CSC2541: Sports Analytics (Challenge 1)

February 25, 2017

The purpose of this project is to investigate video sports analytics. In particular, you will be experimenting with detection and tracking techniques. Additional bonus can be achieved if you do action recognition as well. Towards this goal we will be utilizing a subset of the dataset collected in [1], which contains actions clips of college basketball.

1 Dataset

Video clips corresponding to 11 basketball actions (e.g., shoot, pass) are annotated across 257 NCAA games. In total there are 14548 clips annotated. Each clip corresponds to 20 frames of a 4 second segment of the game. Moreover, each frame is annotated with bounding boxes and tracking data for all the players. For a more comprehensive description of the original dataset, please refer to section 3 of [1].

For this challenge, you are provided with a subset of the data in [1]. In particular, for each of the 11 actions, we have randomly chosen about 50 clips consisting of at most 20 consecutive frames. You can download the data by clicking here (2.6 Gb).

In order to use the dataset, you first need to extract the data using http://www.7-zip.org/. The data is structured in the following way:

- The games are in `data/`
- The clips for each game are in `data/game-id/`
- The extracted frames for each clip are in `/data/game-id/clip-id/`
- For each frame `frame-id.png`, you are also provided with `frame-id-info.csv` file which contains a table of bounding boxes and player tracking ids. At each row, you are given a tuple of (x, y, w, h) where x and y corresponds to the bottom left corner of the bounding box and w and h are its width and height.
- Note that the `frame-ids` are in an increasing order corresponding to the position of the frame in the clip. This is important for the tracking data.
- In `data/ directory`, you will find a python dictionary object where the keys correspond to the basketball actions and the values specify the associated clips. To load the dictionary, you can use the following command:

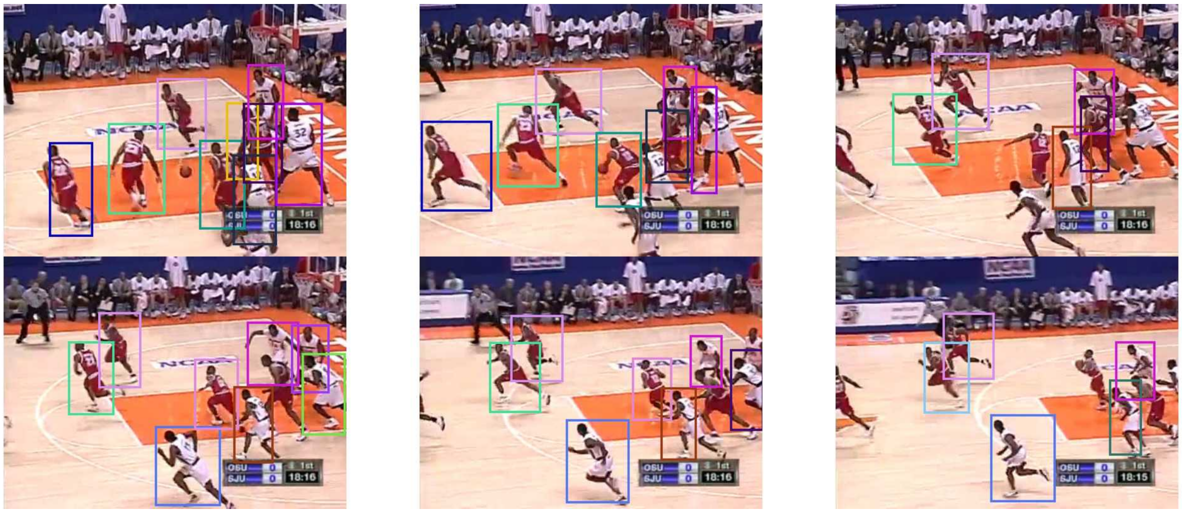
```
import pickle
events = pickle.load(open('CHANGE_THIS/data/events.pkl', "rb"))
```

- You can load and visualize the data using the following code:

```
import pandas as pd
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
import matplotlib.patches as patches

# specify the path to the root data directory here
data_dir = 'CHANGE_HERE/data/'
# specify the game name
game_name = '0En5pOUZN5M'
# specify the clip name
clip_name = 'clip_46'
# specify the image number
img_num = 1
im_path = data_dir+game_name+'/'+clip_name+'/'+'{:02}.png'.format(img_num)
bb_path = data_dir+game_name+'/'+clip_name+'/'+
                       '{:02}_info.csv'.format(img_num)
bbs_data = pd.read_csv(bb_path)
im = mpimg.imread(im_path)
fig = plt.figure(frameon=False)
ax = plt.Axes(fig, [0., 0., 1., 1.])
ax.set_axis_off()
fig.add_axes(ax)
ax.imshow(im)
# iterating over bounding boxes
for _, bbox in bbs_data.iterrows():
        print(bbox)
        rect = patches.Rectangle((bbox.x, bbox.y),
                   bbox.w, bbox.h, linewidth=3,
        facecolor='none')
        ax.add_patch(rect)
```

- Here are some visualizations of the data:



2 Challenge

For this project, you are expected to implement a player detection and tracking framework. You can come up with a new method or use any of the methods/codes covered in class. You can use any code available on the web so long as you properly acknowledge it. The limit is your imagination. For an added bonus, you can try to also do action classification.

Important Note: If the amount of data is too cumbersome for your computational resources, you can reduce the size of your training set at NO penalty.

3 Submission

A report (paper format) should be submitted explaining what you have done and relating it to the state-of-the-art. A presentation will be given in class as well.

References

[1] Vignesh Ramanathan, Jonathan Huang, Sami Abu-El-Haija, Alexander Gorban, Kevin Murphy, and Li Fei-Fei. Detecting events and key actors in multi-person videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3043–3053, 2016.