What will Happen Next? Forecasting Player Moves in Sports Videos

Panna Felsen
UC Berkeley, STATS
panna@berkeley.edu

Pulkit Agrawal
UC Berkeley
pulkitag@berkeley.edu

Jitendra Malik
UC Berkeley
malik@berkeley.edu

Abstract

A large number of very popular team sports involve the act of one team trying to score a goal against the other. During this game play, defending players constantly try to predict the next move of the attackers to prevent them from scoring, whereas attackers constantly try to predict the next move of the defenders in order to defy them and score. Such behavior is a prime example of the general human faculty to make predictions about the future and is an important facet of human intelligence. An algorithmic solution to learning a model of the external world from sensory inputs in order to make forecasts is an important unsolved problem. In this work we develop a generic framework for forecasting future events in team sports videos directly from visual inputs. We introduce water polo and basketball datasets towards this end and compare the predictions of the proposed methods against expert and non-expert humans.

1. Introduction

In 2002, Billy Beane defied conventional wisdom by performing meticulous statistical evaluations of undervalued players to assemble the Oakland Athletics baseball team on a negligible budget. His team made history with a record-setting 20-game win streak, and this tremendous feat is documented in the academy award nominated film *Moneyball*. Their success made an unprecedented case for competitive advantages gained by new analyses of individual players' game play. Now imagine if, in addition to knowing the shot success rate of Stephen Curry, the best basketball shooter, it is also possible to forecast that he is more likely to attempt a shot within zero, one, and two seconds of a pass when his teammates are in a diamond, ring, and triangle formation, respectively. Such forecasts are invaluable to the defending team in planning strategy. Billy Beane's analysis revolutionized strategic thinking in baseball, and similarly, we believe statistical methods for forecasting player moves have the potential to impact how teams plan their play strategies.

Predicting player moves in sports videos is an instance of a much grander research agenda to develop algorithms that can forecast future events directly from visual inputs. The ability to forecast is a key aspect of human intelligence, and as Kenneth Craik famously wrote in 1943, "*If the organism carries a 'small scale model of external reality and its own possible actions within its head, it is able try out various alternatives, conclude which is the best of them, react to future situations before they arise and in every way react in much fuller, safer and more competent manner to emergencies which face it.*" While there has been a lot of interest in this problem [14, 28, 35, 12, 24, 36, 1, 16, 8, 26, 44, 38, 37, 2], we lack a good benchmark for comparing different forecasting algorithms.

For multiple reasons, it appears to us that team sports videos are a very good benchmark for evaluating forecasting algorithms. Firstly, many human activities are social and team sports provide an opportunity to study forecasting in an adversarial multi-agent environment. Secondly, team sports are composed of a large and diverse set of discrete events, such as passing, shooting, dribbling, etc. The sequence of events reflects the game play strategies of the two teams, and thus forecasting requires game specific knowledge combined with other visual cues, such as player pose and game state. This implies that for any system to make accurate predictions directly from visual imagery, it must distill game specific knowledge by crunching large amounts of data. Representing such knowledge is a central problem in forecasting, which is put to test in this setup. Expert players and coaches gain such knowledge via experience gathered over long periods of time. An additional benefit of predicting discrete events is crisp and straightforward evaluation of the information of interest that avoids the problems associated with evaluating pixel-level predictions.

In this work, we present a generic framework for predicting future events in team sports directly from visual inputs, and we introduce water polo and basketball datasets for evaluation. These datasets contain game stream accompanied by annotations of player tracks and seventeen different events. The task of interest is, given a history of obser-

vations, predict what event will happen immediately, after 1s, or after 2s. The seventeen events are answers to questions that are of great interest in team sports such as - *where will the ball go next? will the player score? will there be a "screen" event? will there be a block? will there be a turnover? will there be a dribble?* among many others.

We construct two set of models - ones that forecast from the raw video stream without any pre-processing and other that transform the raw video stream into an "overhead" representation where the players and balls are represented as dots on the playing field prior to forecasting. Using the water polo dataset as a case study, we present the entire system to convert images captured from a single moving camera into the overhead representation which is then fed into the predictor. We find that the overhead representation leads to more accurate predictions than raw image based representation. The performance of our system is close to humans but worse than water polo experts. We then apply the same set of forecasting techniques on a dataset of basketball games and show that our system outperforms humans on forecasting events in basketball games. While we present results on water polo and basketball, we make no game specific assumptions. The techniques developed in this work apply to a wide number of other team sports such as hockey, American football, soccer, handball, lacrosse and rugby.

