Fine grained action recognition in sports videos

Amit Nagpal (anagpal1@stanford.edu)

Abstract

Video, today, is searched and browsed primarily by its cover. To make "Stephen Curry's 3-pointers" accessible, publishers must create a separate video just for Stephen Curry's 3-pointers and label it as such. This is inefficient and has not scaled for obvious reasons. Moreover, an increasing population of sports fans do not wish to consume the entirety of every game. They are only interested in watching key games and "interesting parts" of all other games. Finer grained consumption of video content, through finer grained tagging such as action recognition, is key to the ecosystem of sports.

Unlike in everyday life, action recognition in sports is particularly hard due to the chaotic nature of game play. Occlusion is everywhere, players take on extreme body postures and subtle variations mean completely different moves. It is tough to scale, due to the expense of generating training data for each possible move and its variations. Our motivation for this project is to break down moves and learn to classify their common pool of semantic components such as run, jump and dribble in a multiplayer sports setting. We then apply the learning to identify domain specific fine grained moves. For this project we focus on moves in NBA.

1. Introduction

Action recognition in sports is an important area of research in Computer Vision. It has implications beyond consumption by sports fans. Once indexed it can be used by coaches, allowing them to pull from the vast repository of searchable fine grained moves, that they can then analyze and train on. It can be used by students to self learn from an endless stream of sports videos posted on the web. It can be used for automatic highlight generation for less known and hard to access games not hosted by NBA, NFL for example. Once available for any sports video ever recorded, action instances can be used as a basis for entity / athlete "similar to" recommendations, providing a rich platform for the discovery of new stars in the sports. Applications are endless.

Equally importantly it provides us with the motivation of pushing the state of the art in one of the most complex settings in computer vision. Team sports such as NBA involve a large number of players moving rapidly, often unpredictably and frequently converging into a very small area "over the top of each other", with hundreds of audiences in the background. This magnifies the challenges of occlusion, player identification, pose detection, tracking and other classic problems studied in computer vision.

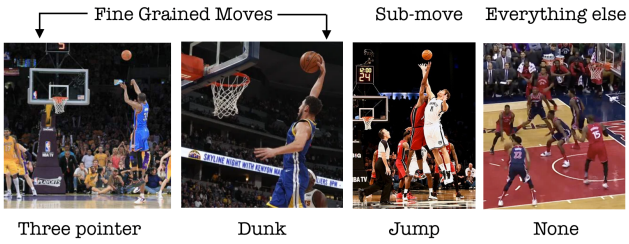


Figure 1. Classes

In this paper we explore the task of building models that can classify a video segment to contain low level semantic sub-moves (jump, dribble etc) as well as fine grained sports specific moves (three pointer, dunks, ..). We limit the scope of this project to classes shown in Figure 7. Our prediction inputs and model experiments are shown in Figure 5.2:

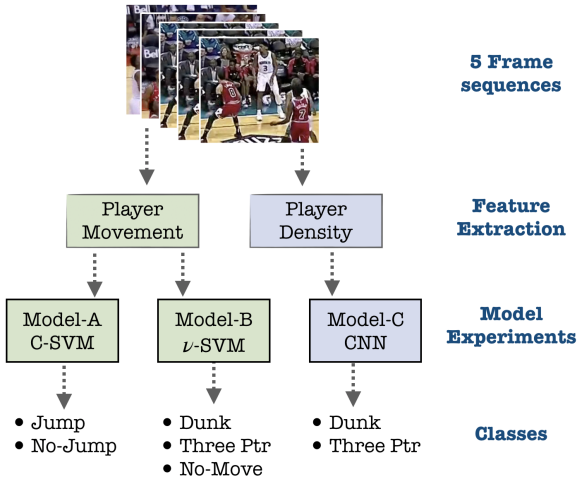


Figure 2. Final prediction cycle

Our solution can then be trivially applied to any length of video by continuously feeding it sequences of 5 frames to identify moves in a game.

