**Exit Test Part -2**

# **HIGHLIGHT EXTRACTOR**

The first step for highlight extractor is action spotting. We will be given a long untrimmed video ,and from there we need to extract the short video of 1 minute duration with the action of interest. Action spotting refers to determining the stamp,at which the required action is available.

# **ACTION SPOTTING**

The first step would be to convert the given video to a low quality video for fast processing. Then convert the video into timeframes so that the further processing can be done on images. Timeframes will preserve the temporal data which is crucial for object tracking.

The action spotting can be split into 3 subtask.

* Object detection – Ball and the player
* Object tracking – Tracking the ball and players.
* Action Prediction – In this case action is goal. So we need to set the criterion for exactly when it is a goal to avoid multiple predictions for the same goal event. The timestamp at which the ball crosses the line is considered goal.

# **SoccerNet**

The given task is a complicated task, if done from scratch, which requires time, huge memory capacity and computing power. The approach taken is direct inference using existing model and also transfer learning.

SoccerNet Action Spotting is part of the SoccerNet-v2 dataset, which  is composed of 550 complete broadcast soccer games and 12 single camera games taken from the major European leagues. SoccerNet is not only dataset, but also yearly challenges where the best teams compete at the international level. Action spotting is one of the subtask in the event. It provides benchmark to start with the challenge. CALF benchmark has better results for Goal prediction.

# **Goal Spotting**

As a first step we cloned the repository and also downloaded a few test videos to do inference using CALF benchmark model. Inference/main.py was used to output the actions in the video. A json file was the output. Then we have to extract the Goal , its timestamp and confidence from the predictions.

The function (final\_video) will take a json file as input, extract the event with action goal, add buffer and outputs the start and end time for each goal highlight video. A confidence threshold of 25% was used for goal. A 30 second buffer was added to the extracted time stamp, so that the final highlight video is of 1 minute duration.

# **Highlight Extract**

Inorder to extract the 1 minute video using timestamp, a python video processing library moviepy was used. The ffmpeg\_extract\_subclip module can extract and save the video to the given path.

# **Happy moment/Loss moment**

This involves emotion recognition .This will work better with high quality videos. Two approach was tried to accomplish this task. One is using FER library and the other is using YOLO.

Using Fer was slow due to sequential processing. First the highlight video is converted to frames and for each frame face is detected and emotion is predicted.

Yolo approach would be faster . Yolov5 can be used to predict two classes, happy or loss. For happy moments , consider face emotions happy and surprised. For loss moment, face emotions -sad, angry, disappoint etc can be considerd. Annotated data is needed for the two classes and then train the YOLOv5. Use trained model to detect the 2 classes in given video. We will get bounding box with tag happy or loss for each detected face of players.