Our main contributions in this work are: (1) Providing water polo and basketball datasets along with detailed annotations and human performance metrics as a benchmark for prediction tasks in adversarial multi-agent environments. (2) Putting forward a framework and machinery for converting images into overhead views and making predictions using both the image space and the overhead view representation. (3) We find that random forests outperform neural networks on our datasets. We suspect that this is due to the fact that neural networks are extremely data hungry. This raises a very interesting question - what auxiliary tasks can we pretrain on to improve prediction performance.

2. Related Work

Video analysis is an active research area. A large body of work has focused on action recognition [5, 42, 20, 31, 6, 30], people and object tracking [33, 41, 43]. In contrast to these works we are interested in the problem of forecasting. Predicting pedestrian trajectories [15, 17, 13, 14, 29] and anticipating future human activities [14, 16, 35, 45, 19, 10] has seen considerable interest over the past few years. However, these works mostly consider predicting events related to a single human, while we attempt to forecast events in multi-agent environments involving adversarial human-human interaction. Other works have explored predicting future world states from single images [7, 38, 25, 8], but have been limited to simulation environments or involve a single agent. Predicting pixel values in future frames has also drawn considerable interest [24, 28] but is limited to very short term predictions.

Sport Video Analysis: Traditional work in computer vision analyzing sports videos [3] has focused on either tracking players [11] or balls [23]. Another body of work assumes the annotations of ball or player tracks to analyze game formations or skill level of individual players. For instance, [39] use tracks of ball trajectories in tennis games to predict where the ball would be hit, [4] analyze soccer matches using player tracks. [22] discover team formation and plays using player role representation instead of player identity. More recently techniques such as [27] have looked at the problem of identifying the key players and events in basketball game videos. Closest to our work is the work of [40] that proposes the use of hidden conditional random fields for predicting which player will receive the ball next in soccer games. They assume the knowledge of game state such as attack, defense, counter attack, free kick etc. and assume that identity of players is known. In contrast, we present a forecasting system that works directly from visual inputs. It either uses images directly or converts them it into an overhead view representation using computer vision techniques. We do not require any external annotations of the game state.

3. Team Sports Datasets

We have focused our efforts on the most popular style of sport, *team goal sports*. We select water polo and basketball as two canonical examples because together they capture many diverse aspects of team goal sports: basketball is fast-moving and high-scoring like hurling and handball, while water polo is low-scoring like soccer and has man-up situations like hockey and lacrosse. Despite the different nuances of each team goal sport, they all share many common "events" during game play. For example, players advance the ball toward the goal themselves in a *drive*, and sometimes this results in a *goal* and other times in a *block* or a *missed shot*. Players *pass* the ball to their teammates, and sometimes the defense intercepts the pass.

3.1. Water polo

A water polo game naturally partitions into a sequence of alternating team possessions. During a possession, the attacking team's field players primarily spend time in their *front court*, which accounts for most of the interesting game play. The attacking team is required to take a shot within 30s, and failure to do so results in a *turnover*. Players of the two teams wear dark colored (typically blue/black) and light colored (typically white) caps. In the remainder of the paper we use dark-cap and light-cap to refer to two teams.

We collected a dataset of front court team possessions from video recordings of 5 water polo games between Di-

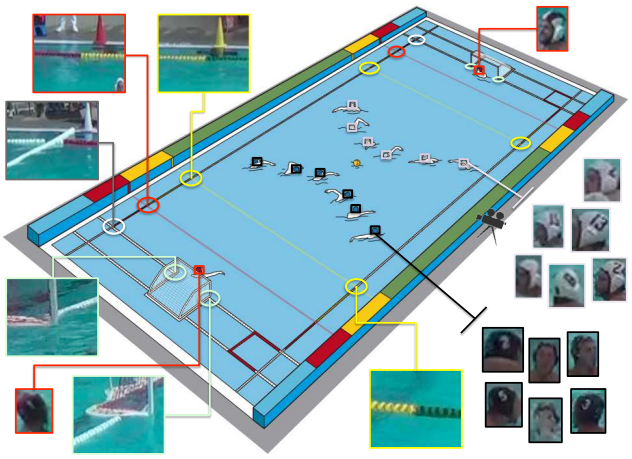

Figure 1: From single-camera video recordings of 5 water polo games, we collected bounding box track annotations of dark, light, and red-cap player heads. We also collected annotations of pool points marking the field of play: the 2m and 5m lines, the corner of the field, and the points where the cage and lane line meet.