2. Related Work

Action recognition is a widely researched topic. Some of the most prominent methods for action recognition can be roughly organized into the following categories:

- Hand-crafted features (Wang et al., 2011; Jain et al., 2013; Sun et al., 2018),
- Two stream neural networks (Simonyan & Zisserman, 2014; Feichtenhofer et al., 2016; Carreira & Zisserman, 2017; Zhu Y., 2019; Cai et al., 2019),
- 3D convolutional networks (Ji et al., 2013; Tran et al., 2018),
- Recurrent neural networks (Baccouche et al., 2010; Donahue et al., 2015),
- Pose-based methods (Yao et al., 2011; Luvizon et al., 2018).

These categories certainly overlap on many levels. The key strength of pose based methods and two stream neural networks is their ability to independently train spatial / temporal features and carry them from one domain to the other. This is important as the application of action recognition is vast and collection of training data costly. For example, there are dozens of "moves" in just NBA. Just trying to scale action recognition to all of NBA is hard, let alone scaling it to all of big 5 sports and chillingly hard to imagine scaling to all sports, let alone other non-sports activities.

3D CNNs have demonstrated better performance in capturing spatio temporal structure in videos than 2D convolutions have in the past. But the biggest drawback of 3D convolutions is the large number of parameters. This makes it particularly unsuitable for fine grained action recognition in practice. Due to the sheer number of classes defined in the taxonomy of any fine grained task, the datasets are usually very small and makes it easy to overfit with the large number of parameters 3D convolutions utilize.

Two stream networks, first proposed in (Simonyan & Zisserman, 2014), have become popular recently. A spatial stream, analyzes a single video frame, such as utilizing Pose as a high level spatial feature. In parallel, a temporal stream uses multi-frame optical flow. This is done via a series of convolutions and fully-connected layers. For final classification, the streams are fused together via averaging or an appropriate linear model. Being separate streams, they can be trained independently with a higher likelihood of transferring from one domain to another.

Our approach uses pose as a foundation. In the first set of models, we use tracking to capture temporal features and study the ability to model common semantics underlying multiple moves, in a hope that multiple fine grained moves can then be modeled with minimum training data over sub-moves such as a "jump" that require more raw features and larger training sets to learn. In our second model, we let the CNN model the temporal movement of "aggregate player movement" which too is a foundational sub-aggregate-movement over which we can build final move classifier using scanty training data.

Our primary motivation is the ability to scale classification to a large number of classes across a large number of sports, with minimum possible training data. For this reason, we try to break down action recognition into action proposal (semantic sub-moves / cues) and action classification (dunk, three pointer).

3. Data

One of the bottlenecks for progress in this area is the lack of open data sets with fine grained action tags. We combined two separate datasets to build a weakly supervised dataset to use as training data for this project.

Game clock in official videos from nba.com is extracted with OCR and reconciled with play by play available on sports.yahoo.com for each game. The work of fine tuning and running OCRs for video game clocks was already done and available for us to use. Structured play by plays in JSON format was also available for us to use as part of proprietary work done elsewhere.

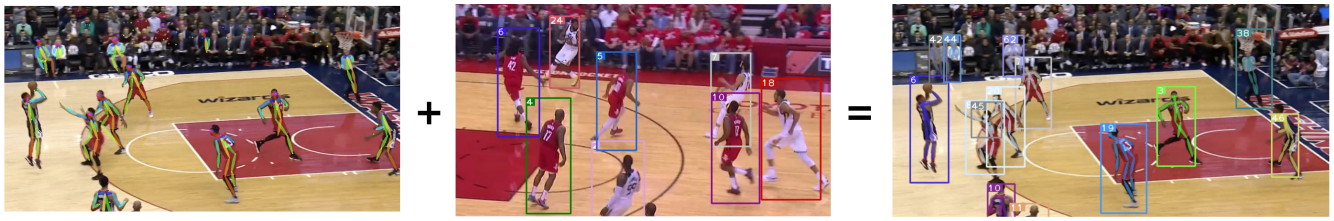
To generate a dataset for this project, we extracted 4.5 seconds of video segments around the time play-by-plays indicated a move had occurred. After reducing the videos from 30 fps down to 10 fps, we had a data set of 240, 45-frame video segments, a frame offset where the action happened and whether the action was a dunk or a three pointer. Using simple heuristics we filtered out bad videos and tags, resulting in a total data set size of 150 video segments.

