An Emotional Sports Highlight Generator*

Margaux Cavagna[1], Ahmed Abdel Rahman[1], Robert Cinciuc[1], and Bruno Sader[1]

INSA de Lyon, Computer Science Department, 69100 Villeurbanne, France

Abstract. We developed a framework that extracts football games' highlights and generates a ready to use video clip using audio and video features. For that purpose, we first implemented different audio and video analysis techniques as basic building blocks and then combined them to build the optimal solution. The results are very promising and show that our highlights could compete against human-generated highlights. This solution could allow small competitions to be able to generate highlights for their games, given that they record it. The source code can be found on our github repository.

Keywords: Machine Learning · Emotion Classification · Audio Features.

1 Introduction

Football highlight extractors are more and more frequent in this day and age. Most, if not all, focus on key moments, whether they be goals, bookings or substitutions and use external sources to accurately create them. In our paper, we study the usage of multimodal models applied to TV streams in order to automatically generate highlights based on "emotional" events. This means that we don't only focus on traditional highlights such as goals or dangerous opportunities, but also alternative ones like fan chants or even "lowlights" such as red cards or missed open chances depending on the reaction of the audience.

To do so, we have compared different approaches for each mode (be it audio or video), studied different model combinations using the basic analysis bricks created in the previous step and determined which one would work best for our use case. In order to score each model, we evaluated them on :

- Accuracy
- Cost of operation
- Generalisability and transposability
- Ease of implementation

We found out that the best combination is a clever mix of audio and video analysis using sound classification, scene change detection and scoreboard text recognition. This combination was then translated into a processing pipeline

* Supported by Ignifai, Liris and Icar.

that takes a game's broadcasting stream as input and returns the corresponding highlights. In Section 2, we present the theoretical knowledge behind each audio/video analysis approach. The implementation of each analysis module is presented in Section 3. Section 4 highlights how the pipeline was constructed. Finally, we present the results of our work in Section 5 and conclude in Section 6.

2 Theoretical background

Our previous works included a study of the state of the art [5] regarding highlight generation in various sports. Based on it, we have decided to implement the techniques that seemed most suited to our use case in accordance with the available resources. In this section, we will briefly explain how these techniques work. For more detailed information, please refer to the original study.

2.1 Audio

We've decided to use two ways of extracting information from the audio source: analyzing the audio characteristics themselves or extracting the text from the speech of commentators and performing text mining.

Audio Classification The idea behind spectral analysis is to extract features from the audio in order to determine the class which is being played. The class may give us insights into whether it represents something of value for our highlight generation. Thus, the audio workflow is composed of three steps: extracting features from the audio, classifying audio segments and separating important classes for highlight generation. First, the audio characteristics can be exploited in two different ways: by extracting features directly from the raw audio or by employing the first order differential coefficients of the Mel Frequency Cepstral Coefficients (MFCC). These differential coefficients are the amplitudes of the spectrum extracted after multiple transformations including Fourier, log and discrete cosine.The first approach consists in dividing the audio from the match into small analysable portions that will be used directly as an input for a classification algorithm. The second method captures the dynamics of the audio, such as the variation of MFCC coefficients with time and uses them as the primary input of the classification algorithm.

Secondly, the classification can be done using a wide variety of algorithms. For this paper, we chose a Hidden Markov Model to be paired with the features extracted from MFCC and a Convolutional Neural Network that analyzes the raw audio. We wanted to perform a comparison between these two methods to see which one is more efficient. Regardless of the algorithm, there is a training step involved in the classification, which is done prior to the execution of the workflow on a particular test match. The output of the algorithm for a segment of audio is a class label among the set of all labels it was trained on. The class labels in our context could be very diverse: silence, applause, whistle, excited commentators' speech, unexcited speech, music, etc. Meanwhile, it must be pointed that the number of classes plays an important role in the complexity. More classes means more data examples are needed to train the classifier.

Another novel approach we decided to study was the implementation of a convolution neural network (CNN). Most audio classification techniques use 2D CNNs and are trained on mel-spectrograms. However, in [4] a model based on a 1 dimensional CNN for environmental sound classification is proposed. The main benefits of this approach is the small parameter space compared to other architectures and drops the signal processing component (calculating MFCC for example) thus is easy to implement and train.

Finally, we perform the selection of pertinent classes. Our work is based on the hypothesis that a strong emotional response from the commentator and the crowd correlates with important moments in the match. Thus, only audio segments of such class labels are selected for generating highlights. When the commentators and the crowd are happy, sad, excited, neutral or angry their reactions create distinguishable audio characteristics that allow for discrimination between important and uninteresting situations. The final contribution of the audio processing is pointing moments that created a strong emotional reaction from both parties.

Speech Analysis This method consists in analysing the commentators' speech to identify their mood by examining the words they use. To do so, we use a Natural Language Processing (NLP) model, combined with a labelled dataset from football games to train it. Just like for the previous approaches, the model is then able to recognize interesting events in the game we want to analyse by looking for particular words like "outstanding", "dangerous", "bad", etc in the commentary and assign in to the different available classes (excited commentator speech, unexcited speech,...). This can be done by parsing the audio directly but the necessary tools for this are not very efficient, so an alternative is analysing the textual transcript of the speech. Another significant parameter concerning this method is the preprocessing step as the noise levels can be very high during football games (crowd, whistle, jingles,...) that can skew the analysis.

2.2 Video

Video analysis is very useful for us as it allows us to gather information and clues that are not necessarily noticeable in the audio stream. Hence, we decided to focus on two different methods: scene change detection and scoreboard text recognition.