vision I National Collegiate Athletic Association varsity teams. Similar to the NBA for basketball, this represents the highest level of water polo play in the United States. We chose to focus only on front court possessions, as most interesting events happen during this period. The time intervals of the front court possessions were hand-marked by an expert water polo coach. All the games, four of which are men's games and the other a women's game, are played at the same outdoor pool on different days at times ranging from morning until night; the dataset exhibits a large range of lighting conditions. The games were recorded with a single freely moving camera that pans between each side of the pool with resolution 720p at 25-30fps. Often the camera is adjusted for a new front court possession, resulting in varied camera motions and zooms.

Player and Pool Track Annotations: Bounding box track annotations (Figure 1) of dark and light-cap player heads, goalkeepers, and the head of the player in possession of the ball were collected using the VATIC toolbox [34] and Amazon Mechanical Turk. Player possession is defined to begin at the moment a player grasps the ball and ends at the moment that same player passes/shoots the ball or another player steals the ball. Additional annotations of specific points marking the field of play: the 5m line, the 2m line, the pool corner, and the cage posts were obtained. These field markings provide necessary point correspondences between the image view and overhead view of the game, which enable the computation of the player trajectories in the overhead space from the player locations in the image view. For increased data diversity, annotations were collected for 11 quarters of play from 20 quarters available in the 5 games.

Train/Test Splits: The splits were as follows - *train*: 7 quarters, randomly sampled from the first 4 games; *validation*: light-capped team front court possessions in all 4 quarters of the fifth game; and *test*: dark-capped team front court possessions in all 4 quarters of the fifth game. In total, each split has 232, 134, and 171 respective examples of a player passing the ball in a team's front court.

Human Benchmark: Human subjects were shown every test image taken just before a player loses possession of the ball and were required to draw a bounding box around the head of the player which they thought would possess the ball next. Two sets of subjects: nine non-experts and four water polo experts were evaluated. Non-experts had never seen or played a water polo game. In order to account for their inexperience, non-experts were shown all examples used to train computer algorithms along with the the ground-truth answer before being asked to make predictions. The experts had all played competitive water polo for at least four years. Expert and non-expert humans accurately predicted the next ball possessor 73% and 55.3% of the time respectively.

3.2. Basketball

The dataset is comprised of ground truth (in contrast to water polo, where it is computed) 2D player and ball trajectories, sampled at 25 Hz, in 82 regular-season NBA games obtained using the STATS SportVU system [32], which is a calibrated six-camera setup in every NBA arena. The data includes labels for 16 basketball events, including free throw, field goal, pass, dribble, (player) possession, etc. that are detailed in the supplementary materials.

Train/Test Splits: A total of roughly $300k$ labeled events were randomly split into $180k$, $30k$, and $90k$ for train, validation, and test examples.

Human Benchmark: A set of 18 subjects familiar with basketball were shown a series of fifteen 5-second clips of basketball data, ending with a labeled event. The ball and player trajectories were removed from the final n seconds of the clip, and the subjects were asked to predict the event at the end of the blanked portion. For each $n \in \{0.4, 1, 2\}$, each subject was shown 5 examples randomly sampled from a pool of 80 examples (5 examples of each of the 16 events). Humans were correct 13.5%, 20.6%, and 24.4% for $n = 2$, 1, and 0.4, respectively.

4. Methods: From Images to Overhead View

2D overhead view of the game where players are represented as dots at their (x, y) coordinate locations is often used by coaches because it provides immediate insight into player spacing, and distills player movement into a canonical, simple representation that is easy to compare across

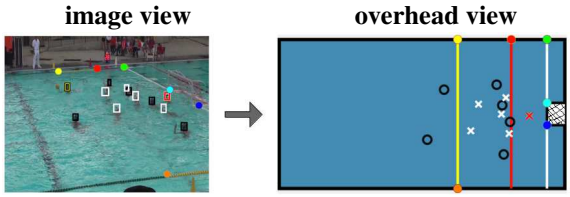

Figure 2: The image is converted into the overhead view by first estimating the homography between the image a canonical model of the playing field using field markings such as 2m/5m lines etc. The players are then detected and their locations are transformed using the estimated homography.

many game plays. We construct the overhead representation by first detecting players and ball. Using knowledge of playing field dimensions and locations of few landmarks, we estimate a homography to transform these detections into a canonical frame. We then link players across frames using tracking. Each step of this process is detailed below.