4. Features

To provide enough semantics to our final classifier we extracted temporally tracked pose features of each player in every frame of the video. This was a three step process as described in section 4.1. To address camera motion and to normalize all location and distance vectors, we also extracted approximate location of the basket in every frame. Presence of basket was also used to only predict an action on frames that contained a basket.

4.1. Player spatio-temporal features

Figure 3 illustrates the three step process to create tracked poses of players. We first annotate every frame of the 5 second videos with player poses based on (Cao et al., 2017)'s approach and open source model. We then use DeepSort



2D pose - We annotate every frame of the 5 second videos with player poses [1]

Tracking - We use deepsort's [2] people model to track players

Matching - We then assign poses to tracks based on maximum body containment,

Figure 3. Tracked pose estimation

(Wojke et al., 2017) and its open source model to find bounding boxes of players that are tracked across the video.

To assign poses to tracks we then use a simple yet effective approach of maximum pose and bounding box intersection on a frame by frame basis. We iteratively pair tracks with poses based on maximum ranked intersections, removing them from the matching process once paired. The simple approach did introduce fickleness of assignment during occlusion and crowding scenarios. But we saw only minimal effect on the recall of our final classification. Adding jersey color to provide consistency through occlusion / crowding scenarios may be an easy trick to increase the effectiveness of pairing. Though, the best approach here will be to combine both pose and tracking into a single optimization.

4.2. Basket to fix origin

To avoid creating a basket specific bounding box, given the time constraint, we used the location of the basket holder's banner ad as a proxy for basket detection. We used Amazon's Rekognition text detection and recognition model available as an AWS service and added appropriate offsets to approximate the location of the basket in a frame. Intermittent frames that were not detected by Amazon's Rekognition were "smoothed" by fitting a straight line travel between previous and next known locations of the basket. We then transformed all player pose vectors to be relative to the new origin we fixed at the basket.

Figure 4. Approximating basket location

4.3. Aggregate features

4.3.1. Player density

For every frame, we calculate density of players in different regions of the court. For distinguishing between a three pointer and dunks, we only needed to group players into 'far' (yellow) and 'near' (blue) regions that mimic the shape of D around the basket. We filtered out audience by requiring them to have a minimum pose height and distance relative to the basket, without which grouping of poses was also skewed towards where the audience was.

Figure 5. Player density

4.3.2. Player movement

For every track and every frame we also calculate the minimum and maximum change in Y coordinate, over the last 5 frames, of the pose's mean and the number of times it had its arms raised.

5. Methods

There is always a very high number of classes defined in the taxonomy of any fine grained task. As a result, datasets available for such tasks are also small. To validate the hypothesis that we can learn to classify fine grained actions over a common set of separately learnt sub-actions or action proposal events, we model three SVMs that work with aggregate player movement and one CNN model that tracks changes in player densities.

5.1. Classification based on Pose movement

Our first method for classification is based on the activity of each individual player via its pose features across the previous 5 frames.

Learning algorithms such as Support Vector Machines (SVM) accept fixed size vectors and cannot work with varying sizes. Number of players and their body parts that we are able to identify varied from one frame to another. In order to benefit from various learning techniques, we needed a mechanism to aggregate sets of local features into discriminative and fixed-size descriptors. So, we pre-aggregated player movement features as described in section 4.3.2.

