Understand baseball players tracking system

Shuya Zhao
New York University
Brooklyn, NY
sz2257@nyu.edu

Abstract

Baseball games are challenging and complicated datasets that are frequently used in the field of object detection, pose estimation, and video processing. In this paper, we start with basic image processing models, like object detection models, to explore the temporal changes in baseball players in baseball games. Also, we evaluate the performance of image processing models on short clips of baseball activities. The analysis and evaluation are based on visual analytics on the four main targets.

1. Introduction

Nowadays, researchers are not satisfied to only analyze images without additional temporal information, video understanding becomes the next step for Artificial Intelligence. Video processing models have developed powerful abilities on multiple tasks, such as activity classification [14], and video alignments [4]. This kind of model takes a sequence of frames into accounts, describing motions that give more much more information than that captured in single frames.

However, new techniques in video processing raise higher requirements for training datasets. For instance, clips of videos with uniform length and formats need to be prepared for training models on action video understanding (moments in time dataset). When we face a new situation, like designing a baseball player tracking system in real-time games, there are not datasets perfectly meeting the requirements. Moreover, the models cannot always be pre-trained and fine-tuned in advance. Therefore, we need to utilize more basic and adaptable models, like object detection and pose estimation.

Object tracking techniques have been applied to the player tracking system to analyze sports games, like baseball. In our tracking system, legotracker, we use two groups of cameras, which are panoramic cameras and narrow cameras. Different kinds of cameras capture characteristics of a single player and overview playground in the baseball game. When the object detection model is applied to frames taken from a fixed angle, we can analyze the players in the scores and areas of the object detection bounding boxes over time.

We focus the temporal results of four main targets: pitchers, batters, catchers, and umpires. Their actions interact with each other to form different activities in baseball games. In this paper, we explore their performance in ''swing'' which includes two main actions: the pitcher throws the ball and the batter swings to hit the ball. Since the short clips have the same length, we can label results in temporal order and do further analysis.

2. Related Work

Object detection models currently are required to deal with multiple target tasks. One popular type of model is built on the region proposal generated by some methods like selective search [7] and output results through a two-shot procedure, one to output region proposal, one to detect the class of each proposal. As one of the canonical model among this type, Faster R-CNN [15] introduces the regional proposal network (RPN) to replace the slow selective search in Fast R-CNN [6]. Other models propose similar techniques to the motivation that achieving maximum speed by sharing CNN among the whole picture. For example, the Region-based Fully Convolutional Net (R-FCN) uses position-sensitive score maps that could recognize certain parts of the object [3]. What's more, a single-shot detector (SSD) [11] gains huge speed over Faster R-CNN by directly generation multiple bounding boxes with C-class scores. (C is the number of object classes). It combines the prediction of class and scores into one step and skips the process of region proposal. Our research use SSD-Inception-V2 model which is implemented in the legotracker tracking system.

Depth estimation models aim to restore 2D images to 3D scenes. This approach is highly common in the application to problems like the rendering of 3D scenes, self-driving systems. The architectures, such as ResNet [10] and encoder-decoder [19] structure, which are frequently implemented in other applications like image segmentation and

classification are also used in the depth estimation models. Many current works employ unsupervised learning [8] to predict depths on videos, avoiding the limitation of datasets.

Explainable Machine Learning gives intuitions into Machine Learning models, the black box, via some methods like visualization and interaction tools. Visual analytics for model understanding [18, 12] and training [17] focus on the interpretability of the parameters, hidden features, outputs and inner structure of Deep Learning models and efficient improvements on building and training models, respectively. To explore how neural networks perform on image generation and classification, researchers mostly use apply visual analytics to understand the architecture employed in the model, like convolutional networks, recurrent networks, and generative models [18, 12]. We are going to utilize this method to understand how the hidden layers work on object detection and pose estimation models.

