**Highlights extractor**

Video analysis is extensively being used for several applications in different domains like sports, entertainment, healthcare, etc. For instance, it helps to perform in-depth analysis of scenarios and human behaviors to learn from or plan strategies in the case of sports.

The system can output three types of highlights:

* Happy moments
* goals
* Loss moments.

**Steps Followed for the final Output:**

****

**Data Gathering and Dataset Creation**

Various Images and Videos are gathered for Creation of Dataset which is uploaded into drive as ExitPart2

A sub folder was created called Images which contain training Images and test images was stored in test subfolder

**Research On Implementation of Model**

**1.Emotion Detection**

Emotion detection refers to the process of recognizing and classifying emotional states, such as happiness, anger, sadness, or fear, based on various cues such as facial expressions, voice tone, or text.

There are several approaches to emotion detection, including:

1. Facial expression recognition: This approach uses computer vision techniques to analyze and interpret facial expressions to determine the emotional state of a person.
2. Body language analysis: This approach uses computer vision techniques to analyze body movements and gestures to determine a person's emotional state.

There are many open-source libraries and tools available for emotion detection, including OpenCV, dlib, and TensorFlow. To implement emotion detection, you typically need to train a machine learning model on a large dataset of labeled examples of emotional expressions. The model can then be used to classify new inputs into the emotions that it has been trained on.

**2.Object tracking**

Object tracking is a technique in computer vision that involves locating an object in consecutive frames of a video stream. The goal of object tracking is to keep track of an object's position as it moves in the video over time.

There are several approaches to object tracking, including:

1. Color-based tracking: This approach uses the color information of the object to track it. The idea is to define a color histogram for the object, and then use that histogram to search for the object in each frame.
2. Feature-based tracking: This approach uses features of the object, such as corners, edges, or blobs, to track it. The idea is to detect and track these features in each frame to determine the object's position.
3. Deep learning-based tracking: This approach uses deep learning algorithms, such as Convolutional Neural Networks (CNNs), to track objects. These algorithms can learn to recognize objects in a video and track them automatically.

Object tracking is used in many applications, such as video surveillance, robotics, autonomous vehicles, and sports analysis.

OpenCV, a popular open-source computer vision library, provides several functions for object tracking, including the CAMShift and MeanShift algorithms for color-based tracking, the GoodFeaturesToTrack function for feature-based tracking, and the dnn module for deep learning-based tracking.

**How Model Implemented**

Model was implemented using Opencv deep learning techniques and python The vocal inflections and facial expressions in the visual data, along with the language appearing in a textual transcript, provide important cues to better identify affective states of opinion holders, creating a more robust emotion recognition model. Though the primary focus of the experiment I will summarize below was to classify sentiment in videos (positive, neutral, or negative), their findings can be incorporated into the more specific task of emotion classification across a wider spectrum of basic emotional categories (anger, happiness, sadness, neutral, excitement, frustration, fear, surprise, and other).

**MobileNet**, a predefined model was used for Emotion detection. MobileNet is a streamlined architecture that uses depth wise separable convolutions to construct lightweight deep convolutional neural networks and provides an efficient model for mobile and embedded vision applications.

**Keras ImageDataGenerator** is used for getting the input of the original data and further, it makes the transformation of this data on a random basis and gives the output resultant containing only the data that is newly transformed. It does not add the data. Keras image data generator class is also used to carry out data augmentation where we aim to gain the overall increment in the generalization of the model. Operations such as rotations, translations, shearin, scale changes, and horizontal flips are carried out randomly in data augmentation using an image data generator.

### **EarlyStoppingclass** Stop training when a monitored metric has stopped improving. Assuming the goal of a training is to minimize the loss.

**ModelCheckpoint** Callback to save the Keras model or model weights at some frequency.ModelCheckpoint callback is used in conjunction with training using model.fit() to save a model or weights (in a checkpoint file) at some interval, so the model or weights can be loaded later to continue the training from the state saved.

.

**Deployment Of App**

Streamlit is an open-source Python library for building and deploying interactive web applications for data science and machine learning. It provides a simple, intuitive interface for building modern, dynamic web applications in just a few lines of code.

the app uses OpenCV to extract goal, happy, and loss moments from a video. The app starts by uploading a video file using the **file\_uploader** function from Streamlit. Once the file is uploaded, it is read using the **cv2.VideoCapture** function from OpenCV. The app then identifies the goal moments and extracts 30-second segments before and after each goal moment. Finally, the app displays the extracted segments using the **st.image** function from Streamlit.