Player Detection: We finetune VGG-16 network pretrained on Imagenet for detecting light and dark cap players using Fast R-CNN and the annotations we collected described in section 3.1. The performance of dark and light color cap person detectors was 73.4% and 60.4%, respectively. We attribute the worse performance of the light-color cap detector to a few confounding factors: 1) many light-color caps were annotated, by one turker, with loose bounding boxes, 2) overhead lights produce specularities and water splashes can appear visually similar to light-color caps.

Player Tracking: We track by detection. The Hungarian matching algorithm [18] is used to link Fast-RCNN player detections to form player tracks. The pairwise affinity between detections in two sequential frames is a linear combination of Euclidean distance and bounding box overlap.

Overhead Projection: In the case of water polo (Figure 2) we used the annotations of 2m and 5m lines, the pool corner, and the cage posts to estimate the homography between the current image captured by the camera and a canonical 2D coordinate system representing the field of play using the normalized direct linear transformation (DLT) algorithm [9]. Next, we transform the midpoint of bottom edge of the player bounding box into a (x, y) location in the canonical frame. We use the bottom edge because that is the point of the player that is closest to the field of play, which in turn is mapped to the canonical frame by the homography transformation.

5. Forecasting Future Ball Position

The movement of the ball determines the game outcome, and therefore, it is the most important object in play at any moment of the game. We focus directly on the most important question during the game: where will the ball go next? We study two slightly different variants of this question: In the water polo dataset, we only consider the frame before which the ball possessor is about to lose of the possession of the ball, and we try to forecast which player will be in possession of ball next. In the basketball domain, we have access to much more data, and we additionally attempt the more general problem: where will the ball be in one or two seconds in the future?

5.1. Water polo: Who will possess the ball next?

In the typical front court water polo scene, there are 6 field players on the attack, defended by 1 goalkeeper and 6 field players on the opposing team. For example, in Figure 2, the dark-cap players are on the attack and the light-cap players are on defense. By definition, one of the attacking team players is in possession of the ball. Our system takes as input the frame just before the player loses ball possession by either making a pass to a teammate, shooting the ball, or committing a turnover. The task is to predict which player will possess the ball next.

A random choice of player from either team would be correct roughly $\frac{1}{12} \approx 8.3\%$ of the time. As a player is more likely to pass the ball to his teammate, a random choice of player from the same team would be correct approximately 20% of the time (empirically validated on the test set). Such random guesses are very naive. Players often tend to pass the ball to nearer teammates, as shorter passes are easier to execute and minimize turnover risk. Predicting the nearest teammate as the next possessor is correct 28.1% of the time. Players also tend to pass the ball to open teammates, those who are not closely guarded by defenders. Predicting a pass to a teammate who is furthest from his nearest defender (i.e. most open) has accuracy of 36.7%. These baselines are considerably worse than an average human with no water polo expertise, who is correct 55.3% of the time.

In the next two sections, we describe how performance can be improved: (1) using additional player features estimated from the overhead representation, and (2) automatically learning feature representations directly from the input image. We operationalize these approaches in the following way: Let there be K players each with feature vector $F^i (i \in \{1, 2.., K\})$, let $b \in \{1, 2.., K\}$ be a discrete random variable that encodes the player in possession of the ball after a pass is made. The goal is to find the player who is most likley to receive the ball, i.e. $\text{argmax}_i \, P(b = i | F^1..F^K)$.