We define the following aggregate and raw feature names:

- X,Y = mean distance of pose from basket
- dY = change in Y w.r.t. last 5 frames of a track
- A = number of times pose's arms were up over the last 5 frames

5.1.1. C-SVM: Action Proposal

Here we learn to classify a sequence of 5 frames as containing or not containing a jump by any player on the court. To handle class imbalance between hard-to-find jumps and always-there no-jumps, we use C-SVM so we can separately tune mis-classification costs for jumps vs no-jumps.

$$\begin{aligned} \underset{w,b,\xi}{\text{argmin}} \quad & \frac{1}{2}||w||^2 + C\left[C_1 \sum_{\{i|y_i=1\}} \xi_i + C_{-1} \sum_{\{i|y_i=-1\}} \xi_i\right] \\ \text{subject to} \quad & y_i(w^T x + b) \geq 1 - \xi_i. \end{aligned}$$

This classifier is run on every track identified in every frame of a given video, to identify the frame and x,y coordinate of a jump, if any. Input features to this algorithm is a 3 element vector per frame, $F = [min(dY), max(dY), A]$

5.1.2. ν-SVM: Action Recognition over Proposal

For the frames that we know a jump occurred, we then write a classifier to demonstrate how most fine grained classification can be trivially built over common sub moves / events / action proposals. Here we use a simple ν-SVM with a

$$\begin{aligned} \min_{\boldsymbol{w},b,\boldsymbol{\xi},\rho} \quad & \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} - \nu\rho + \frac{1}{l}\sum_{i=1}^{l}\xi_i \\ \text{subject to} \quad & y_i(\boldsymbol{w}^T\phi(\boldsymbol{x}_i) + b) \geq \rho - \xi_i, \\ & \xi_i \geq 0, i = 1, \ldots, l, \quad \rho \geq 0. \end{aligned}$$

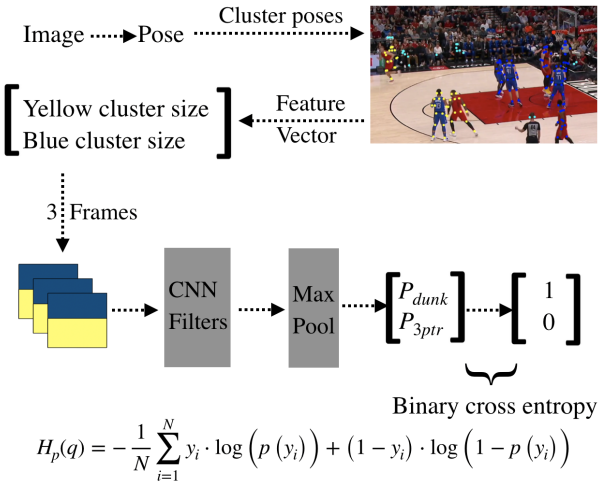
feature vector [X, Y] representing the jump location.

5.1.3. ν-SVM: Single Step Action Recognition

To demonstrate that joint models can be easily built once underlying semantic components have been identified, we simply concatenate the feature set of previous two models as $F = [min(dY), max(dY), A, X, Y]$ to train a one step fine grained classification model, again using ν-SVM.

5.2. Classification based on player density shifts

To model another semantic sub component, we model the fine grained classification by training a CNN to learn how player density patterns shift differently across frames between a dunk and a three pointer. Due to scanty training data we feed the CNN pre-aggregated notion of player densities and only train the CNN to learn the shift pattern as shown below:



$$H_p(q) = -\frac{1}{N}\sum_{i=1}^{N} y_i \cdot \log\left(p\left(y_i\right)\right) + \left(1 - y_i\right) \cdot \log\left(1 - p\left(y_i\right)\right)$$

We use binary cross entropy to train our probabilities against one class label setting. We run CNN filters of size 1 and size 2 over the input matrix $\in R^{3\times 2}$ to capture relative densities within a frame and across frames sequences.

6. Experiments, Results and Discussion

We trained the first C-SVM model that predicts Jump vs No-Jump with 89 positive and 306 negative examples of jumps. Positives were capped by how many videos we were able to manually label. For negatives we programmatically picked frame sequences that would be a close match to a jump, for example, when players had their arms raised, or their mean Y suddenly increased over consecutive frames. We wanted to capture some variations in poses that looked like jumps, but weren't. We did a 80/20 train/test split. During our first try, the model learnt to almost always classify everything as not a Jump. To handle the class imbalance, we set misclassification weight for Jumps to 22 and no-jumps to 1. Were were able to achieve accuracies of 76.71% and 85.55% on training and test sets.

