events and tennis point boundaries in this paper, but the pipeline can be extended for detecting other game patterns as well.

Our main contributions in this paper are summarized as follows:

(1) We propose a pipeline for tennis event detection that includes the preprocessing of sounds to detect the existence of potential events, feature extraction from sounds, the detection of basic atomic events (such as ball hit) in tennis, and the detection of the start and end times of a tennis point which includes a sequence of multiple basic events.
(2) We present the overall system architecture that uses container technology and enables the efficient deployment of the tennis event detection service in a cloud.
(3) We present a data labeling tool that allows the user to efficiently label sound segments that potentially contain tennis events, where these labeled sound segments are used for model training, as well as a web interface for the efficient management of recorded sounds and trained models.
(4) We present experimentation results that show the accuracy of our models.

We note that different from most existing works which assume that the training dataset is already given to the system beforehand, our work in this paper considers the *entire life-cycle of the system* where initially the system has no labeled data. We provide tools for human users to efficiently assign labels to data (sounds) containing tennis events, and manage labeled and unlabeled data as well as trained models. This design consideration is important for the practical applicability of the system.

The rest of this paper is organized as follows. We review the related work in Section 2. In Section 3, we present our pipeline for tennis event detection with machine learning models. Section 4 describes the system architecture. The user interfaces for data labeling and model/sound management are described in Section 5. The experimentation setup and results are presented in Section 6 and Section 7 draws conclusions.

2 RELATED WORK

2.1 Automatic Video Annotations

With the growth of video content, there is an increasing demand for automatic interpretation of videos. Within sports, video meta tagging and excitement ranking by Merler et al. ranked videos with multimodal signatures such as actor gesture, commentator, crowd noise, and speaker content [18, 19]. The work includes streaming professional tennis and golf videos with several deep learning algorithms to find highlight start and end times. However, these methods based on video processing generally have very high computational complexity as mentioned in Section 1, which may not scale well to scenarios with a large amount of video data.

Sports highlight detection using only audio data has been studied in [9, 13], which is based on sounds such as whistles, speech, and crowd noise. These works only use shallow models such as the Gaussian Mixture Model (GMM), and only focus on specific machine learning approaches instead of the overall system design.

2.2 Acoustic Recognition

The classification of acoustic signals into different classes has received much attention in the literature, for various application scenarios. The current state of the art within acoustic recognition attempts to use different architectures of deep neural networks. For example, Oines et al. use Long Short-Term Memory (LSTM), Feedforward Sequential Memory Network (FSMN), and a combination of LSTM and FSMN with a Connectionist Temporal Classification (CTC) criteria for acoustic modeling [21]. They showed that their hybrid model performs better on acoustic classification over 3 languages. A Kaggle competition on acoustic event classification had 69 teams [1], where the winning team implemented a weighted ensemble that uses several different Convolutional Neural Networks (CNNs) [11]. In the DCASE 2017 competition, the detection of rare sound events and acoustic event classification were largely solved by CNNs as well [20].

However, when a smaller and focused feature set is engineered, conventional algorithms such as GMM and Support Vector Machine (SVM) may perform well in specific application domains too [7]. Some works use Mel-Frequency Cepstral Coefficient (MFCC) as input into environmental sound recognition [6, 12]. Chu et al. proposed a system that uses GMM to classify sounds from 14 predetermined classes [6]. Lu et al. proposed a method that is based on both K nearest neighbor and GMM algorithms for audio classification and segmentation [16].

Li et al. discuss comparisons of several variants of deep neural networks and shallow models [15]. For their domain, they found that deep neural networks provide superior performance. However, work by Dai found that conventional models such as GMM and SVM perform better on some classification tasks [7].

None of the above approaches mentioned in this subsection are developed for classifying tennis events. We propose an approach for detecting and classifying tennis events in this paper.

2.3 Time Series Classification

Evidence trending over time provides a temporal basis for time series classification and event detection. Various deep learning

methods can be used for time series classification [10]. For example, LSTM was used to model long-range structural dependencies in video for automatic video summarization [22]. Other works such as [8] use streams event processing based on domain encoded rules and equations, which have already been discovered to predict highlight markers. Dynamic Time Warping (DTW) and nearest neighbor models have been used as a method for time series classification as well [4]. Bagnall et al. maintain a repository of algorithms and datasets for time series classification, which shows the depth of this area [5]. As an extension to the body of work on time series classification, in this paper, we propose an approach for detecting start and end boundaries of tennis points from the result provided by the atomic tennis event detection model.

In summary, while there exist works on machine learning approaches related to our work in this paper, none of them provides a scalable system that supports the entire life-cycle of tennis event detection and addresses all the challenges mentioned in Section 1. We present such a system in this paper. Although we focus on tennis, our system can be extended to other sports applications (or sound event detection applications in general) in a relatively straightforward manner.