Scene detection is a valuable tool when generating a highlight, it enhances the viewing experience. Using the detected segment, we can generate a more cohesive highlight. Often camera operators in sport will follow an action with a close up of a player, of a celebration, of a brawl etc... We can therefore use this clue to extract either the previous action or the replay. Detecting camera changes can also help us cut the video at appropriate timestamps, and not in the middle of the action. In [1], Xu and al. use histograms to isolate logo transitions (a color histogram is a representation of the distribution of colors in an image). Using their approach, we can easily detect a sequence change by comparing the last two adjacent frames. Therefore, an important and sudden variation in the color histogram usually implies that the scene has changed. If the difference, computed using the correlation in between two histograms 1 (implemented in opencv), is higher than our threshold, we conclude that the scene has changed.

$$
\begin{aligned}
&d(H_1, H_2) = \frac{\sum_I (H_1(I) - \bar{H_1})(H_2(I) - \bar{H_2})}{\sqrt{\sum_I (H_1(I) - \bar{H_1})^2 \sum_I (H_2(I) - \bar{H_2})^2}} \\
&\text{where} \\
&\bar{H_k} = \frac{1}{N} \sum_J H_k(J) \\
&\text{and N is the total number of histogram bins.}
\end{aligned}
\tag{1}
$$

Scoreboard text recognition Recognising a scoreboard is very useful because we are able to read the score and time from it. Moreover, the scoreboard isn't shown on screen during replays. So, we can track key moments by recognizing when the scoreboard isn't on screen (since it means that an action is being replayed). The task of recognizing a score bar has been done by Shukla and al. in [2]. Using an OCR (Optical Character Recognition) classifier, they locate the positions of the digits on the screen using bounding boxes. They train the classifier to recognize digits and dashes, which are present inside the scoreboard. They were then able to extract the digits from the scoreboard and analyse them.

Recognising a scoreboard is very useful because we are able to read the score and time from it. Moreover, the scoreboard isn't shown on screen during replays. So, we can track key moments by recognizing when the scoreboard isn't on screen (it means that an action is being replayed). The task of recognizing a score bar has been done by Shukla and al. in [2]. Using an OCR (Optical Character Recognition) classifier, they locate the positions of the digits on the screen using bounding boxes. They train the classifier to recognize digits and dashes, which are present inside the scoreboard. They were then able to extract the digits from the scoreboard and analyse them.

3 Implementation and evaluation of the analysis bricks

In this section, we discuss how we implemented the methods presented in section 2 and the individual results they produced, which allowed us to design the optimal processing pipeline (see section 4).

3.1 Database creation

Before implementing the classification model, we first had to choose the set of classes that would be useful for our context. We decided to let our classifier operate with the following classes : ambient sound, crowd chanting, excited commentary and unexcited commentary. The choice of the last two classes is fundamental because the scream of the commentator is the most powerful audio characteristic of a powerful moment in the game. The choice to add an ambient class label along a crowd cheering one is to help the model distinguish between the constant humm of the stadium and the important chanting of spectators during emotional moments. The next step was the creation of the database on which we would train our classification models. We used the audio from the football matches we found in an open database. The process of associating class labels to audio segments had to be done by hand as we had no other source of data. The process consisted of importing the entire audio from the match into Audacity and exporting small segments corresponding to the classes we were looking for. We have tested two versions of this database: one with audio segments of different length (on average 5 seconds) and the other one with all the audio segments split into 1 second cuts, thus normalizing the length on all examples. Upon testing, we concluded that the second database provides more accurate class labels for any audio length.

3.2 Hidden Markov Model classification

We now had to create and train the Hidden Markov Model. The class created for this purpose iterates through all the audio files from the database directory and extracts the MFCC features from each once. Next, it creates a HMMTrainer instance (imported from the hmmlearn library) with 10 components (through testing, this number of components yielded the best results) and trains it on the array of MFCC extracted features. The model is saved in a separate file for later use in testing. The testing for an audio segment is done by first reading the HMM model from the file, then by extracting the MFCC features from the corresponding audio and computing a score for each of the classes present in the HMM. The audio segment is associated with the class with the highest score. The length of the audio segment given to the classifier should be of the same duration as the training examples for a good accuracy. Thus, because our database is composed of audio segments of exactly one second, we feed the classifier audio segments of this length. Now that every second of the audio is classified into its appropriate category, we apply a selection algorithm written by us based on our practical findings on how to better select important moments. We analyzed official highlight videos and compared them with their respective full matches to understand the temporal relation between scenes that make it to the final product. The exact way of selecting scenes based on classes found inside their time length is explained in the section 4 (Pipeline construction). An advantage of this technique over the visual ones is that the probability of choosing a scene is directly proportional to the emotional audio response of the crowd and the commentators. Thus, a scene that visually doesn't give clues of it being important (the zoom is too far away to be used for computer vision), it still may indicate that an action is happening and that it may be important for the highlight generation. Meanwhile, our technique is the result of very subjective emotional responses. Because of this, the exact same match video would yield different results depending on where the teams are playing (which football field holds the game). We can counter these subjective downsides with the help of our visual methods described in the paragraph 5 of this section.

3.3 Signal analysis and 1D convolutional neural network

We've also built the aforementioned neural network in order to compare the efficiency, the accuracy and the generalisability with the previous model. Our architecture (seen in figure 1 and detailed in table 1) consists of 3 1D CNN and 3 fully connected layers. It is very easily trained (using our database) and produces create results.

