Report on Code Update for Player Classification

Have implemented a significant change in the process of determining the best color space for player tracking. The new method involves using histogram features of color spaces rather than the mean color values.

Previous Approach:
Initially, the classification of players based on their shirt colors was determined using the mean values of colors within a selected color space (LAB, HSV, RGB). K-means clustering was applied directly on these mean color values to identify distinct clusters corresponding to different players.

Possible Issues with Previous Approach:
The previous method, while straightforward, had several limitations:
1. Sensitivity to Color Variations: Mean color values are highly sensitive to slight variations in lighting, shading, and camera angles, which can significantly affect the robustness of player tracking.
2. Inadequacy for Complex Scenes: In complex scenes with multiple overlapping players or rapid movements, mean color values can blend and lose the distinctiveness necessary for accurate player differentiation.

Updated Approach:
The approach involves the following steps:
1. Histogram Computation: For each player region identified, histograms are computed for three color spaces (LAB, HSV, RGB). Each histogram is quantized into 8 bins, providing a comprehensive profile of color distribution within the region.
2. Feature Concatenation: The histograms from the three color spaces are concatenated to form a single feature vector with 512 features per player region.
3. Standardization and Clustering: These concatenated histogram feature vectors are then standardized and used as inputs for K-means clustering.

Adjustments in Clustering Process:
Have also revised the method to determine the number of frames where players belong to the same cluster:
1. Euclidean Distance Measurement: Instead of checking for exact matches between histograms, we now use Euclidean distance to measure the closeness of feature vectors to their respective cluster centroids.
2. Dynamic Thresholding: A threshold based on the standard deviation of the dataset is used to decide if a histogram feature vector is close enough to a centroid. This method accommodates natural variations in histograms due to changes in environmental conditions and player movements.
3. Consistency Check: Only when two players' histograms are within their threshold distance to the same centroid are they considered to be in the same cluster for that frame.

Modified functions :

1. Player_tracker_utils.py : - colour_matching_algorithm
2. C_space_utils.py : get_all_players_code_values
3. Tracker_processor.py - initialise_teams
4. Team_utils.py - `assign_color_values_for_initialization`