3 TENNIS EVENT DETECTION PIPELINE

In a tennis game, two players or teams start on opposing sides of a net dividing a firm, rectangular, flat surface. The playing surface is turf, clay, concrete or a painted synthetic. A legal serve starts a rally as each player takes a turn alternately striking the ball. As the player's feet and the tennis ball impact the court surface or the player's racquet, a set of characteristic acoustic signatures are generated. Our goal is to automatically identify different tennis events and point boundaries based on the sounds recorded in the tennis game, using machine learning approaches.

3.1 Preprocessing

To classify the acoustic events from a stream of sounds recorded in a tennis match, the sound needs to be segmented into small sound windows on which a machine learning model is applied. Traditional acoustic analysis systems often partition the sound into equally-sized windows starting from the beginning of the sound recording. However, since tennis events are sparely distributed over time and usually have a very short duration, the above approach may cause the sound of interest span over two different windows, which degrades the sound classification accuracy. To tackle this problem, we implement a peak detection approach that detects the sound of interest and isolates it within a single window.

In our proposed peak detection algorithm, we first obtain the sound amplitude over time from the raw waveform of the sound. The algorithm scans for local maxima points where the slope of the amplitude changes from positive to negative as the central reference point for a sound classification task. When a peak is found, the sound is segmented on both sides for a total duration of 1 second, i.e., starting from the center of the peak, 0.5 second before and 0.5 second after the midpoint are segmented into one sound window. This resulting sound window is then used for further processing as explained below.

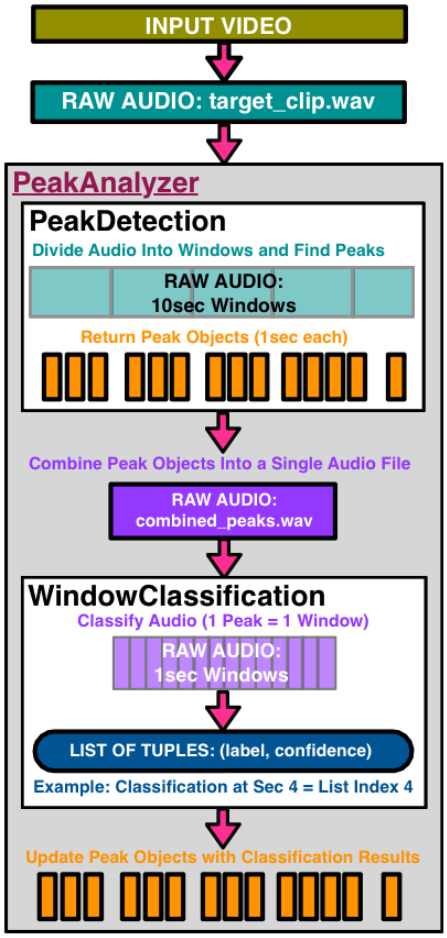

Figure 1: Peak sound analyzer.

The overall procedure of peak detection together with its connection to acoustic classification is shown in Fig. 1. The search for peaks is performed within a range of 10 seconds, where the length of this search range was determined empirically. The window length of 1 second was determined empirically as well which represents the duration of most tennis sound events.

With the 1-second windows containing potential sound events provided by the peak detection algorithm, the next step is to detect which event exists in each sound window, as well as to determine the point boundary in the tennis game. The overall pipeline of processing each 1-second sound window is shown in Fig. 2, which includes three main parts: acoustic feature extraction, acoustic classification and basic event detection, and point boundary detection. We explain each of these main building blocks in the following.

3.2 Acoustic Feature Extraction

For machine learning models to work on sounds, feature representations need to be extracted from the raw waveform of the original 1-second sound windows.

At the feature extraction phase, the acoustic signal is first stratified into time frames of 20 ms. Then, Mel-Frequency Cepstral

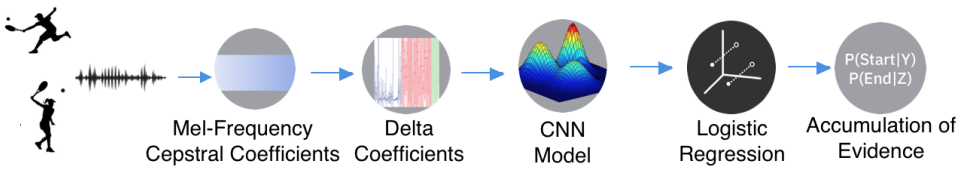

Figure 2: Acoustic pipeline for tennis event detection.