Video Understanding extends connectivity in Deep Learning models on images to a temporal dimension beyond the spatial dimension. Some Convolutional Neural Networks in action classification models [9] fuse continuous frames within a temporal window through high-level layers in the networks. Other models use 3D-ConvNet [2] and Recurrent Neural Networks [16] to capture the feature in actions. Besides the modification in architecture, other methods, like cycle-consistency loss which shows powerful performance in image generation [20], are implemented in the representation learning of actions [5].

3. Dataset

We use the MLB-YouTube dataset consisting of 20 baseball games from the 2017 MLB post-season provided in [13]. This paper works on activity classification on baseball game videos and it prepares two formats of video clips: segmented dataset and continuous dataset. The former dataset consists of 2128 1-2 minute long clips and the latter one contains 4290 video clips with several seconds. Since the segmented video dataset is prepared for the multi-label classification task, each clip may involve several activities.

In our project, however, we want short clips that have the same kind of activity and duration. Therefore, we use the continuous dataset for activity annotation.Then we extract 1.8 sec clips with the label of "swing" from continuous video clips. The activity of swing contains the process of that the pitcher throws the ball and the batter tries to hit it by swinging the baseball bat.

We use 10 fps to extract 19 frames from each clip and apply the object detection model to measure the detection scores on four main targets: pitcher, batter, catcher, and umpire. To reduce the noise caused by inconsistent features among frames, We only use clips of fixed-angle scenes. Therefore, the positions and areas of detection boxes are relatively stable among frames and the differences between clips are controlled in an acceptable range. As a result, we can take reasonable inference given the statistics of bounding boxes. Due to the limitation of manual labeling, we select 100 videos to implement the temporal analysis.

4. Method

4.1. Object Detection

We implement the most accurate version in Single-Shot Detector (SSD), SSD-Inception-v2, to the video clips we introduced in the last section. A COCO dataset pre-trained model of it is available on the Jetson TX2 Module, the hardware we choose to control cameras in the tracking system. Similarly, we use another COCO pre-trained model provided by Tensorflow object detection API. Based on Jetson-inference API, we compare three models on COCO 14 minival dataset (90 classes) in Table 4.1. So we can find that ssd_inception_v2_coco has the highest accuracy among the three versions even though it takes a longer time during processing. Since we do not have ground truth labels for object detection boxes in our dataset, we would directly use a pre-trained model.

SSD Models	Speed (ms)	COCO mAP
ssd_mobilenet_v1_coco	30	21
ssd_mobilenet_v2_coco	31	22
ssd_inception_v2_coco	42	24

Table 1. The Performance of SSD models on COCO 14 dataset. The measurements are provided by Tensorflow Object Detection API. Speed refers to the running time in ms per 600x600 image, and mAP is mean average precision.

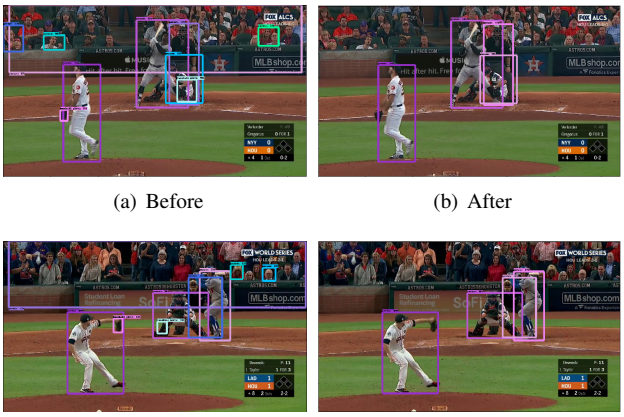
4.2. Remove Unrelated Boxes

Considering our purpose, we only need to keep the results of the class "Person". Furthermore, except for our four main targets, other people detected on the frames would not be taken into account in the final analysis. In order to remove this redundant information in detection results, we take the following steps to filter out invalid bounding boxes.

Hyper-parameters	min_area	max_area	depth_quantile
Value	0.015	0.19	0.64

Table 2. The hyper-parameters for filtering out invalid bounding boxes.

