

Here's a structured, point-wise explanation of the implemented approach:

1. Image Dataset Preparation

- **YOLO Model for Object Detection:** A pre-trained YOLOv8 segmentation model (yolov8s-seg.pt) is used to detect and segment players from an image (output.jpg).
- **Bounding Box Extraction:** The model detects the bounding boxes for each player in the image, and each detected player is cropped and saved as a separate image in the 'reference_images' folder for further processing.

2. Feature Extraction Using Pre-Trained Model

- **Mobilenet-v2 for Feature Vectors:** The feature vectors of all images are extracted using the pre-trained 'mobilenet-v2-tensorflow2-tf2' model, which is loaded via TensorFlow Hub.
- **Image Preprocessing:**
 - Images are resized to a fixed size of 224x224 pixels and converted to a float32 data type.
 - The processed images are passed through the mobilenet-v2 model to obtain feature vectors.
- **TensorVector Class:** A class is defined to encapsulate the feature extraction process for an individual image.

3. Reference Image Selection

- **Reference Images:** Four reference images (player0.png, player1.png, player2.png, player3.png) are used to represent different players.
- **Vector Extraction:** The feature vectors for these reference images are extracted and stored in a dictionary for future comparison.

4. Image Grouping Based on Folders

- **Target Folders:**
 - Two folders, 'two_players_bot' and 'two_players_top', contain images to be compared against the reference images.

5. Vector Collection for Clustering

- **Image Vector Extraction:** For each image in the 'two_players_bot' and 'two_players_top' folders, feature vectors are extracted using the same mobilenet-v2 model.
- **Collecting All Vectors:** All image feature vectors (including those from the reference images) are collected for clustering.

6. K-Means Clustering

- **Number of Clusters:** K-Means clustering is applied to the extracted feature vectors with the number of clusters set to 4, representing the four players.

- **Cluster Assignment:** Each image (both reference and target) is assigned a cluster label based on its similarity to the reference feature vectors.

7. Image Segregation into Classes

- **Output Directory Structure:**
 - An output folder (output) is created.
 - Inside this folder, four subfolders (player0, player1, player2, and player3) are created, each corresponding to one cluster.
- **Image Placement:**
 - Based on the cluster labels assigned by K-Means, images from the 'two_players_bot' and 'two_players_top' folders are moved to the corresponding subfolders in the output folder.
 - Reference images are not moved but serve as the basis for cluster assignments.