Coefficients (MFCC) features with 25 bands is computed on each 20 ms sound frame, which determines the energy level within each of the frequency regions [14, 17]. Much like human hearing, MFCC takes the energy level at a log scale; and afterwards, a Discrete Cosine Transform (DCT) is taken across the filterbanks to decorrelate the bins. At this point, the signal is described by power spectral features.

Because much of the event information is encoded within the dynamics of the sound, we also include the delta features [17] as part of the sound features, which capture the fluctuations of the acoustic signal over time. The delta features are calculated as:

$$d_t = \frac{\sum_{n=1}^{N} n\,(c_{t+n} - c_{t-n})}{2\sum_{n=1}^{N} n^2}, \tag{1}$$

where c_t denotes the original power spectral feature vector at time frame t, N is a window size to look back and forward (a typical value of N is 2), and d_t is the delta feature vector obtained at time frame t. The window size may be reduced at the first and last few frames of a sound. By applying (1) on the delta feature vector again, we obtain the acceleration (i.e., second-order delta) features. The delta and acceleration feature vectors have the same dimension as the original power spectral feature vector. They are appended to the power spectral features so that the vector encodes both static and dynamic properties of the sound.

3.3 Basic Event Detection

The feature vector of each sound window resulting from the above is used as input to a machine learning model for sound event classification. Our system primarily uses a CNN classifier together with MFCC with delta and acceleration features as explained above.

Our proposed CNN model includes 8 layers of different sizes and types in the following sequence (from top layer to bottom layer):

- 7 Softmax
- 64 Dense
- 64 Dense
- $5 \times 5 \times 48$ Convolution
- 2×2 MaxPool
- $5 \times 5 \times 48$ Convolution
- 2×2 MaxPool
- $5 \times 5 \times 24$ Convolution

The output layer gives the probabilities of 7 different labels as described below, the last dense layer uses a sigmoid activation function, the convolutional layers and the other dense layer all use the rectified linear unit (ReLU) as the activation function. This CNN architecture was empirically determined. The CNN also has $\mathcal{L}^2$ regularization, and is trained using stochastic gradient decent. Our experiments in Section 6 also consider other classifier and feature extractor combinations for comparison.

The classifier is trained to classify the following tennis events:

- Announcer – human voice of an announcement
- Applause – clapping generated by spectators
- Feet – the squeaks from the shoes worn by players
- Hit – the sound that is generated when a ball strikes the racket generated by a player's swing
- Nonplay – non-characteristic noise generated by the tennis court venue, usually the hushed white noise heard right before a player serves.
- Out – pronounced yell/call from a line judge

These classes (which we refer to as basic or atomic events) were chosen because they represent a set of typical events in a tennis game. They also provide us with enough label diversification to perform downstream time series classification for tennis point boundary detection, as explained next.

3.4 Point Boundary Detection

Tennis point boundaries specify the start and end times of a tennis point. These boundaries can be considered as advanced events compared to the basic events mentioned above, because they are related to the overall progression of the entire tennis game and we are essentially detecting the boundaries out of a time series. Point boundary detection is an extremely valuable utility for tennis coaches and players to measure fatigue, stroke effectiveness, game strategy, and opponent weaknesses.

The detection of tennis point boundaries use features derived from the basic event classifier outputs and confidence values (label probabilities) of multiple neighboring sound windows. These features are empirically determined, where the core patterns that provided the context of a start and end point was defined from tennis domain experts, and experimentation using the point boundary detection models described below uncovered additional features that included trends of sound classifier confidence. Table 1 shows the list of features used for detecting the start or end boundaries, or both.

Taking these features as input, we use two logistic regression models for detecting the start and end boundary points, respectively. The output of the logistic regression model is defined as

$$p(\mathbf{x}) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1 + \ldots + \beta_k x_k)}}, \tag{2}$$

where the vector $\mathbf{x} = [x_1, x_2, \ldots, x_k]$ denotes the input of the model (i.e., the features defined above), $\beta_1, \beta_2, \ldots, \beta_k$ are trainable parameters that represent the predictive power of each feature. We define that the start and end boundaries are triggered by a sound of hit.

Table 1: The features used for point boundary detection

Model	Features
Start	Sound duration, announcer average label probability, applause label count, duration to next label, duration to previous label, duration to previous two labels, duration between hit labels
End	Label probability, hit label average probability, nonplay label count, nonplay average label probability, next label probability, next two label probabilities, previous two label probabilities
Both	Announcer label count, applause average label probability, feet label count, feet average label probability, out label count, hit label count, out average label probability, previous label probability, current label probability, next label probability, previous two labels probability, next two labels probability

The output of the model can be interpreted as the probability of the sound of a hit defining the start time of a point or a hit indicating the end time of a point, depending on whether the model is for predicting the start or end boundary. External systems can then accumulate the boundaries of different tennis point for automated content tagging and highlight window creation.

