Here is how I would like to update my code to implement piece tracking into my current logic as of 3/15/2026

1. **If this is the first run (gridlines are not yet stored), then we can skip this whole piece of logic,** because the pixel counting inherently relies on having the exact intersecting coordinates from the initial setup to isolate and analyze the individual bounding boxes for each square.
2. **Get a pixel count for each square on the 8x8 warped board between each of the stored gridlines,** calculating the active pixel density to establish a numerical weight that represents how much physical material is currently sitting inside that specific region.
3. **See if we already have a pixel count stored in the JSON, and if not, store the baseline count and detect which side of the board is white to locate A1.** Since the camera is set up side-on, this one-time step evaluates the pixel densities on the left and right sides of the frame to reliably map the visual grid to standard chess coordinates.
4. **If we do already have one stored, we compare the stored pixel values to the new pixel values** to calculate the delta—or the exact difference in pixel density—for all 64 squares between the previous state and the current frame.
5. **Check to see if any square gained a significant amount of pixels (meaning a piece moved to that square),** utilizing a predefined threshold to filter out visual noise like shadows or camera vibrations so you only capture intentional piece placements.
6. **Find pieces based off the stored FEN that could have moved to the square, and see if any of these squares where that piece was has fewer pixels than it did on the previous run.** This smartly uses the chess engine's rules to narrow your search, preventing false positives (like a hand moving across the board) by only checking the origin squares of legally viable candidate pieces.
7. **If this move was valid, store the move in a move_list on the JSON file (new variable),** creating a persistent, chronological history of the game that your app can easily read and display.
8. **Update the FEN with the new board state and update the JSON with the new pixel count,** effectively resetting your baseline so the system is fully prepped to detect the next move.