First, we remove the boxes without the label "swing". Second, we discard boxes that are too large or too small. That is because large boxes might be the prediction based on multiple real targets (i.e. combine the features of multiple people to generate a single box) and small boxes only consider part of a target. We cannot make any valid conclusions with these boxes. Third, we need to discard the boxes



(a) Before (b) After

(c) Before (d) After

Figure 1. The comparison between before and after filtering out unrelated boxes. Different colors of box boundaries are used to distinguish each box, and boxes with darker color have higher scores in Figure 1(b) and 1(d). In the descending order of detection scores, the targets are pitcher, batter, umpire and catcher in Figure 1(b), and the targets are pitcher, catcher, batter, and umpire in Figure 1(d).

of the audience on the back. Here we introduce a depth estimation model [1] to measure the approximate depth within the boxes. The main targets are closer to the camera so the average depth in their bounding boxes is smaller than the average depth of the audience boxes. Then we add a mask to discard bounding boxes the average depths of which are higher than the thresholds (i.e., certain depth quantile for each frame). The thresholds for filtering out invalid boxes are listed in table 4.2. The bounding box areas and depths are both normalized for each frame.

The results of the filtering process are shown in Figure 1. Then we manually assign the boxes to each target. If there lack the boxes for some targets, we would assign zero to the score and normalized area of the detection box. On the one hand, if there are multiple boxes referred to the same target, we would keep the one with the highest score. On the other hand, if a single box contains multiple targets, we would assign the box to the one standing closer to the camera.

5. Evaluation and Results

We choose 100 videos that have a consistent field (i.e., fixed camera angles) and implement the object detection model and our methods. To estimate the temporal changes in each target, we classify the results, boxes of each target, into 19 bins corresponding to 19 frames we extracted from each short clips. The scores and normalized areas of the boxes are presented in Figure 2.

5.1. Detection scores

Figure 2(a) and 2(b) plot the changes in detection scores along the order of frames in clips. Figure 2(b) shows the distribution of scores on four targets after discarding all zero