4 SYSTEM ARCHITECTURE

If all of the professional matches in the past 13 years of tennis Grand Slams were analyzed by our system, over 1,000,000 sounds would have to be classified and 100,000's of tennis points detected. In all, over 900 hours of video and 1,000,000,000's of player and data tracking points from Hawkeye would need to be processed to discover all point events related tennis statistics. To achieve the goal of a production system that is continuously available and scalable, each of the acoustic detection components of our system needs to be designed in a distributed architecture. In this way, model, code, data, and configuration changes can be easily deployed and orchestrated to our system. We describe our design in the following.

4.1 Containerization

The core acoustic engine is created with several layers within a Docker container. The core layer contains Ubuntu. Several Unix packages are installed into a docker image that support our system. The image contains file permission configurations. At deployment, the library with multiple acoustic classification algorithms including those described in Section 3 is pulled from a git repository and compiled and built. The library has dozens of acoustic classification algorithms that are designed for the tennis domain. In addition, a workbench application that includes a web-based user interface enables the management of annotated sounds and trained models (see Section 5.2).

A RESTful webservice is exposed to post sound files and videos to our system. The work is distributed and routed first to the models for basic event detection and then to the models for point boundary detection. The results are stored within a configurable data store

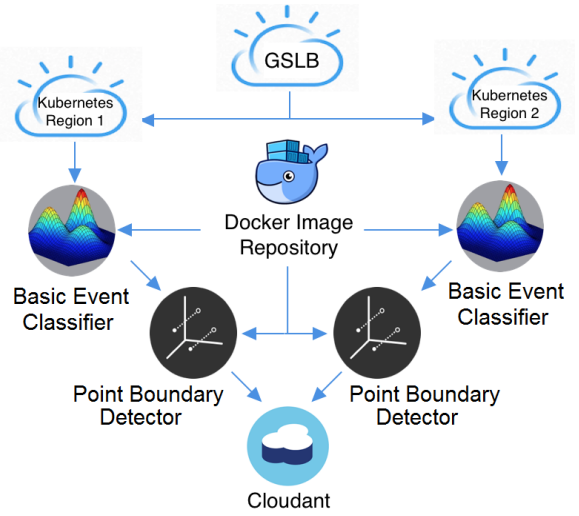

Figure 3: Overall system architecture.

such as Cloudant [2] that can be retrieved through a RESTful interface from our application. The application in the form of a Docker image can be deployed as a service.

4.2 Tennis Event Detection as a Service

As shown in Fig. 3, Kubernetes technology exposes our system as a scaled and distributed service in the cloud. The Docker image is pushed to a Kubernetes image library. The image can be shifted and pulled to any region on any cloud. Any number of Kubernetes pods with a desired set of workers are rolled out to the cloud with our image. After we specify our node port, our system is publicly available for tennis event classification. A global server load balancer (GSLB) can distribute traffic across different regions to Kubernetes clusters. Kubernetes then routes the traffic to workers. A shared data store such as Cloudant stores the results of our system.

A client posts the sound files to the Domain Name Service (DNS) that populates the request to our service. The basic event and point boundary detection functionalities are ready for any consumer. Any changes to models, configuration, or code are rolled out to the service by updating the docker image.

5 USER INTERFACE

5.1 Data Labeling Tool

The machine learning models presented in Section 3 need to be trained on labeled tennis sounds first, before they can be used in analyzing tennis sounds with unknown labels. It would be too time-consuming for a human to watch (and listen to) entire tennis match recordings where events only occur relatively sparsely in the recording. To assist the human labeling process, we developed an annotation tool as shown in Fig. 4. The tool makes use of our peak detection approach described in Section 3.1. It ingests a selected match's video and audio footage and shows (and plays) 1 second duration video (and audio) segments centered around sound peaks to the user. If applicable, the user selects a checkbox marking the start or end of a tennis point and the most appropriate basic event

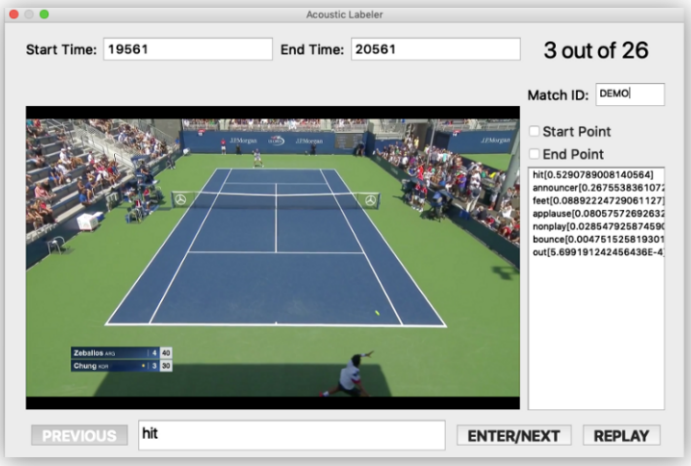

Figure 4: Data labeling tool.

label from a predefined list, for the currently shown video and sound segment. Then, the 1 second sound is extracted from the video, named with the selected label and an incremental sequence number, and saved as a file.