MODEL	TRAIN	TEST
POSE MOVEMENT MODELS		
C-SVM PROPOSAL ONLY	76.71%	85.55%
ν-SVM RECOGNITION	96.0%	98.22%
ν-SVM JOINT MODEL	87.59%	85.56%
PLAYER DENSITY MODEL		
CNN	92.21%	98.43%

Table 1. Classification accuracies.

To demonstrate building simple classifiers over common semantic events, we trained a trivial model that classified a jump event to be a dunk vs three pointer based on location of of a jump event (action proposal). We unsurprisingly achieved accuracies of 96 and 98% on train and test sets.

Finally, we trained a joint ν-SVM model as a multi-class classifier (Dunk, Three pointer, No-Move) over a concatenated feature set of the previous two classifiers. We set $\nu = 0.155$. We arrived at the optimum ν by starting at 0.5 and following the direction of max performance in a gradient descent like manner.

We designed the CNN to learn patterns of change in player density across the court. After filtering out audience, we were able to achieve good accuracies around 1000 epochs. Please see Figure 6 for progression of training. We used 4 set of size 1 and size 2 filters, standard adam gradient descent and sigmoid activation for the dense layer.

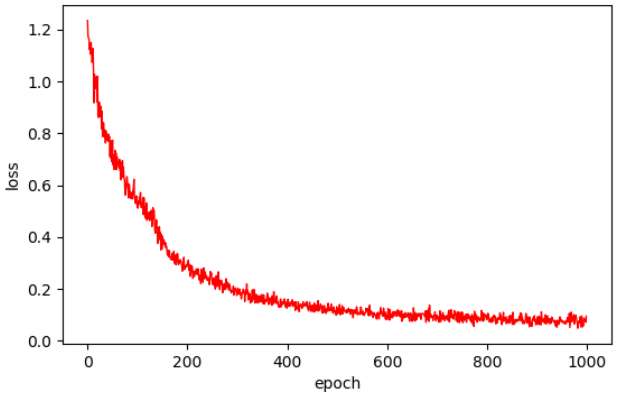


Figure 6. CNN training loss

The two most difficult issues we faced were:

- Pose estimation and tracking in key frames - Key action frames are often occluded. Player poses are often missing key joints and limbs right when they are most needed. Right when the player hangs on the basket or jumps to make a dunk or three pointer. A joint model for pose and tracking with regularization to encode semantics / constraints of a particular sport will greatly help here. This aspect had the biggest impact on the quality of predictions.

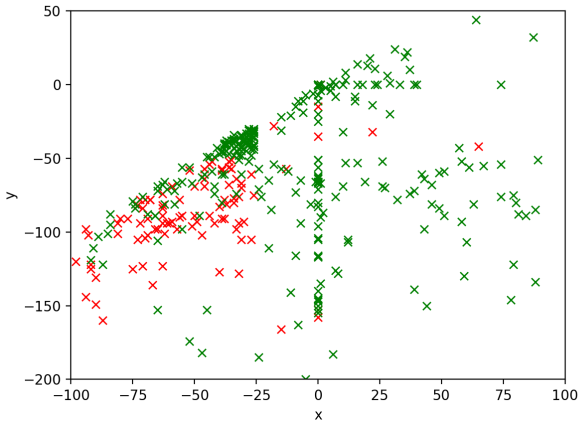


Figure 7. Player movement dY and dX features

- Noise from audience - Noise filtering was not a problem at with the SVMs as they were trained on individual player movement. Audience don't move. Where-as we had to aggressively filter out audience to get reasonable performance out of our CNNs. We need more algorithmic ways of filtering out audiences.

7. Conclusion and Future Work

Fine grained action recognition is an important area of research. It can be overwhelming, but we think, if we break it down into a set of semantics such as "jump", "dribble", "group movement" that are few and are possible to learn from large data sets outside of each specific domain, it is possible to train fine grained action recognition with minimal training data which is key to scaling any practical application across a large taxonomy of fine grained actions across a long list of sports.