5.1.1 Hand designed features from overhead view

When deciding where to pass the ball, players consider which teammates are in good position to: score, advance the ball, and receive a pass. We formalize these insights and characterize each player using a 9-D feature vector extracted from the overhead representation: the (x, y) player coordinates, the (x, y) coordinates of the nearest player on

$F_{[1,2]}$: (x,y) of player with ball
$F_{[3,4]}$: (x,y) of player
$F_{[5,6]}$: (x,y) of nearest defender
F_7: same-team flag
F_8: $||F_{[3,4]} - F_{[1,2]}||_2$
F_9: $||F_{[3,4]} - F_{[5,6]}||_2$

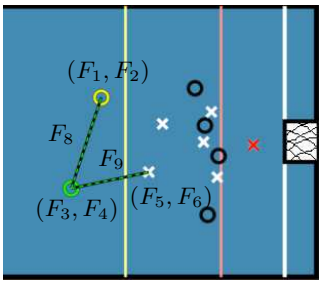

Figure 3: The features $F_{[1...9]}$ extracted from the 2D overhead view are used to train a random forest to classify players as either receiving the ball next or not.

the opposite team, the (x, y) coordinates of the player in possession of the ball, an indicator flag for whether the player is on the same team as the player in possession of the ball, and the Euclidean distances of the player to the player with the ball and to his nearest defender. This player-centric feature vector is illustrated in Figure 3. We assume that features $F^1..F^k$ are mutually independent, and therefore computing $P(b = i|F^1..F^K)$ reduces to estimating $P(b = i|F^i)$.

We train a system to infer which player will possess the ball next in the following way: we used the pipeline described in section 4 to convert the raw image into it's corresponding overhead representation. Next, feature vector of each player was computed from the overhead representation. Finally, a random forest classifier was trained on these features using the training data to estimate $P(b = i|F^i)$. Five-fold cross-validation was performed to chose the optimal depth and number of trees. This system achieved a performance of 45.5% (see Table 1) and outperformed the baseline methods on the testing set. Analysis of the results revealed that this method is biased towards predicting the most open perimeter player as the one receiving the ball.

A common failure mode is predicting an open perimeter player, when he is not even facing the player in possession of the ball. These mistakes are not surprising as the overhead view has no access to visual appearance cues. Another possible reason for failures is that the pipeline for converting image data into overhead representation is inaccurate. To tease this apart, we re-ran the analysis using ground truth (instead of estimated) detections. As reported in Table 1, the accuracy gap with and without using ground truth detection is within the error bar of the performance on the testing set. This suggests that the pipeline for obtaining overhead representation is accurate and further performance improvements will be gained by building better forecasting models.

5.1.2 Forecasting directly from image space

While the overhead view provides a good representation for analyzing game play, it loses subtle visual cues, such as the pose of the player and direction they are facing, that might be very relevant for forecasting. Instead of hand-designing such features, is it possible to automatically discover features that are useful for forecasting next ball possession?

Method	Ground Truth Heads	Detected Heads
Random, either team	9.5 ± 2.2	9.2 ± 2.2
Random teammate	19.1 ± 3.1	17.0 ± 2.8
Nearest neighbor teammate	28.1 ± 3.4	22.2 ± 3.2
Most open teammate	36.7 ± 3.7	28.7 ± 3.4
$F\,[8 \ldots 9]$	42.5 ± 3.8	35.2 ± 3.6
$F\,[7 \ldots 9]$	45.4 ± 3.4	38.4 ± 4.0
$F\,[3 \ldots 9]$	$\mathbf{48.8 \pm 4.3}$	44.1 ± 3.7
$F\,[1 \ldots 9]$	47.1 ± 3.8	$\mathbf{45.5 \pm 3.5}$
FCN, teammate	38.1 ± 3.5	35.2 ± 3.6
Human, Non-Expert	55.3 ± 7.9	-
Human, Expert	73.1 ± 2.0	-

Table 1: Each row reports accuracy of a different method for predicting which player will possess the ball next. The first four methods are baselines. The intermediate rows provide an ablation study of using various features defined above. The FCN is a deep learning based method and the last two rows report human performance. Performance metrics are reported for two circumstances: using ground truth player locations (column 1) and when detected instead of ground-truth locations (column 2) are used.

The set of features $F^1..F^k$ is represented by image I_t and we compute $P(b = i|I_t)$ in the following manner: Let l_b, p^k be random variables denoting the future location of the ball and the k^{th} player respectively after the passed ball is received. Since only one player can receive the ball, we assume that if the ball is at location l_b it will be received by the player who has highest probability of presence at l_b (i.e. $\arg\max_k P(p^k = l_b)$. Let l_b^i denote the set of all locations at which $i = \arg\max_k P(p^k = l_b)$. With this,

$$P(b = i|I_t) = \int_{l_b \in l_b^i} P(p^k = l_b, l_b|I_t) \quad (1)$$

assuming conditional independence,

$$= \int_{l_b \in l_b^i} P(p^k = l_b|I_t)P(l_b|I_t) \quad (2)$$

We model $P(l_b|I_t)$ using a Fully convolutional neural network (FCN; [21]), that takes I_t as input and predicts a confidence mask of the same size as the image encoding $P(l_b|I_t)$. The ground truth value of mask is set to 1 in pixel locations corresponding to bounding box of the player who receives the ball and zero otherwise. The player bounding box is a proxy for future ball location. We finetuned Imagenet pre-trained VGG-16 network for this task.