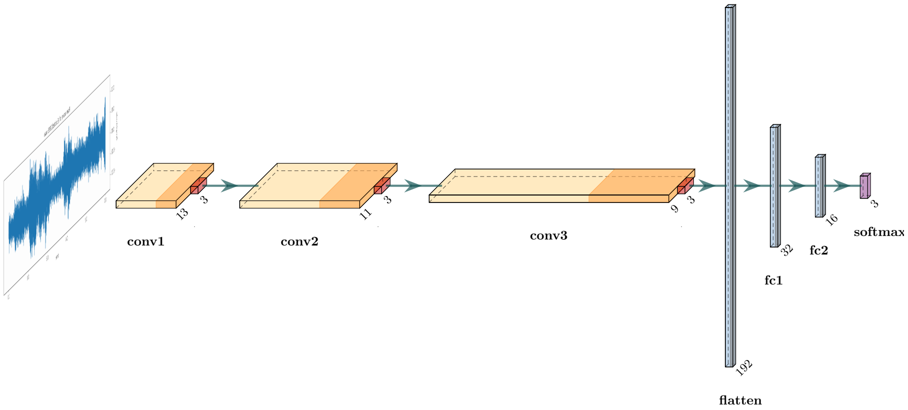


Fig. 1: Neural network used to analyse audio input (44100,1). Note : Depth represents the number of filters

Layer	Kernel	Filters	Activation	Input	Output
Input				(44100,1)	(44100,1)
Conv1	13	8	ReLu	(44100,1)	(4409,8)
MaxPool1	3			(4409,8)	(1469,8)
Dropout	0.3			(1469,8)	(1469,8)
Conv2	11	16	ReLu	(1469,8)	(292,16)
MaxPool2	3			(292,16)	(97,16)
Dropout	0.3			(97,16)	(97,16)
Conv3	9	32	ReLu	(97,16)	(18,32)
MaxPool3	3			(18,32)	(6,32)
Dropout	0.3			(6,32)	(6,32)
Flatten				(6,32)	192
Dense1			ReLu	192	32
Dropout	0.3			32	32
Dense2			ReLu	32	16
Dropout	0.3			16	16
Dense3			Softmax	16	3

Table 1: Architecture used for the neural network

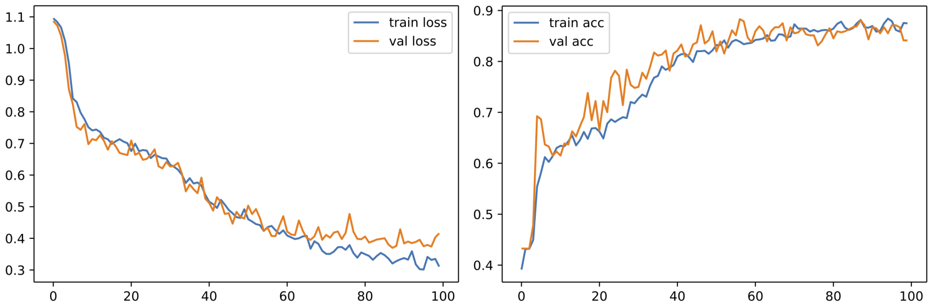


Fig. 2: Loss and accuracy after epoch 100

Looking at figure 2 we can confidently say that our model is learning correctly and that not overfitting. Note that we tried a more complex model but our loss was much higher and we noticed some overfitting. As mentioned earlier, the benefit of using this model is the small parameter space and the lack (low amount) of prepossessing needed.

Finally analysing the confusion matrix (figure 3) produced by our model, we can see that our model classifies correctly each class over 82% of the time. We can note that a sample classified as crowd may not be interesting for the highlight generation (see section 4) thus an 11% confusion rate between labels "Crowd" and "Unexcited" is not much of a problem.