As the labeled dataset grows, a model is trained and integrated with the labeling tool, so that the tool can suggest a sound label for unseen samples. The user has the option to accept or override this suggestion. This annotation process enabled a large dataset to be accumulated very quickly. In our user experiment, a large collection of over 20,000 labeled sounds was obtained within 5 days.

5.2 Sound and Model Management Workbench

The annotated sound dataset generated using the labeling tool described in Section 5.1 is sent to the system for training models for basic event detection and point boundary detection. As the dataset grows (e.g., recordings of new tennis games may be added over time), new models are trained possibly using different datasets and for different purposes. There may also exist sounds in the dataset which were mislabeled at first and need to be corrected later. In addition, sounds that are recorded by the user when using the system may represent some interesting event instances that the system did not know or gave a label with low confidence.

In all the above cases, the user may need to re-label, delete, or add sounds into the training dataset; the user may also need to delete or re-train existing models, or train new models on an updated dataset. The workbench shown in Fig. 5 is the web-interface that allows the user to perform these operations. Some details are explained as follows.

- The "Manage Sounds" tab manages all the sounds in the current training dataset. Here, each sound can be played, its labels can be edited, or the entire sound can be deleted. Note that each sound can have multiple categories of labels (such as "event: hit; sport: tennis") if models for distinguishing different aspects of sounds need to be trained. Although our pipeline described in Section 3 currently works on a single label category, this design allows us to extend the system to more complex models and use cases in the future. In addition, multiple label categories are also useful in the search for specific categories of sounds. For example, the database may store sounds of different sports and the user may want to train a model only for tennis sounds.
- The "Review" tab keeps all the sounds for which an event has been detected, where the sounds are uploaded by the user when running the system for event detection using pre-trained models (i.e., during the testing phase). It allows the user to label sounds that the system has determined to be unknown and add them to the training dataset. It also allows the user to listen to sounds and check the correctness of the labels provided by the system; if the user verifies that the label is correct (or provides a correct label if it was labeled incorrectly by the system), the sound can be added to the training dataset as well.
- The "Classify" and "Train" tabs are for model management, where a pre-trained model can be used to test against sounds uploaded to the workbench directly in the "Classify" tab, to examine its accuracy on a test dataset provided by the user. Under the "Train" tab, new models can be created (trained) using all or a selected subset of the training data, and unused models can be deleted.
- The "Visualization" tab visualizes the correlations among different labels based on the distances of sounds with different labels in the feature space. This allows the user to explore which sets of labels correspond to similar sounds and which sets of labels correspond to very different sounds, to obtain insight on why some sounds may be more difficult to distinguish than others.

6 EXPERIMENTS AND RESULTS

6.1 Setup

In the experiments, videos captured from a series of tennis matches on a concrete surface at the same overall acoustic venue environment were used. The acoustic dataset for training and testing was annotated from a variety of professional tennis matches in the US Open Series. This dataset provided a high degree of diversity of sound within the context of a tennis match on a given surface. Only hard-court matches were selected from a variety of very large, large, medium, and small indoor and outdoor venues. The dataset had diversity of ambient noise. The production quality of the match footage and sound also varied from commercial broadcast quality to remote controlled robo camera production. No voice over commentary was used. Many times, there was overlap with the individual noises such as a ball strike event being a combination of feet shuffling, grunting, and the sound of a ball being struck with a racquet.

The original unlabeled tennis video footage was labeled by a human user using our data labeling tool described in Section 5.1. The duration of each sound was 1 second to match with the usual length of a tennis hit sound, as explained in Section 3.1. The set of basic events described in Section 3.3 was used as labels. The match identifier along with timestamp was used to track all of the labels within a temporal sequence. After some clean-ups and

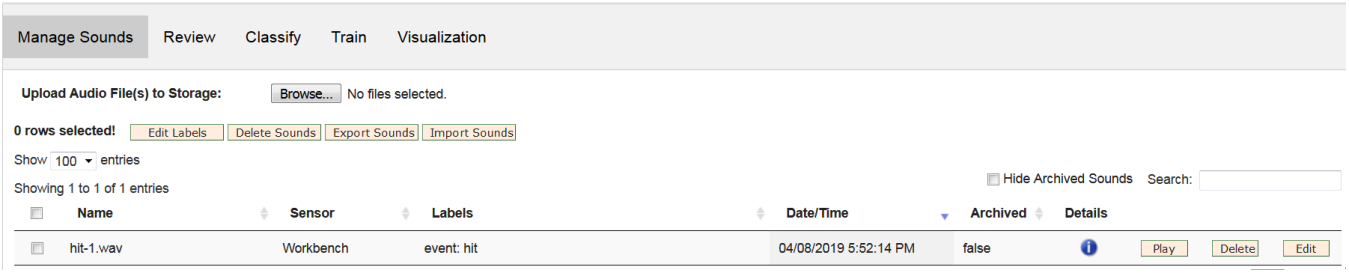

Figure 5: Sound and model management workbench.