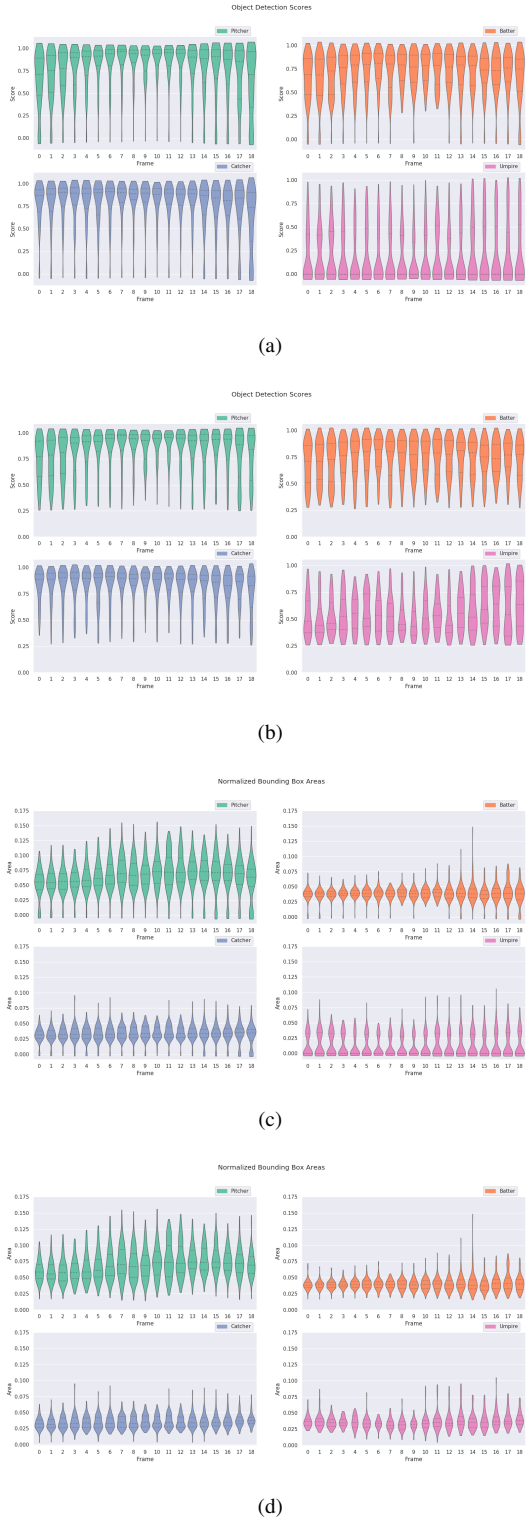


Figure 2. Figure (a) and (b) are scores of boxes plotted in temporal order, and Figure (c) and (d) are normalized areas of boxes.

observations. Based on the original observations in Figure 2(a), we notice that the object detection models cannot recognize the umpire on most frames. In the distribution of the umpires, most observations gather on the zero because um-

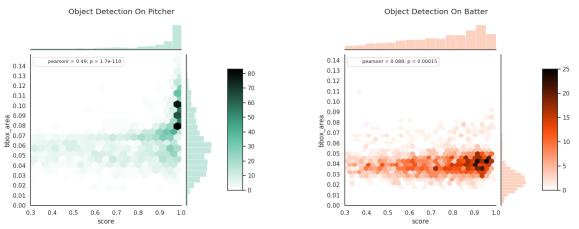
pire usually stands behind other players in the scenes captured in the short clips. To avoid the influence from the zero observations, we also plot the non-zero observations in the Figure 2(b) and Figure 2(d). The temporal observations about scores show an interesting difference between the pitcher and the other two players.

In terms of detection scores, the median scores on both the pitcher and the batter have a peak in the middle of the clips. The peak of median scores of the pitcher is sharper while the "peak" in batter has slight fluctuation. If we take a look into actions in the clips, we can link the changes in scores to the time pitcher and batter throw and swing, respectively. Pitchers would stretch out their arms and move their legs when they are trying to throw out the ball, so do batters. And batters' moves would come after pitchers' actions a little bit. Since our clips cannot guarantee that the speeds and reaction time of all players are the same and annotations of activities on original long clips are perfectly accurate. Under the effect of these factors, the time of throwing and swing varies among different clips, and the action time of batters would have more inconsistency because their actions correspond to the pitchers' actions. The complicated situation of the batters also explains why the temporal peak of median scores on them is not as smooth as the peak on the pitchers. In short, we can only have a general picture of the players' performance given the information shown in curves.

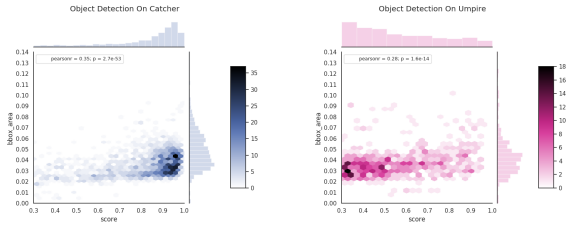
5.2. Normalized Box Areas

Additionally, the curves about the normalized box area reveal 3D spatial details. The areas of boxes on the pitchers are far larger than others because they always stand in front of other players (see Figure 2(d)), and large boxes also have more fluctuation over time correspondingly. However, the curves of areas of boxes on other players are relatively stable. And there is even no huge difference through frames, even across different players (i.e., batter, catchers, and, umpires) for the reason that their positions are close to each other during the swing activity. Particularly, their positions (i.e., the center of boxes) sometimes overlap. Given that, the attributes of the boxes are not helpful in the analysis of spatial information, unlike the detection scores.