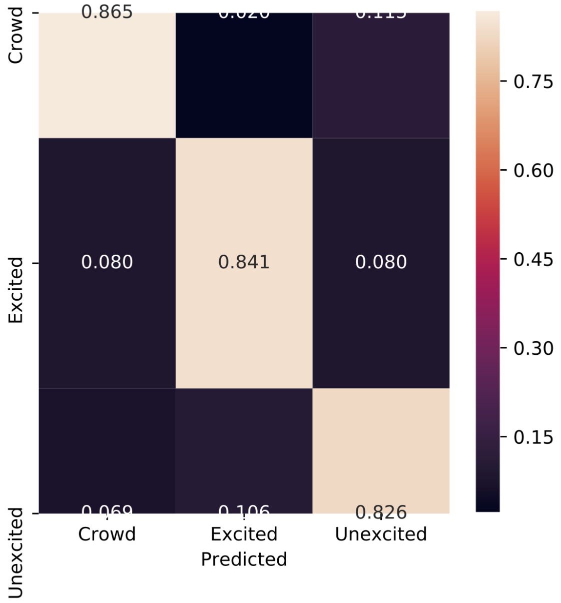


Fig. 3: Confusion matrix

3.4 Speech Recognition

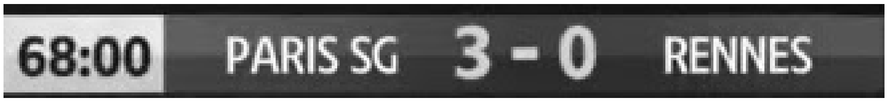
After studying the feasibility of such a method (see section 2.a.ii), and deciding that we were going to focus on french commentary, we divided the practical implementation in various steps. On the one hand, we had to preprocess the audio to remove the background noise. We manage to do it by using the noisereduce library, based on Audacity's noise reduction effect. It takes two inputs, the audio we want to clean (called "signal") and an audio clip of the noise we want to remove (called "noise"). The filtered audio file is then generated by applying a mask, based on the Fast Fourier Transformations (FFT) of the noise and of the signal, on the original clip. Then, we exploited the available functions of the SpeechRecognition library and Google's Cloud Speech API to convert these chunks into text. Each transcript is then written in a .txt file temporarily until the processing is complete. On the other hand, we manually labelled commentaries from three different 2016-2017 Ligue 1 games to create seven different classes (positive, negative, excited, sad, angry, mixed and neutral). With this dataset, we managed to train a CamemBERT model (NLP pretrained model focused on French language). It allows us to assign a class to the analysed commentary. We made the choice to filter out all sequences that are assigned to the mixed and neutral categories as it is difficult for us to know if the play is interesting or not based on this information only. Unfortunately, because of the noise levels that can go very high during football games, and also because noise reduction solutions are not very effective in this use case, the results were not very successful. The speech-to-text tool is not able to identify any word when the crowd are chanting/shouting, which counteracts our whole concept of keeping the emotional sequences. In fact these moments, that are not analysable with this brick, are usually the most interesting ones.

3.5 Scoreboard text recognition

In order to implement the scoreboard text recognition, we chose to use an OCR (Optical Character Recognition). We based our implementation on work done by Bence Kővári [6]. We used the very popular tool pytesseract. Pytesseract is a python based optical character recognition, that is able to read and recognize the text embedded in images. It is a python wrapper for the open-source Google-founded OCR tool Tesseract. We use openCV to read images from the video and apply transformations to these images.

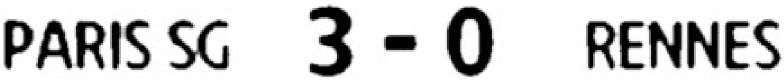
The scoreboard text recognition works in the following way. It is based on real-time analysis : we conduct the analysis as the match is playing.

- A screenshot of the match image is taken every second throughout the match
- The scoreboard is selected from the screenshot and saved as another image
- We then split the scoreboard into time and score images
- We apply Gaussian transformations (such as Gaussian Blur, to reduce image noise and detail) and Morphological Transformations (such as erosion, to remove noise, and dilatation, to dilate the eroded image back to its original size) to the images in order to make them more readable by the OCR.
- We feed the images to pytesseract as input. The OCR is then able to extract the text from the scoreboard and we can later interpret it in our code.



68:00 PARIS SG 3 - 0 RENNES

Fig. 4: The scoreboard being extracted from the video



PARIS SG 3 - 0 RENNES

Fig. 5: The score image after the transformations. This is the image that pytesseract is going to read

We are doing this process throughout the whole match, second by second. We write in a separate text file the values read by the OCR on the scoreboard and the corresponding video timestamp, so we can easily cut the video at the appropriate timestamps afterwards. During a replay or an interesting moment, the scoreboard disappears from the screen and our code doesn't detect a scoreboard. Based on the length of the interruption (= the length of time during which the scoreboard disappears and our OCR is unable to read a score or a time), we are able to extract interesting moments from the video. By writing the times to a separate text file and conducting the analysis after the match ends, we are given flexibility to adjust our highlights to our needs.

Using this technique only to isolate highlights from the match gives us very complete and interesting results, depending on highlights length. For example, on a 15-minute extract of a football game, with a highlight length of 10 seconds, the OCR is able to extract 5 interesting moments : 2 missed goal opportunities, 2 fouls including one resulting in a yellow card and 1 goal. If we increase the highlight length to 15 seconds, we get one of the opportunities and the goal. If we increase highlight length to 20 seconds, we only get the goal.

We can therefore conclude that the OCR is very efficient in extracting key moments from the match, and that it is highly flexible according to our needs.

4 Pipeline construction

In this section, we explain how we designed the whole pipeline from the basic building block we had and their different characteristics and results. We want our solution to select the most emotional moments of the game while having a reasonable execution time to have a significant advantage on the traditional highlight generation techniques. So we had to choose the blocks that gave the best results and assemble them to optimum effect.

Note that we simulate the input stream behavior by reading sequentially the video file from our directory, frame by frame. We then needed a pertinent way of diving our video into smaller analyzable segments. We applied the scene change detection algorithm to follow the dynamics of the in-match camera and preserve the continuity of the action inside each scene. This algorithm returns a list of scenes and passes it to the next algorithm in the pipeline (the audio classification).

The Hidden Markov Model explained in section 3 receives the list of scenes generated by the previous algorithm. This truncation style of the audio allows us to be more aligned with the pace of the actual match. For instance, it allows us to accept as a highlight a couple of seconds which alone seem uninteresting, but become important if surrounded by interesting moments inside the same scene. After classifying each second of all passed in scenes, the algorithm decides at first which scenes to save. Analyzing the classification of our algorithm, we decided that the only class that's helping generate qualitative highlights is "Excited Commentary". The "Crowd Cheering" class was not selected because it created a very heterogeneous behaviour from a match to another. By analyzing multiple football matches, we deduced that only scenes that end on an interesting moment are worth saving. Whether it is a goal, a fault or a missed chance, the commentator shouts towards the end of the scene and the camera quickly shifts towards the close-up of the player responsible for the action. If the shouting happens in the middle of the scene, then most likely the action didn't result in anything and there is an immediate counter attack. In the meantime, after every goal, there is a lot of celebration and shouting from all the sources that may last for a long time. Hence, the decision to save only the first part of a longer important action spanning across multiple scenes. Thus, we obtain shorter highlights that capture the action of the game but are free of the visual shots of the stadium, players' celebration and redundant slow-motion replays. Another small improvement in keeping the length of the highlights short was to cut the beginning of a scene if its own beginning is uninteresting. This is done only when a long scene corresponds to the beginning of an important and long action. We cut directly to a number of seconds before the actual exciting seconds arrive (3 seconds before in our case). Finally, we eliminate exciting segments (of possibly multiple scenes) when their contiguous length is below a certain threshold (5 seconds in our case). We found out that in practice, our algorithm generates numerous small moments that correspond to random accentuated words by the commentators, which didn't correspond to important moments in the game, hence the rule described.

