**Basketball Dribbling Analysis Documentation**

**- Pratik Dwivedi**

**1. Introduction**

The program utilizes computer vision techniques to extract meaningful insights from a video of a basketball dribble.

Evaluation Metrics are as follows:

1. **Bounces**: The number of times the ball bounces on the ground
2. **Speed**: The speed of the dribble (Fast, Medium, Slow)
3. **Bounce type**: The type of bounce (High, Medium, Low)
4. **Direction**: The direction of the dribble (Across, Up & Down)

**2. Analysis Methods and Algorithms**

The program utilizes **YOLO (You Only Look Once)** object detection model for detecting the basketball in each frame of the video. Exact model used is the YOLOv8 nano [1] which is pretrained using a custom **RoboFlow** dataset [2] from their repository.

**The Kalman filter algorithm** is employed for **predicting the trajectory of the basketball** based on its detected positions, which is the crucial part of the calculating our metrics.

Various metrics such as the number of bounces, dribble speed, bounce type, and dribble direction are **measured using the detected and predicted basketball positions**.

**3. Implementation Details**

The program is implemented using **Python** programming language and **Streamlit** library for building the **user interface [4]**.

**OpenCV** library is used for video capture, frame processing, and visualization.

**YOLO object detection model** is loaded using the Ultralytics library with **custom weights** of a model trained to detect only a basketball in the image.

**KalmanFilter class** is implemented to predict the trajectory of the basketball based on its detected positions (details of classes and functions is documented in the code [3] itself).

**Analyzer class** is implemented to analyze the video file or webcam feed, detect basketball positions, and measure the defined metrics.

**4. Challenges Faced**

Integration of YOLO object detection model with Streamlit application as it required a different method for visualization and frame processing according to Streamlit.

Ensuring real-time performance and accuracy of the analysis algorithm in accordance to the metrics that require precise data to perform calculations.

Implementing a robust method for predicting the trajectory of the basketball which required maintaining a fixed length queue of the ball’s position so that it can be used to get a better prediction of its trajectory.

**5. Analysis Tasks and Results**

The program successfully detects the basketball in each frame of the video.

It accurately measures the number of bounces, dribble speed, bounce type, and dribble direction based on the detected basketball positions.

The metrics are overlaid on the frames of the video for visualization and analysis.

**6. Custom Analysis Tasks**

The example metric given in the assignment was to count the number of bounces of the ball in the video, I implemented some other metrics that we can calculate from the video such as:

1. Speed of the ball which is essentially the time taken between each bounce of the ball.
2. Dribble type – if a player is switching hands to dribble or using just one hand.
3. How high the bounce of the ball is according to the time taken between bounces.

Although these metrics may not be robust enough to work in all scenarios of basketball dribbling videos, but it provides a baseline of what can be achieved and how to achieve it.

**7. Conclusion**

The Basketball Dribbling Analysis program effectively utilizes computer vision techniques to extract meaningful insights from the basketball dribble video.

It provides a user-friendly interface for analyzing the video and visualizing the measured metrics.

Further improvements can be made to enhance the accuracy and robustness of the analysis algorithm.

1. a better fine-tuned YOLO model with a bigger basketball dataset to accurately detect and localize the ball in the frame.
2. Switching to a Object Segmentation model like the YOLOv8-seg which can help in several ways such as:
   1. making the analysis more robust since the segmented part of the basketball can give us the area of the ball that can be used to calculate the distance of the ball from the camera and adjust the metric thresholds dynamically.
   2. Provide proper coordinates of the ball for accurately predicting the trajectory of the ball further enhancing the robustness of the analysis algorithm.

**8. References**

1. [YOLOv8](https://github.com/ultralytics/ultralytics)
2. [RoboFlow Dataset](https://universe.roboflow.com/vaishnavi-jade/basketball-ec2xx/dataset/1)
3. [GitHub Repository](https://github.com/Dekode1859/Basketball_YOLO)
4. [Deployed Webapp](https://bbyolo.streamlit.app)
5. [Kalman Filter Implementation](https://www.youtube.com/watch?v=3iqRhbXBVRE)