Furthermore, we explore the connection between the scores and areas (see Figure 3). Given the Pearson's correlation coefficient, the scores and box areas on the pitcher and the catcher have moderate degree correlation, and the correlation on the umpire is on the low degree while there is nearly no correlation in the joint distribution on the batter. Obviously, we cannot conclude any patterns between the temporal changes in detection scores and box areas.



(a) Pitcher (b) Batter



(c) Catcher (d) Umpire

Figure 3. The joint distribution of detection scores and normalized box areas in all frames.

Players	Pitcher	Batter	Catcher	Umpire
# observations	1790	1842	1793	739

Table 3. The number of non-zero observations corresponding to distributions in Figure 3. The total number of observations including zero is 1900 for each player .

6. Conclusion

In sum, the performance of object detection models is under the effect of the actions of players. Usually, the model could capture higher scores during the action of certain players. On the contraru, the box areas mainly relate to the types of players.In the scenes of our clips, neither the positions on the images nor the depth of the players has much changes. Even though we can compare the detection results of different clips in a same domain as the fixed angle video share similar scenes and activities, the limitation brought by the single view of the sport field makes it impossible to evaluate more details of the players' actions.

To implement our methods on large dataset, we need to add an image classification model to replace manual labeling boxes in the next step. Then, we would try other video activity recognition models to explore the temporal-spatial connection among clips.

References

[1] I. Alhashim and P. Wonka. High quality monocular depth estimation via transfer learning. *arXiv e-prints*, abs/1812.11941, 2018.

[2] J. Carreira and A. Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4724–4733, 2017.

[3] J. Dai, Y. Li, K. He, and J. Sun. R-fcn: Object detection via region-based fully convolutional networks. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, page 379–387, Red Hook, NY, USA, 2016. Curran Associates Inc.

[4] D. Dwibedi, Y. Aytar, J. Tompson, P. Sermanet, and A. Zisserman. Temporal cycle-consistency learning. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1801–1810, 2019.

[5] D. Dwibedi, Y. Aytar, J. Tompson, P. Sermanet, and A. Zisserman. Temporal cycle-consistency learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[6] R. Girshick. Fast r-cnn. In *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, ICCV '15, page 1440–1448, USA, 2015. IEEE Computer Society.

[7] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 580–587, 2014.

[8] C. Godard, O. M. Aodha, and G. J. Brostow. Unsupervised monocular depth estimation with left-right consistency. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6602–6611, 2017.

[9] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1725–1732, 2014.

[10] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab. Deeper depth prediction with fully convolutional residual networks. In *2016 Fourth International Conference on 3D Vision (3DV)*, pages 239–248, 2016.

[11] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. Ssd: Single shot multibox detector. In *ECCV*, 2016.

[12] C. Olah, A. Mordvintsev, and L. Schubert. Feature visualization. *Distill*, 2017. https://distill.pub/2017/feature-visualization.

[13] A. Piergiovanni and M. S. Ryoo. Fine-grained activity recognition in baseball videos. In *CVPR Workshop on Computer Vision in Sports*, 2018.

[14] A. Piergiovanni and M. S. Ryoo. Learning shared multimodal embeddings with unpaired data. *arXiv preprint arXiv:1806.08251*, 2018.

[15] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6):1137–1149, June 2017.

[16] G. Varol, I. Laptev, and C. Schmid. Long-term temporal convolutions for action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(6):1510–1517, 2018.

[17] Q. Wang, Y. Ming, Z. Jin, Q. Shen, D. Liu, M. J. Smith, K. Veeramachaneni, and H. Qu. Atmseer: Increasing transparency and controllability in automated machine learning. In *ACM Conference on Human Factors in Computing Systems (CHI), 2019, Glasgow, Scotland UK*, 2019.

[18] M. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *Computer Vision, ECCV 2014 - 13th European Conference, Proceedings*, number PART 1, pages 818–833. Springer Verlag, 2014.

[19] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe. Unsupervised learning of depth and ego-motion from video. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[20] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networkss. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, 2017.