As we only have 232 training examples, this vanilla system unsurprisingly did not perform well and overfit very easily even with standard data augmentation techniques such as image cropping and dropout regularization. One of

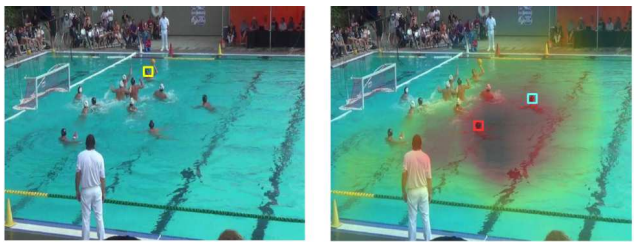

Figure 4: The FCN method (section 5.1.2) takes the left image as input and predicts a heatmap (shown overlaid on right) encoding probable ball locations after the ball is passed. The yellow, cyan and red squares indicate the player with the ball, the ground truth player who actually receive the ball next, and the player predicted to receive the ball by the FCN method respectively.

		Non-expert Humans	
		Correct	Incorrect
Random Forest	Correct	32.8	15.8
	Incorrect	22.8	28.6

Table 2: Comparing agreement between the predictions of next ball possessor made by humans and our best algorithm on the water polo data. Humans and the algorithm both make correct and incorrect predictions on the same examples more often than not.

our contributions is in showing that the performance can be significantly improved (from 10% to 38.1%) by requiring the FCN system to output the location of players in addition to which player will possess the ball next. Our hypothesis about why this modifications helps is that forcing the CNN to predict player locations results in representations that capture the important feature of player configurations and are thus more likely to generalize than other nuisance factors that the CNN can latch onto given the small size of the training set. This finding is interesting because it suggests that it might be possible to learn even better features by forecasting future player locations for which no additional annotation data is required once the detection and tracking pipeline described in the previous sections is setup.

To estimate $P(p^k = l_b|I_t)$ we first detect all the players in image I_t using the method described in section 4. We assume that players will be at the same location after the pass is made. In order to make the ball assignment among players to be mutually exclusive, we use the player locations to perform a Voronoi decomposition of the playing field. Let c^k be the voronoi cell corresponding to the k^{th} player. $P(p^k = l_b)$ is then to set to $\frac{1}{|c^k|}$ if $l_b \in c^k$ and zero otherwise. We then use equation (2) to compute $P(b = i|I_t)$.

This method performs comparably to the baseline that predicts the most open teammate. Visualization in Figure 4 shows a dominant pattern with FCN predictions: it consistently places higher likelihood around the perimeter of team in possession of the ball. This is a very sensible strategy to learn because players around the perimeter are often more open and statistical analysis reveals that there are more passes between perimeter players. Given the limited amount of data, the FCN based approach is unable to capture more nuanced aspects of player configurations or more fine grained visual cues such as the player pose.