The convolutional neural network unlike the HMM, uses the crowd cheering class as an indicator of Excitement. The structure differs a little bit. First of all, similarly to the previous pipeline, it only checks the last 3 seconds of a scene. Using the 3 last seconds, 3 conditions are tested (that have been determined empirically). We check if in the last three seconds, 2 or more have been classified as "Unexciting". If not, we then check if one or more have been classified as "Exciting" and that our model is confident about its prediction (meaning that the average predicted probability of the excited segments are higher than 80%). If this condition is met, we assume that this scene is intresting and is then saved for our highlight. If not, we then check if we then check if one or more seconds have been classified as "Crowd" and that the prediction score is higher than 90%. If so, we don't save the scene, but add it to the start of the next scene and re-evaluate then. This final condition allows us to generate more coherent segments by keeping the intermediate short scenes after a goal and a celebration (close up of a player, team celebration etc...) where the commentator doesn't speak. Moreover, the pipeline is changed when generating a highlight for a video game. Often, in games, crowd cheering and excited commentators go hand in hand. In so, we decided to remove the last condition, and change the second one to : If at least a Crowd OR a Excited second is detected, keep the scene. Finally, as stated above, segments are truncated using the same second by second information.

The combination algorithm 's task is to take the list of important moments found by the audio classification and the OCR and combine them together. The algorithm concatenates both lists into one and sorts it by the end of the segments. Moments found by the OCR or by the audio classification that are timely far from other moments are saved by default. Then, if the algorithm finds a segment from the audio classification that overlaps or is close to a segment from the OCR, it has to decide between the two. Usually, if the OCR found a replay longer than 40 seconds, this means it most probably is a goal. Thus, to save time in the highlights, we save the audio segment. Otherwise, if the OCR moment's length is smaller than that, we save it instead of the audio, so that we can see the replay of the action.

5 Experimental Results

In this section we aim to test every model and brick implemented and compare them in order to decide which model answers best our criteria.

Brick	Generation Time	Emotional Analysis
HMM	Real-time	True
CNN	Real-time	True
OCR	Match length	False

Table 2: Implemented bricks comparison

First and foremost it is important to understand that the OCR brick doesn't function in real time and is not based on emotions. Thus, it can be used as a stand-alone brick, or combined with the others. In order to test accurately our models, we decided to evaluate them on 5 different subjective standards : completeness (if every important action was found), length of the highlight, editing and cutting, junk (if many junk scenes are present), enjoyment. Each viewer/tester had to rate each criterion from a scale of 0-10. We tested each model and brick combination against each-other on 3 french "Ligue 1" matches, 2 international matches and finally on 1 FIFA video-game match (with french commentary).

Match	CNN	HMM	CNN + OCR	HMM + OCR	OCR	League
PSG - Nice	38	31	?	?	?	French Ligue 1
OM - PSG	33.25	28.75	?	13.5	34.5	French Ligue 1
Montpellier - PSG	38.75	38.5	?	?	30.25	French Ligue 1
Napoli - Benfica	21.25	21.5	NA	NA	NA	UEFA Champions League
Real Madrid - Atletico Madrid	15.5	21	NA	NA	NA	UEFA Champions League
Fifa	44.25	39.25	NA	NA	NA	None
Global Average	35.33	32.13	?	7.3	32.38	None

Table 3: Score over 50 per brick combination for each tested match

Note that NA means that the brick can not produce a highlight without adapting certain OCR parameters for a given match and that '?' means that no conclusive result has been made on this model caused by videos being too long (around 30min per 45min halves) or too short (20s to 1min per half).

Looking at the data, we can assess, thanks to the global average, that the CNN was the favorite model. The global average is a weighted average that favors matches in french language (weight of 1 for Ligue1, 0.5 for the FIFA match and finally 0.25 for international matches). When evaluating our bricks, we concluded that adding the OCR to our pre-trained models did not improve enough our models. In most cases it hindered the enjoyment of the highlight due to the extended length and the extra junk scenes that made it into the final video. Evaluating the OCR by itself, we can see that basing our model solely on the visual cue, we can extract interesting highlights (without the need of training models etc...). We believe it could outperform the two other models for international matches if adapted to specific leagues. On the other hand, the OCR doesn't work as well for FIFA games or international games. The scoreboards aren't always located at the same place and thus the OCR fails the "Ease of implementation" criterion. As stated earlier, the HMM and CNN models do not perform well on languages different from french. The extracted videos were often too long and missed key moments.

Looking at the HMM, the actual classification of scenes gives very impressive results. While examining the class statistics, we agreed with much of the classification done for the most important class - Excited commentary. One downfall of the audio classification was its misunderstanding between the crowd cheering and excited commentary. Since in the database we created, the commentators' screams were always accompanied by the crowd cheering, the HMM learned that it's a part of the class. Thus, when the crowds were cheering extremely loud, it was classified as an excited commentary, adding impurity to our classification. The most prevalent problem with this audio classification is the selection algorithm that is too permissive in the case of HMM. Changing the threshold of selection just by a small value changes drastically the outcome. Thus, because we wanted to make sure all important scenes were included in the highlights, our selection algorithm seems too permissive and makes our final video result too long.