The next step would be to build a joint model between pose estimation and tracking that can handle occlusion and crowding better. But more importantly formalize a way to model some core semantics specific to NBA or any given sports as regularization in the joint model to leverage known constraints for prediction performance when it is needed the most. We would like to source more training data and implement ways to work with weakly supervised data in a more robust way.

8. Code

https://github.com/amitnagpal229/cs229project

References

Baccouche, M., Mamalet, F., Wolf, C., Garcia, C., and Baskurt, A. Action classification in soccer videos with long short-term memory recurrent neural networks. In *Proceedings of the 20th International Conference on Artificial Neural Networks: Part II*, ICANN'10, pp. 154–159, Berlin, Heidelberg, 2010. Springer-Verlag. ISBN 3-642-15821-8, 978-3-642-15821-6. URL http://dl.acm.org/citation.cfm?id=1889001.1889024.

Cai, Z., Neher, H., Vats, K., Clausi, D. A., and Zelek, J. Temporal hockey action recognition via pose and optical flows. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2019.

Cao, Z., Simon, T., Wei, S.-E., and Sheikh, Y. Realtime multi-person 2d pose estimation using part affinity fields. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

Carreira, J. and Zisserman, A. Quo vadis, action recognition? a new model and the kinetics dataset. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

Donahue, J., Anne Hendricks, L., Guadarrama, S., Rohrbach, M., Venugopalan, S., Saenko, K., and Darrell, T. Long-term recurrent convolutional networks for visual recognition and description. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

Feichtenhofer, C., Pinz, A., and Zisserman, A. Convolutional two-stream network fusion for video action recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

Jain, M., Jegou, H., and Bouthemy, P. Better exploiting motion for better action recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2013.

Ji, S., Xu, W., Yang, M., and Yu, K. 3d convolutional neural networks for human action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(1):221–231, Jan 2013. ISSN 1939-3539. doi: 10.1109/TPAMI.2012.59.

Luvizon, D. C., Picard, D., and Tabia, H. 2d/3d pose estimation and action recognition using multitask deep learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

Simonyan, K. and Zisserman, A. Two-stream convolutional networks for action recognition in videos. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N. D., and Weinberger, K. Q. (eds.), *Advances in Neural Information Processing Systems 27*, pp. 568–576. Curran Associates, Inc., 2014. URL https://bit.ly/35j1eg9.

Sun, S., Kuang, Z., Sheng, L., Ouyang, W., and Zhang, W. Optical flow guided feature: A fast and robust motion representation for video action recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

Tran, D., Wang, H., Torresani, L., Ray, J., LeCun, Y., and Paluri, M. A closer look at spatiotemporal convolutions for action recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

Wang, H., Klaser, A., Schmid, C., and Liu, C.-L. Action recognition by dense trajectories. In *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '11, pp. 3169–3176, Washington, DC, USA, 2011. IEEE Computer Society. ISBN 978-1-4577-0394-2. doi: 10.1109/CVPR.2011.5995407. URL https://doi.org/10.1109/CVPR.2011.5995407.

Wojke, N., Bewley, A., and Paulus, D. Simple online and realtime tracking with a deep association metric. In *2017 IEEE International Conference on Image Processing (ICIP)*, pp. 3645–3649. IEEE, 2017. doi: 10.1109/ICIP.2017.8296962.

Yao, A., Gall, J., Fanelli, G., and Van Gool, L. Does human action recognition benefit from pose estimation? pp. 67.1–67.11, 01 2011. ISBN 1-901725-43-X. doi: 10.5244/C.25.67.

Zhu Y., Lan Z., N. S. H. A. Hidden two-stream convolutional networks for action recognition. in: Jawahar c., li h., mori g., schindler k. (eds) computer vision – accv 2018. accv 2018. lecture notes in computer science, vol 11363. springer, cham, 2019.