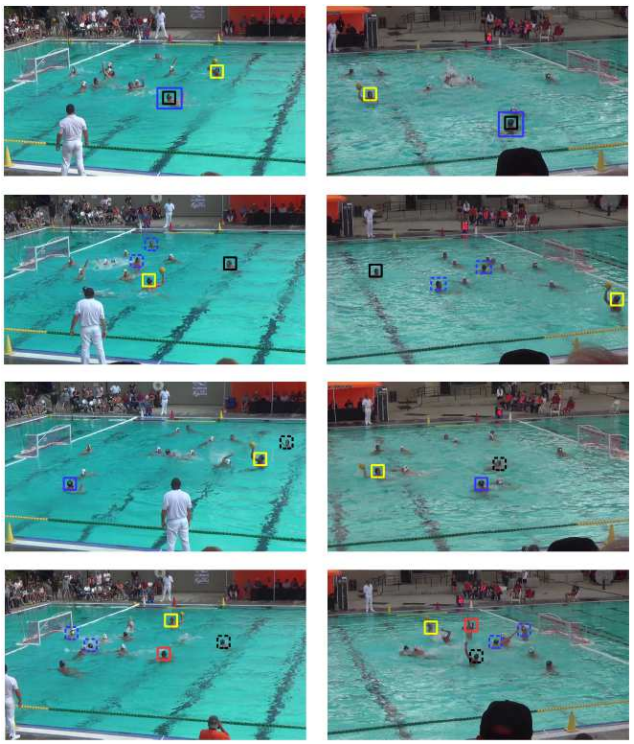

Figure 5: Sample predictions of our algorithm (black) and of water polo laymen (blue). The player in possession of the ball is marked in yellow, and in cases where both our algorithm and the humans made incorrect predictions, the player who actually received the ball is marked in red. A solid line indicates a correct prediction, whereas a dashed line indicates an incorrect prediction. Row 1 shows examples where both made the correct prediction. Row 2 shows examples where the algorithm is correct, but humans are incorrect. Row 3 shows examples where humans are correct, but our algorithm is incorrect. Finally, row 4 shows examples where both our algorithm and humans were incorrect.

5.1.3 Comparison to Human Performance

Figure 5 compares the predictions of human non-experts against our best performing system. Some common trends are: Non-experts are more likely to incorrectly predict players near the cage. Table 2 reports agreement statistics between the predictions of our systems and non-expert humans. These numbers suggest that humans and our system

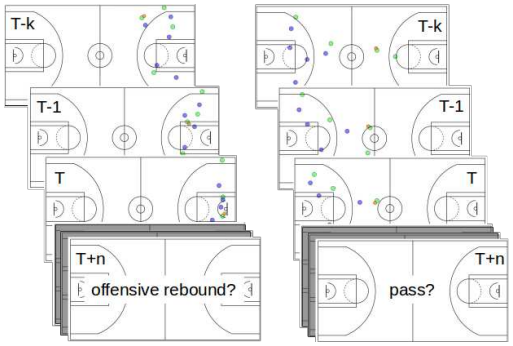

Figure 6: Examples of the basketball event prediction task: forecast an event n seconds in the future, provided a k-second history of the player and ball trajectories.

have similar biases and are accurate/prone to errors on similar examples.

5.2. Basketball: Where will the ball go?

As more data was available for basketball, we attacked the more general problem of predicting where the ball will go next after one and two second respectively. We represented the overhead view as 64x64x3 images where the three channels corresponded to location of players of team 1, players of team 2 and the ball respectively. For capturing temporal context, we included 5 images from the past taken at times $\{t, t-1, ...t-4\}s$ respectively. The task was to predict the ball location at times $\{t+1, t+2\}s$ respectively. To account for multimodality in the output, we formulate this as a classification problem with the xy plane discretized into 256 bins.

We experiment with two different CNN training strategies: (a) early fusion of the temporal information by concatenating 5 images into a 15 channel image that was fed into a CNN or, (b) late fusion by using a LSTM on the output of CNN feature representation of the 5 images. The CNN architecture comprised of 4 convolutional layers containing 32 filters each of size 3x3, stride 2 and ReLU nonlinearity. In case of early fusion, the output of the last convolutional layer was fed into a 512-D fully connected layer which in turn fed into the prediction layer. In case of late fusion, the output of the last convolutional layer was fed into a 512-D LSTM layer which in turn fed into a prediction layer. The performance of these networks and some baseline methods is reported in Table 3.

We consider two baselines - one which predicts that the ball at time $t+1, t+2$ will remain at the same location as at time t (i.e. Last position). This is a reasonable baseline because in many frames the player is in possession of the ball and he does not move. The second baseline estimates the ball velocity at time t and uses it to forecast the future location. We report mean and median errors in the distance and the angle of prediction. The distance between the ground truth and predicted location is reported as the percentage of the length of the basketball court. The angular error is the angle between the vector 1 pointing from current position to ground truth position in the future and vector 2 pointing from current to predicted position. We find that the proposed methods outperform the baseline and the early fusion method performs slightly better than the late fusion method. As expected, the prediction errors in distance are larger when predicting for 2s as compared to 1s. However, the errors in angle follow the reverse trend. One explanation is that in a shorter period, the ball moves by small distances and therefore angle measures are not robust.

