Analysis of Basketball Dribble : No. of dribbles and Frequency

Methodology:

Background Subtraction for Motion Detection:

Utilized OpenCV's createBackgroundSubtractorMOG2() function to detect moving ball by subtracting the current frame from the background model.

Circle Detection using HoughCircles:

I have used the Hough Circle Transform via cv2.HoughCircles() function to detect circular objects in the video frames, with some parameters to detect the basketball. It may detect some other circular objects as well, which can be removed by changing some of the parameters.

Counting the Dribbles:

This code Detects the downward movement of the basketball to count each dribble. A dribble was counted whenever the ball moved downward in subsequent frames.

Dribble Frequency Analysis:

Calculated the dribble frequency as the number of dribbles per second based on the frame count and the frame rate of the video, and displayed on the screen.

Challenges Faced:

Accurate Ball Detection:

Ensuring accurate detection of the basketball amidst various circular objects in the scene, requiring fine-tuning of parameters for Hough Circle Transform and background subtraction.

False Positives:

In this program, we may have to deal with false positives, such as detecting circular objects on the player's clothing or background, by implementing filters based on position and size of the detected circles.

Code Documentation:

The provided code snippet consists of a function analyze_basketball_dribble() designed to analyze a basketball dribble video and extract two key metrics: the number of dribbles performed by the player and the dribble frequency (dribbles per second).

Function Documentation:

```
def analyze_basketball_dribble(video_path):
    """
    Analyzes a basketball dribble video to count the number of
dribbles performed by the player
```

```
    and calculates the dribble frequency (dribbles per second).

    Arguments:
    - video_path (str): Path to the input video file.

    Returns / output:
    - dribble_count (int): Number of dribbles performed by the
player.
    - dribble_frequency (float): Dribble frequency (dribbles
per second).
    """
```

This function takes a single argument video_path. It returns two values: dribble_count which represents the number of dribbles performed by the player, and dribble_frequency, representing the dribble frequency in dribbles per second.

Implementation:

The implementation code within the function is omitted, as indicated by the comment # Implementation code goes here.... This is where the actual analysis of the basketball dribble video takes place, including motion detection, circle detection, and counting of dribbles.

Usage Example:

```
# Input video
video = 'basketball.mp4'

# Analyze the basketball dribble video
dribble_count, dribble_frequency =
analyze_basketball_dribble(video)

# Print the results
print("Number of dribbles performed: ", dribble_count)
print("Dribble frequency: ", dribble_frequency)
```

This section demonstrates how to use the analyze_basketball_dribble() function by providing the path to the input video file. It then prints the number of dribbles performed by the player and the dribble frequency obtained from the analysis.

Conclusion:

The provided code snippet offers a simple and concise method for analyzing basketball dribble videos, enabling users to extract meaningful insights regarding the player's dribbling performance. In the real time, we can add other features as well, such as players pose or movement detection or recognition, using Mediapipe Library, speed, and other things as well. This video was confined to very few metrics, so we calculated only to Metrics i.e. The number of dribbles , and it’s frequency.