Both the HMM and the CNN model work extremely well in generating highlights from video games. Overall, all three methods perform well and have high potential on different areas.

6 Conclusion

We are very pleased with our results. Because Hidden Markov Models are historically used for audio classification and emotional analysis, we expected it to have great results and thought it could be used as benchmark. We tried to better that by using newer state of the art models such as CNN. Overall, we think our results exceeded our expectations because we built a solution that is highly flexible, not only for football matches but also other forms of media, such as video games. We also believe that the selection algorithm plays a bigger role than we initially expected. The CNN brick uses a simpler but more restrictive approach to the scene picking in comparison to the HMM brick.

All in all, using the CNN implementation is more efficient, uses less memory and produces better results.

6.1 Discussion

We have decided that our system would only run on game halves to avoid having to handle half-time video streams. The user can then concatenate the highlights to get the entire reel for the game. We also considered that the transmission of the video stream is out of the scope of this paper since it is not part of the core of our research. We mostly trained our models on French-based matches in order to improve consistency. This might introduce a bias in some of our models.

6.2 Improvements

With our current resources processing a high-quality video stream was highly time and memory consuming, so we prioritised execution speed over quality. But this issue needs to be solved since, nowadays, the audience prefers having content with good image quality. Then, since the process was too heavy to run while keeping the data in the memory, we had to perform some intermediate saves. Finding a solution for this issue can significantly increase the performance of HiLite. For the selection of the moments, a good improvement would be doing a random search or cross validation for the meta parameters that influence the decision. While the audio classification is rather robust, the selection algorithm impacts greatly the final result. Indeed, finding the best thresholds for our selection algorithm would ensure it generates highlights as intended. It would also allow our solution to be more robust as a whole and not depend on subjective estimations. We also ran into some problems when using other matches than Ligue 1. This is due to a variety of factors such as other languages influencing audio analysis and different scoreboard color schemes and location. We would need to train our models on more varied data in order to make our solution more adapted to different competitions. Another great way of reducing the fluctuations in the results across matches from multiple leagues and countries is doing an on-the-fly sampling of classes like ambient noise and crowd cheering during the testing execution. This way, it would be possible to adapt certain thresholds to the match in question, making our solution robust to great variations in the audio levels. Moreover, today, there are some channels that don't remove the scoreboard during replays (such as la Liga during the season 2020-2021). While not predominant in the football industry, this could cause problems to our OCR model, since it is based on scoreboard appearance and disappearance. We would have to adapt our code to interpret score changes instead. On another note, a simple way of improving the solution's functionalities can be allowing the user to indicate a maximum duration for the highlights. Then we could rank the different sequences resulting from the analysis and filter them to only keep the most interesting ones.

In the end, HiLite allows communication teams from broadcasting groups or football clubs to free some time to do more interesting work. It could also help small competitions gain more visibility without having to invest big amounts of money in the necessary equipment or workforce.

References

1. Xu, Guangyou, Linmi Tao, and Guoying Jin. "SLOW-MOTION REPLAY DETECTION IN SOCCER VIDEOS BASED ON MULTI-LEVEL HMM INTEGRATED WITH SHOT DETECTION," n.d., 5.
2. Shukla, Pushkar, Hemant Sadana, Apaar Bansal, Deepak Verma, Carlos Elmadjian, Balasubramanian Raman, and Matthew Turk. "Automatic Cricket Highlight Generation Using Event-Driven and Excitement-Based Features," n.d., 9.
3. 1.Guo J, Gurrin C, Lao S, Foley C, Smeaton AF. Localization and Recognition of the Scoreboard in Sports Video Based on SIFT Point Matching. In: Lee K-T, Tsai W-H, Liao H-YM, Chen T, Hsieh J-W, Tseng C-C, éditeurs. Advances in Multimedia Modeling [Internet]. Berlin, Heidelberg: Springer Berlin Heidelberg; 2011 [cité 11 nov 2020]. p. 337-47. (Lecture Notes in Computer Science; vol. 6524). http://link.springer.com/10.1007/978-3-642-17829-0_32
4. Abdoli, Sajjad, Patrick Cardinal, and Alessandro Lameiras Koerich. "End-to-End Environmental Sound Classification Using a 1D Convolutional Neural Network." Expert Systems With Applications, 2019, 12.
5. M. Cavagna, A. Abdel Rahman, R. Cinciuc and B. Sader, "State of the Art - An Emotional Sports Highlight Generator", Lyon, November 2020, http://github.com/BrunoSader/An-emotional-sports-highlight-generator/blob/main/State_of_the_art___An_Emotional__.pdf.
6. 1.Kővári B. Football Monitor (football-score-monitor) [Internet]. 2018 [cité 10 nov 2020]. Disponible sur: https://github.com/bkovari/football-score-monitor

7 Appendix

Extended references can be found below.

