Smart Summaries

AI-driven video summarization applied to soccer matches

By 2022, online videos will make up more than 82% of all consumer internet traffic – that's 15 times higher than it was in 2017 [Cisco]. Sports videos will account for a good portion of this and every broadcaster or media company wants to be the first to publish new, relevant content. Google reported that, in 2017, searches for football highlight videos increased by 90% on YouTube and the amount of time people spent watching sports highlight videos was up by more than 80% [Google report]. To top it all off, COVID-19 has increased the digital screen time in general.

To meet this skyrocketing demand, we built an AI software application that selects highlights from soccer matches and collates these into a highlight movie. This makes it easier for experts to summarize a match under enormous time pressure. Let's explain how it works.

To detect the highlights, we split the video of the match into an audio and a video stream. Both streams are cut up into fragments and for each fragment we predict whether or not it contains a highlight. Below, we describe the video and audio model, but beware – it will get technical. If you want to skip ahead to the demo, jump to the last paragraph.

The video model

The video model processes the video data and predicts, per video fragment, whether or not it contains a highlight. But there is more to it than that. The soccer videos that we used as the input for our application contain more than just soccer. Many of the videos are preceded by a short title sequence, some shots of the audience and the arena, and a commercial during half-time. This affects the mean audio value, without being actual game-related sounds. To filter these out, we trained a video model that distinguishes between parts that are and parts that are not related to the match.

Our video model uses a Tensorflow multimodal versatile network (MMV) to extract features from fragments. This network can be found here. This MMV was pretrained on a set of videos to learn the features that can be used in classifying the gestures and motions performed in fragments of those videos. We added a dropout layer of 20% and a densely connected layer with 512 nodes on top of the network to let it learn which features to use. We finished the network with a classification node that predicts if the input contains soccer or not. We labeled the training set by indicating the start of the match and the half-time start and end times. The final model requires an input consisting of 32 frames, which we spread out over multiple seconds by sampling the video at 3 FPS. This way we increase the amount of action in a set of frames without changing the input shape.

Now, knowing we can predict if a video fragment contains soccer, this model might also be able to predict if it contains a highlight. We trained a separate model for this using the same pretrained model as the base, but with a dropout layer of 40% and two dense layers of 256 and 128 nodes, respectively. We labeled where videos contain goals, since these fragments should always be included in a highlight reel. This time, we sampled the videos at 6 FPS.

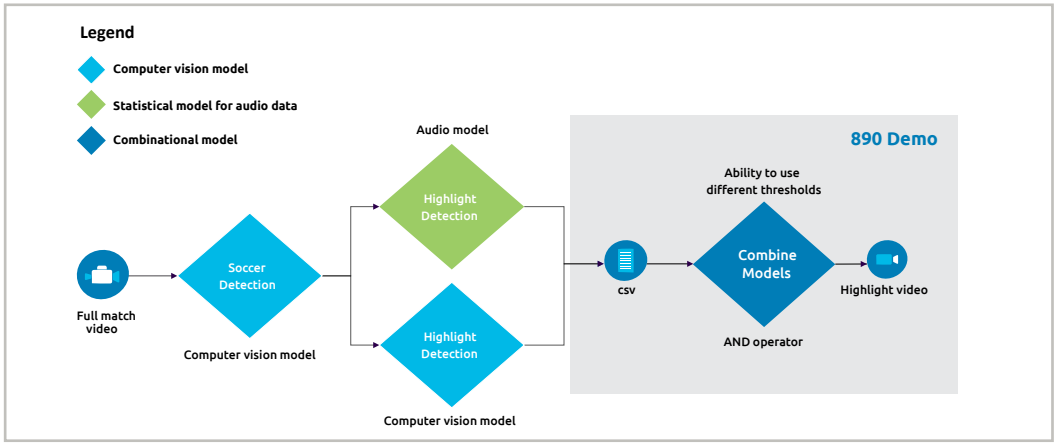
In the final program, the soccer detection model runs first to filter the video. The filtered video is then fed to the highlight detection model. The result is a probability between 0 and 1 for every second that that second contains a highlight or not.

The audio model

A prediction based on the audio data follows from a statistical model that determines if a certain fragment contains significantly loud cheering or not. Now, we will dive into the details of the model. For each second of audio data, we sum up the audio energies to obtain the integrated "loudness" for each second. As a next step, we get rid of peaks that are too narrow (outliers) while preserving wider peaks by applying a convolution with an array of 10 entries of 1/10. We do this because an actual highlight should be accompanied by cheering that lasts a number of seconds. For each of the resulting datapoints, we calculate a Z-score. These Z-scores measure how far away from the mean each datapoint lies. Transforming these into a p-value is easy using the standard normal survival function. The challenge is that a lot of p-values are very close to one. This would require us to use a threshold of about 0.997 when deciding whether or not a particular second contains a highlight. To rescale the p-values we use $p_{rescaled} = 1000^{p\_value}/1000$. To give you an idea of this transformation: an original p-value of 1 remains 1, but an original p-value of 0.9 becomes 0.5. Therefore, this transformation practically zooms in on the p-values close to 1.

Combining the results

For each model we assign an individual threshold. If a prediction is above this threshold, the model considers that particular second a highlight. The results are combined by selecting only those fragments that are considered highlights by both models.



We've published the result on the AI Gallery and the global I&D platform 890. The demo allows you to choose between a number of matches, for example the 2018 World Cup final between Croatia and France. We generate a summary based on the video and it is available for you to watch. In another tab, we invite you to dive into the results of the individual models. Feel free to play around with the parameters to see how this affects the resulting summary!

Authors

Capgemini Nederland B.V. Insights & Data

Jorrit Bootsma
Team Lead
jorrit.bootsma@capgemini.com

Pearl Stam
Senior Data Science Consultant
pearl.stam@capgemini.com

Sjaak ten Caat
Computer Vision Specialist
sjaak.ten.caat@capgemini.com

Vincent van der Meij
Data Science Consultant
vincent.vander.meij@capgemini.com