Predicted ->	announcer	applause	feet	hit	nonplay	out
announcer	95.86%	1.45%	0.38%	0.51%	1.12%	0.68%
applause	1.83%	89.02%	1.34%	1.02%	6.26%	0.53%
feet	0.45%	0.86%	93.60%	1.10%	3.58%	0.41%
hit	0.99%	1.64%	2.73%	90.60%	2.73%	1.31%
nonplay	1.46%	4.58%	3.67%	1.81%	87.87%	0.61%
out	0.48%	0.03%	0.51%	0.57%	0.19%	98.22%

Figure 6: Confusion matrix of the proposed CNN + MFCC-Delta-Acceleration model.

balancing among the number of sounds for different labels, the assembled dataset included 6,568 individual tennis sounds, which is equivalent to a total duration of 1.83 hours of annotated sounds that were available for machine learning tasks. The class distribution and total duration of different types of sounds are summarized in Table 2. Throughout the machine learning tasks, cross-validation with 4 folds was used with this entire dataset.

Table 2: Distribution of acoustic classes

Class	Count	Duration (min:sec)
Announcer	3383	56:23
Applause	2842	47:22
Feet	2455	40:55
Hit	3596	59:56
Nonplay	3710	61:50
Out	3137	52:17

6.2 Basic Event Classification Results

To evaluate basic event classification, we compared our proposed approach of CNN with MFCC-Delta-Acceleration features with several shallow models including GMM, nearest neighbor, and SVM, as well as feature extraction mechanisms including log-mel energy with or without delta/acceleration. The log-mel energy feature is MFCC without the last step of DCT transform. Some of these shallow models we use in comparison here are also used in the related work as described in Section 2.

The GMM model used in comparison had 8 Gaussians in the mixture model. A separate GMM is trained on data corresponding to each label. The probabilities of different labels in the test phase is computed from the likelihood values provided by each GMM, which is the typical approach in audio classification. For the nearest neighbor model, we chose an $\mathcal{L}^p$ norm distance metric with $p = 0.5$ for distance computation, which, strictly speaking, is not a real distance metric because the triangle inequality does not hold with this value of p, but it performed the best according to our empirical study. The SVM model was defined with a linear kernel.

The precision, recall, and F1-score of several top performing combinations of models and feature extractors are shown in Table 3. We see that our proposed CNN model with MFCC plus delta and acceleration features achieved the best overall performance with an F1-score of 92.39%. The confusion matrix for the proposed CNN + MFCC-Delta-Acceleration model is shown in Fig. 6. Generally, sounds that are clearly distinguishable from other classes have higher classification accuracies, whereas sounds that sound similar to other classes have lower accuracies, as one would intuitively expect.

Note that all these results were obtained together with the peak detection algorithm described in Section 3.1. The peak detector worked well because the tennis event sounds are loud and sudden.

6.3 Point Boundary Detection Results

The prediction of the start time of a tennis point with our proposed approach described in Section 3.4, based on the output of our CNN + MFCC-Delta-Acceleration model, had a precision of 84% and a recall of 82%. The end time prediction had a slightly lower precision of 77% and a recall of 77%. Many of the sounds for the start or end of a point overlapped. For example, the start of a point could occur when ball was hit and the crowd was cheering. However, at other times, the sound of a ball hit could be occluded by applause or nonplay noise. The constructive sounds during the point event decreased the point boundary detection performance. The look-ahead and

Table 3: Basic event classification results of different models and feature extractors

Model Type	Precision %	Recall %	F1 %
CNN + MFCC	90.01	89.89	89.88
CNN + MFCC-Delta-Acc	**92.47**	**92.40**	**92.39**
GMM + MFCC-Delta	78.41	77.53	77.50
GMM + MFCC	76.65	75.76	75.87
GMM + Log-Mel	72.67	71.83	71.61
Nearest Neighbor + MFCC	72.59	68.53	68.57
Nearest Neighbor + Log-Mel	73.64	70.21	70.26
SVM + MFCC	37.19	20.12	23.94