References

1. Review of Speech-to-Text Recognition Technology for Enhancing Learning p. 21
2. Abdoli, S., Cardinal, P., Koerich, A.L.: End-to-end environmental sound classification using a 1D convolutional neural network. Expert Systems With Applications p. 12 (2019)
3. Baijal, A., Jaeyoun Cho, Woojung Lee, Byeong-Seob Ko: Sports highlights generation bas ed on acoustic events detection: A rugby case study. In: 2015 IEEE International Conference on Consumer Electronics (ICCE). pp. 20–23 (Jan 2015). https://doi.org/10.1109/ICCE.2015.7066303, iSSN: 2158-4001
4. Chung, J., Gulcehre, C., Cho, K., Bengio, Y.: Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. arXiv:1412.3555 [cs] (Dec 2014), http://arxiv.org/abs/1412.3555, arXiv: 1412.3555
5. Daoudi, K., Fohr, D., Antoine, C.: Dynamic Bayesian Networks for multi-band automatic speech recognition. Computer Speech and Language **17**(2-3), 263–285 (2003). https://doi.org/10.1016/S0885-2308(03)00011-1, https://hal.inria.fr/inria-00099530, publisher: Elsevier
6. Decroos, T., Dzyuba, V., Haaren, J.V., Davis, J.: Predicting Soccer Highlights from Spatio-temporal Match Event Streams p. 7
7. Deng, J., Guo, J., Zhou, Y., Yu, J., Kotsia, I., Zafeiriou, S.: RetinaFace: Single-stage Dense Face Localisation in the Wild. arXiv:1905.00641 [cs] (May 2019), http://arxiv.org/abs/1905.00641, arXiv: 1905.00641
8. Fohr, D., Mella, O., Antoine, C.: The automatic speech recognition engine ESPERE : experiments on telephone speech. In: ICSLP. p. 4 p. Pékin, China (2000), https://hal.inria.fr/inria-00099154
9. Giancola, S., Amine, M., Dghaily, T., Ghanem, B.: SoccerNet: A Scalable Dataset for Action Spotting in Soccer Videos. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW) pp. 1792–179210 (Jun 2018). https://doi.org/10.1109/CVPRW.2018.00223, http://arxiv.org/abs/1804.04527, arXiv: 1804.04527
10. Guo, J., Gurrin, C., Lao, S., Foley, C., Smeaton, A.F.: Localization and Recognition of the Scoreboard in Sports Video Based on SIFT Point Matching. In: Lee, K.T., Tsai, W.H., Liao, H.Y.M., Chen, T., Hsieh, J.W., Tseng, C.C. (eds.) Advances in Multimedia Modeling, vol. 6524, pp. 337–347. Springer Berlin Heidelberg, Berlin, Heidelberg (2011). https://doi.org/10.1007/978-3-642-17829-0$_3$2, http://link.springer.com/10.1007/978-3-642-17829-0_32, series Title: Lecture Notes in Computer Science
11. Islam, M.R., Paul, M., Antolovich, M., Kabir, A.: Sports Highlights Generation using Decomposed Audio Information. In: 2019 IEEE International Conference on Multimedia Expo Workshops (ICMEW). pp. 579–584 (Jul 2019). https://doi.org/10.1109/ICMEW.2019.00105
12. Jančovič, P., Köküer, M.: Incorporating the Voicing Information into HMM-based Automatic Speech Recognition in Noisy Environments. Speech Communication **51**(5), 438 (Mar 2009). https://doi.org/10.1016/j.specom.2009.01.003, https://hal.archives-ouvertes.fr/hal-00516741, publisher: Elsevier : North-Holland

13. Jia, S., Wang, S., Hu, C., Webster, P., Li, X.: Detection of Genuine and Posed Facial Expressions of Emotion: A Review. arXiv:2008.11353 [cs] (Aug 2020), http://arxiv.org/abs/2008.11353, arXiv: 2008.11353
14. Kuo, C.M., Lai, S.H., Sarkis, M., Diego, S.: A Compact Deep Learning Model for Robust Facial Expression Recognition p. 9
15. Kővári, B.: Football Monitor (football-score-monitor) (2018), https://github.com/bkovari/football-score-monitor
16. Li, S., Deng, W., Du, J.: Reliable Crowdsourcing and Deep Locality-Preserving Learning for Expression Recognition in the Wild. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2584–2593. IEEE, Honolulu, HI (Jul 2017). https://doi.org/10.1109/CVPR.2017.277, http://ieeexplore.ieee.org/document/8099760/
17. Louis Martin, Benjamin Muller, Pedro Javier Ortiz Suarez, Yoann Dupont, Laurent Romary, Eric Villemonte de la Clergerie, Djamé Seddah, Benoît Sagot: CamemBERT (2020), https://camembert-model.fr/publication/camembert/
18. Lowe, D.G.: Distinctive Image Features from Scale-Invariant Keypoints. International Journal of Computer Vision **60**(2), 91–110 (Nov 2004). https://doi.org/10.1023/B:VISI.0000029664.99615.94, http://link.springer.com/10.1023/B:VISI.0000029664.99615.94
19. Luo, H., Han, J.: Nonnegative Matrix Factorization Based Transfer Subspace Learning for Cross-Corpus Speech Emotion Recognition. IEEE/ACM Transactions on Audio, Speech, and Language Processing **28**, 2047–2060 (2020). https://doi.org/10.1109/TASLP.2020.3006331, conference Name: IEEE/ACM Transactions on Audio, Speech, and Language Processing
20. Merler, M., Joshi, D., Nguyen, Q.B., Hammer, S., Kent, J., Smith, J.R., Feris, R.S.: Automatic Curation of Golf Highlights using Multimodal Excitement Features. arXiv:1707.07075 [cs] (Jul 2017), http://arxiv.org/abs/1707.07075, arXiv: 1707.07075
21. Munasinghe, M.I.N.P.: Facial Expression Recognition Using Facial Landmarks and Random Forest Classifier. In: 2018 IEEE/ACIS 17th International Conference on Computer and Information Science (ICIS). pp. 423–427. IEEE, Singapore (Jun 2018). https://doi.org/10.1109/ICIS.2018.8466510, https://ieeexplore.ieee.org/document/8466510/
22. Mustaqeem, Kwon, S.: A CNN-Assisted Enhanced Audio Signal Processing for Speech Emotion Recognition. Sensors **20**(1), 183 (Jan 2020). https://doi.org/10.3390/s20010183, https://www.mdpi.com/1424-8220/20/1/183, number: 1 Publisher: Multidisciplinary Digital Publishing Institute
23. Nichols, J., Mahmud, J., Drews, C.: Summarizing sporting events using twitter. In: Proceedings of the 2012 ACM international conference on Intelligent User Interfaces - IUI '12. p. 189. ACM Press, Lisbon, Portugal (2012). https://doi.org/10.1145/2166966.2166999, http://dl.acm.org/citation.cfm?doid=2166966.2166999
24. Pantofaru, C., Essa, I., Bettadapura, V.: Leveraging Contextual Cues for Generating Basketball Highlights. In: Proceedings of the 2016 ACM on Multimedia Conference - MM '16. pp. 908–917. ACM Press, Amsterdam, The Netherlands (2016). https://doi.org/10.1145/2964284.2964286, http://dl.acm.org/citation.cfm?doid=2964284.2964286
25. Pappalardo, L., Cintia, P., Rossi, A., Massucco, E., Ferragina, P., Pedreschi, D., Giannotti, F.: A public data set of spatio-temporal match events in soccer competitions. Scientific Data **6**(1), 236 (Dec 2019). https://doi.org/10.1038/s41597-019-0247-7, http://www.nature.com/articles/s41597-019-0247-7

