Ball-Tracking Project Report

Objective: To track the Ball (Blue Reactangle) and Bat (Green Rectangle) in a cricket match, with the help of openCV and Machine Learning.

Folder Structure:

cricket ball tracking ==> processed_videos ==> 1.mp4
==> 2.mp4
==> 3.mp4
==> 4.mp4
==> csv_files ==> ball_video1.txt
==> ball_video2.txt
==> ball_video3.txt
==> ball_video4.txt
==> bat_video1.txt
==> bat_video2.txt
==> bat_video3.txt
==> bat_video4.txt
==> Ball_tracking.ipynb
==> finalized_model.sav
==> ball-tracking project report.pdf

Prcedure: To track the ball and bat this is the procedure that I followed.

1. Finding appropriate contours:
    1. Read the video.
    2. Break each video into frames and save them in frames directory.
    3. Crop each frame, so that only important contours will be detected, which is in the middle of the screen.
    4. Convert each frame to numpy array.
    5. Now claculate the threshhold difference between each frames(It will be required later).
    6. Convert each frame to gray-scale.
    7. Blur each frame.
    8. Apply threshold masking, for this I applied Binary threshold. In the pitch frame(pitch is where batsman bats and bowler bowls) the minimum threshold value is 200, and when the ball is in the air or in the field it's value is 140.
    9. For this I have used threshold difference between each frame. And if the difference is more than given threshold then changing the value..
    10. From the the mask find the contours.

2. Training the Model:
    1. For training a model we need data. So for that we have 3 sub directories inside “images” directory – ball, bat and miscellaneous.
    2. First we are listing all the directories from images directory.
    3. Creating two lists, One is features and other one is labels.
    4. Reading each images from each sub directory
    5. Converting each images to gray-scale.
    6. Resizing them to 70 X 70.
    7. Appending each image to images list.
    8. Appending their folder names into labels list.
    9. Reshaping the image according to its length.
    10. converting the labels into categorical data.
    11. Split the data set for training and validation.
    12. Created a Random forrest model with depth = 30 and criterion = ‘entropy’.
    13. Fitting the model in training data.
    14. Getting classification report with validation data.

3. Predicting the object in a frame:
    1. Create data frames ball_df and bat_df.
    2. For each contour find the x, y coordinate and width and height of that contour.
    3. Calculate the ratio between width and height (take the maximum value between those and make it the numerator).
    4. Calculate xmin, xmax, ymin and ymax.
    5. If the ratio is greater than 0.2 , width and height is less than 100 then insert that contour’s frame, Frame Number, Frame Height, Frame Width, xmin, xmax, ymin and ymax in a dataframe .
    6. Make a directory called patch and save each contour based on xmin, xmax, ymin and ymax in that directory.
    7. List all the files from “patch” directory.
    8. Sort the files.
    9. Resizing them to 70 X 70
    10. Appending each image to test list.
    11. Predict each image with previously trained model.
    12. If bat or ball is predicted then include them in bat_df or ball_df dataframe.
    13. Remove the patch file.
    14. Repeat this process for each frame.

4. Creating tracked video:
    1. Removing the rows if NaN values exist in the ball_df and bat_df.
    2. Resetting indexes so that there is no skip over in the indexes.

3. Taking the frame values from ball_df and bat_df.
4. Drawing the contours in each frame based on the xmin, xmax, ymin and ymax in the ball_df and bat_df .
5. Listing all the frames and combining them to create a video.
6. Drop 'index' and 'frame' columns from the dataframe.
7. Save ball_df and bat_df as csv file.
8. Save the model.

What I have tried and failed:

a. First I created my own CNN But beacause of very less amount of data it was performing poorly.
b. Then I used Transfer Learning(MobileNetV2) again it was giving poor result.
c. Then I have tried Voting classifier to combine multiple models but that gave me worse result than my current model which is only Random forrest. Although it performed better than deep learning models .
d. Preprocessing wise, with minimum threshold value 200 it was marking ball as a contour only in the pitch but in the ground or when it is in the air it was not happening. So when scene is changing I am changing this value to 140.
e. I have tried adaptive threshold, which was taking a lot of time even after that the contours were not perfect.
f. For blurring, Although blurring the image with high value will pace up the process cause there will be less contour, but then bat was not getting detected and the high value is not effective when the ball is not on the picth.
g. I have tried not cropping each frame but then the process becomes a lot slower cause there were a lot more contours and most of them were uneccessary cause generally ball and bat stays in the middle of the frame.
h. When the ball is the picth or in the field it is working properly but when it is in the air then detecting it becomes difficult cause there are so many background objects. Though with more data and proper algorithm it will become easier.
i. There are so many objects in the background which looks like ball, so differentiating them will be a problem.

Future Possibilities: With more data we can create more precise model, then analyse a cricket match to get different poinnt of view. Proper angle between bat and ball, what need to be improved in a game or in player etc can be detected. The future possiblilties of this project is enormous.

Article that I followed: https://www.analyticsvidhya.com/blog/2020/03/ball-tracking-cricket-computer-vision/