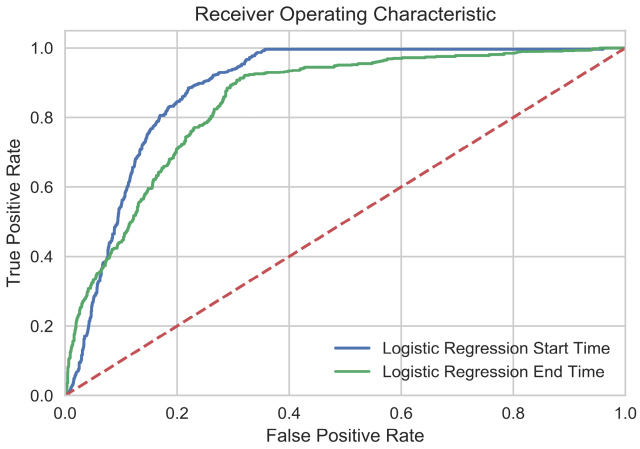

Figure 7: The ROC curve of point boundary detection.

look-back features in Table 1 of Section 3.4 were designed to help deconstruct the time series patterns.

The duration or time between labels was not as important for the end boundary detector as it was for the start boundary detector. For example, the duration between two closest tennis hit labels was very important for the start boundary detector. Intuitively, the timing between a serve and return has small variability. The label confidences (probabilities) from the sound classifier were more predictive for detecting the end boundary. If the confidences of next two labels and previous two labels from the sound classifier were above 50%, the sound was more likely an end boundary marker. The Receiver Operator Characteristic (ROC) curve in Fig. 7 depicts the performance of our point boundary detection approach. Marking the end of a point is slightly more difficult than marking the start of a point, because there are usually more overlapping sounds at the end of a point than at the start of a point.

6.4 Real-World Deployment

At the US Open 2019 (a tennis grand slam event), the detection of sound-based events and point boundary classification provided acoustic experiences for tennis fans. For example, we classified tennis sounds to determine streaming video cuts for the automatic generation of tennis highlights. The system processed hours of live tennis match data from 7 courts to create tennis highlights that

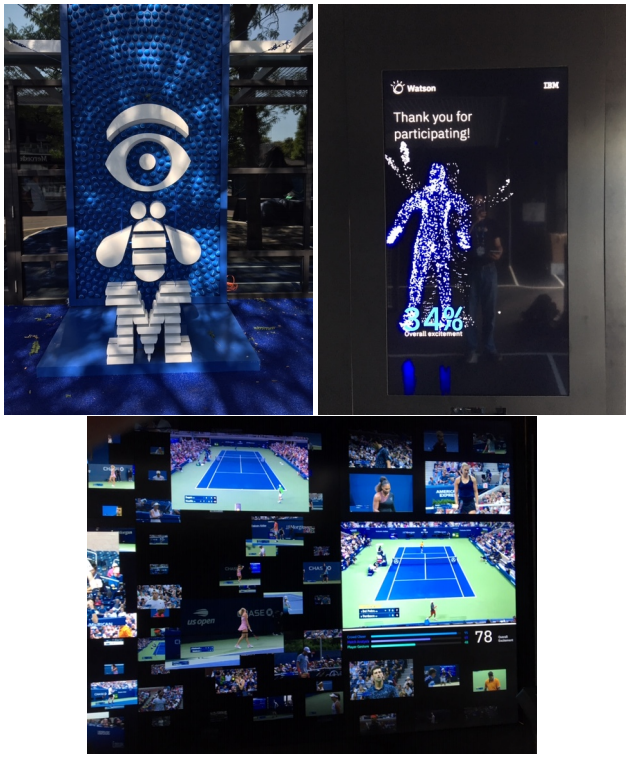

Figure 8: Acoustics used in IBM US Open Experience.

were distributed around the world. In addition, thousands of people interacted with interpreted sounds within highlights (see Fig. 8). At the IBM experience attached to the Aurthur Ashe Stadium, tennis fans could create their own highlight by moving in front of a camera and speaking into a microphone. Deep learning algorithms that are similar to those presented within this paper were used to rank the excitement levels of the participant.

7 CONCLUSION

In this paper, we have presented a system for detecting tennis events from acoustic data. It includes tools to support data labeling and sound/model management by users, and a complete pipeline for tennis event detection. The system is provided as a cloud service that can span over a large geographical area. Experimentation results confirm the effectiveness of our proposed system. Our system can be used for automatic detection of tennis highlights with event markers. In addition, results provided by our system can give additional meta-data and indexes into tennis videos to provide tennis players with more data and insights as they train and prepare for matches.

ACKNOWLEDGMENT

We would like to thank Jack Frost, Keith Grueneberg, John Kent, Kloey, Bong Jun Ko, Noah Syken, Lee Tilt, Xiping Wang, and David Wood who participated in developing the acoustics system.