26. Radhakrishan, R., Ziyou Xiong, Divakaxan, A., Ishikawa, Y.: Generation of sports highlights using a combination of supervised & unsupervised learning in audio domain. In: Fourth International Conference on Information, Communications and Signal Processing, 2003 and the Fourth Pacific Rim Conference on Multimedia. Proceedings of the 2003 Joint. vol. 2, pp. 935–939. IEEE, Singapore (2003). https://doi.org/10.1109/ICICS.2003.1292595, http://ieeexplore.ieee.org/document/1292595/
27. Rehman, A., Saba, T.: Features extraction for soccer video semantic analysis: current achievements and remaining issues p. 11
28. Sainburg, T.: NoiseReduce (2019), https://github.com/timsainb/noisereduce
29. Shukla, P., Sadana, H., Bansal, A., Verma, D., Elmadjian, C., Raman, B., Turk, M.: Automatic Cricket Highlight Generation Using Event-Driven and Excitement-Based Features p. 9
30. Skirpan, M., Yeh, T.: Designing a Moral Compass for the Future of Computer Vision using Speculative Analysis p. 10
31. Staudemeyer, R.C., Morris, E.R.: Understanding LSTM – a tutorial into Long Short-Term Memory Recurrent Neural Networks. arXiv:1909.09586 [cs] (Sep 2019), http://arxiv.org/abs/1909.09586, arXiv: 1909.09586
32. Staudemeyer, R.C., Morris, E.R.: Understanding LSTM – a tutorial into Long Short-Term Memory Recurrent Neural Networks. arXiv:1909.09586 [cs] (Sep 2019), http://arxiv.org/abs/1909.09586, arXiv: 1909.09586
33. Sun, X., Yang, Q., Liu, S., Yuan, X.: Improving Low-Resource Speech Recognition Based on Improved NN-HMM Structures. IEEE Access **8**, 73005–73014 (2020). https://doi.org/10.1109/ACCESS.2020.2988365, conference Name: IEEE Access
34. Tarnowski, P., Kołodziej, M., Majkowski, A., Rak, R.J.: Emotion recognition using facial expressions. Procedia Computer Science **108**, 1175–1184 (2017). https://doi.org/10.1016/j.procs.2017.05.025, https://linkinghub.elsevier.com/retrieve/pii/S1877050917305264
35. Vanderplaetse, B., Dupont, S.: Improved Soccer Action Spotting using both Audio and Video Streams. arXiv:2011.04258 [cs] (Nov 2020), http://arxiv.org/abs/2011.04258, arXiv: 2011.04258
36. Xiong, Z., Radhakrishnan, R., Divakaran, A., Huang, T.: Audio events detection based highlights extraction from baseball, golf and soccer games in a unified framework. In: 2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP '03). vol. 5, pp. V–632 (Apr 2003). https://doi.org/10.1109/ICASSP.2003.1200049, iSSN: 1520-6149
37. Xu, G., Tao, L., Jin, G.: SLOW-MOTION REPLAY DETECTION IN SOCCER VIDEOS BASED ON MULTI-LEVEL HMM INTEGRATED WITH SHOT DETECTION p. 5
38. Zhang, A.: SpeechRecognition (2017), https://pypi.org/project/SpeechRecognition/
39. Zhang, B., Dou, W., Chen, L.: Combining Short and Long Term Audio Features for TV Sports Highlight Detection, vol. 3936 (Apr 2006). https://doi.org/10.1007/11735106$_4$4, pages: 475
40. Zhang, B., Dou, W., Chen, L.: Combining Short and Long Term Audio Features for TV Sports Highlight Detection. In: Lalmas, M., MacFarlane, A., Rüger, S., Tombros, A., Tsikrika, T., Yavlinsky, A. (eds.) Advances in Information Retrieval. pp. 472–475. Lecture Notes in Computer Science, Springer, Berlin, Heidelberg (2006). https://doi.org/10.1007/11735106$_4$4