REFERENCES

[1] 2018. TUT Acoustic Scene Classification. (2018). https://www.kaggle.com/c/acoustic-scene-2018

[2] 2019. Cloudant. (2019). https://www.ibm.com/cloud/cloudant

[3] 2019. Hawk-Eye Line-Calling System. (2019). https://www.topendsports.com/sport/tennis/hawkeye.htm

[4] Anthony Bagnall and Jason Lines. 2014. An experimental evaluation of nearest neighbour time series classification. *arXiv preprint arXiv:1406.4757* (2014).

[5] Anthony Bagnall, Jason Lines, William Vickers, and Eamonn Keogh. 2018. The UEA & UCR time series classification repository. *URL http://www. timeseriesclassification. com* (2018).

[6] Selina Chu, Shrikanth Narayanan, and C-C Jay Kuo. 2009. Environmental sound recognition with time–frequency audio features. *IEEE Transactions on Audio, Speech, and Language Processing* 17, 6 (2009), 1142–1158.

[7] Wei Dai. 2016. Acoustic Scene Recognition with Deep Learning. (2016).

[8] Tom Decroos, Vladimir Dzyuba, Jan Van Haaren, and Jesse Davis. 2017. Predicting soccer highlights from spatio-temporal match event streams. In *Thirty-First AAAI Conference on Artificial Intelligence*.

[9] Helenca Duxans, Xavier Anguera, and David Conejero. 2009. Audio based soccer game summarization. In *2009 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting*. IEEE, 1–6.

[10] Hassan Ismail Fawaz, Germain Forestier, Jonathan Weber, Lhassane Idoumghar, and Pierre-Alain Muller. 2018. Deep learning for time series classification: a review. *arXiv preprint arXiv:1809.04356* (2018).

[11] Shayan Gharib, Honain Derrar, Daisuke Niizumi, Tuukka Senttula, Janne Tommola, Toni Heittola, Tuomas Virtanen, and Heikki Huttunen. 2018. Acoustic Scene Classification: a competition review. In *2018 IEEE 28th International Workshop on Machine Learning for Signal Processing (MLSP)*. IEEE, 1–6.

[12] Md Afzal Hossan, Sheeraz Memon, and Mark A Gregory. 2010. A novel approach for MFCC feature extraction. In *2010 4th International Conference on Signal Processing and Communication Systems*. IEEE, 1–5.

[13] Qiang Huang and Stephen Cox. 2010. Hierarchical language modeling for audio events detection in a sports game. In *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2286–2289.

[14] Xuedong Huang, Alex Acero, Hsiao-Wuen Hon, and Raj Reddy. 2001. *Spoken language processing: A guide to theory, algorithm, and system development*. Vol. 1. Prentice hall PTR Upper Saddle River.

[15] Juncheng Li, Wei Dai, Florian Metze, Shuhui Qu, and Samarjit Das. 2017. A comparison of deep learning methods for environmental sound detection. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 126–130.

[16] Lie Lu, Hong-Jiang Zhang, and Hao Jiang. 2002. Content analysis for audio classification and segmentation. *IEEE Transactions on speech and audio processing* 10, 7 (2002), 504–516.

[17] James Lyons. 2015. Mel frequency cepstral coefficient (MFCC) tutorial. *Practical Cryptography* (2015).

[18] Michele Merler, Dhiraj Joshi, Quoc-Bao Nguyen, Stephen Hammer, John Kent, John R Smith, and Rogerio S Feris. 2017. Automatic curation of golf highlights using multimodal excitement features. In *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. IEEE, 57–65.

[19] Michele Merler, Khoi-Nguyen C Mac, Dhiraj Joshi, Quoc-Bao Nguyen, Stephen Hammer, John Kent, Jinjun Xiong, Minh N Do, John R Smith, and Rogerio Feris. 2018. Automatic Curation of Sports Highlights using Multimodal Excitement Features. *IEEE Transactions on Multimedia* (2018).

[20] Annamaria Mesaros, Toni Heittola, Aleksandr Diment, Benjamin Elizalde, Ankit Shah, Emmanuel Vincent, Bhiksha Raj, and Tuomas Virtanen. 2017. DCASE 2017 challenge setup: Tasks, datasets and baseline system. In *DCASE 2017-Workshop on Detection and Classification of Acoustic Scenes and Events*.

[21] Asa Oines, Eugene Weinstein, and Pedro Moreno. 2018. Hybrid Lstm-Fsmn Networks for Acoustic Modeling. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 5844–5848.

[22] Ke Zhang, Wei-Lun Chao, Fei Sha, and Kristen Grauman. 2016. Video summarization with long short-term memory. In *European conference on computer vision*. Springer, 766–782.