Method	Error (1s in Future)				Error (2s in Future)			
	Distance (%)		Angle (^{o})		Distance (%)		Angle (^{o})	
	Mean	Median	Mean	Median	Mean	Median	Mean	Median
Last Position	11.7	10.4	-	-	20.0	18.3	-	-
Ball Velocity	100	100	89.3	88.7	100	100	88.8	85.9
CNN + LSTM	11.4	8.6	61.8	46.6	17.1	14.1	**53.1**	**38.1**
CNN (Early Fusion)	**10.8**	**8.3**	**60.2**	**44.1**	**16.8**	**13.8**	54.3	38.3

Table 3: The early fusion CNN outperforms Last Position and Ball Velocity baseline methods and a late fusion CNN based approach in predicting (basket)ball position 1s and 2s in the future. We report mean and median errors in the distance and angle of predicted ball positions.

5.3. Transferring from Basketball to Water polo

Basketball and water polo are both team sports that require scoring baskets/goals. This suggests that there maybe general game play strategies, e.g., pass to the most open player, that are shared between these two games. If this is indeed the case then a model trained on one of these sports should perform reasonably well on forecasting events in the other sport. In order to test this hypothesis we trained a random forest model on the basketball data (the larger dataset) for predicting which player will get the ball next using the same features as described in 5.1.1 and then tested it on the water polo testing set.

The accuracy of this model on basketball itself was 69.9% and 36.8% on water polo. The performance on water polo is worse than a model trained directly on water polo (which achieves 45.5%) but same as the most open teammate baseline with 36.7% accuracy (Table 1). One explanation of these results is that differences in game strategies arise from the differences in game rules, number of players, and field size. Therefore the basketball model is outperformed by a model trained on water polo itself. However, the transfer performance is significantly better than chance performance and nearest teammate baseline, suggesting that our method is capable of learning game-independent ball passing strategies. A more detailed analysis of the error modes is provided in the supplementary materials.

Method	Dataset	Δ T	FT made	FT miss	FG made	FG miss	Off. Rebound	Def. Rebound	Turnover	Foul	Time Out	Dribble	Pass	Possession	Block	Assist	Drive	Screen	mAP
Avg. Human	H	1s	100.0	0	60.0	12.5	11.1	16.7	0	0	33.3	66.7	0	0	0	0	0	28.6	20.6
Random Forest	H	1s	100.0	0	20.0	0	20.0	40.0	0	0	0	100.0	20.0	40.0	0	0	0	0	21.3
Image CNN	A	1s	46.0	20.3	3.9	4.8	2.0	5.5	0.9	1.6	0	61.7	16.5	22.9	0.9	1.8	1.6	3.7	11.9
Overhead CNN	A	1s	62.2	22.7	38.6	16.4	9.4	43.9	1.8	5.2	3.5	76.1	25.5	37.6	0.8	3.4	1.6	13.1	22.6
Random Forest	A	1s	75.5	41.4	41.3	15.7	11.8	61.2	2.3	5.6	4.5	80.5	26.7	40.9	1.0	3.5	1.2	8.5	26.4
Avg. Human	H	2s	33.3	20.0	14.3	0	0	0	0	0	37.5	75.0	0	0	0	0	16.7	20.0	13.5
Random Forest	H	2s	100.0	0	0	0	20.0	40.0	0	0	0	100.0	0	20.0	0	0	0	0	17.5
Image CNN	A	2s	32.5	7.8	1.9	2.5	0.9	2.7	0.5	0.8	0.2	53.8	14.7	19.9	0	0.6	0.6	2.9	8.8
Overhead CNN	A	2s	39.8	19.0	7.3	6.9	3.8	12.9	1.5	2.2	1.6	71.0	18.3	25.3	0.4	2.7	1.1	5.8	13.7
Random Forest	A	2s	66.9	29.7	11.8	7.3	5.0	35.4	1.5	2.6	2.7	76.4	21.4	30.2	0.3	2.5	0.9	5.0	18.7
Avg. Human	H	40ms	28.6	28.6	83.3	0	50.0	0	0	0	0	25.0	57.1	14.3	0	0	20.0	83.3	24.4
Random Forest	H	40ms	100.0	0	40.0	80.0	40.0	100.0	0	20.0	0	100.0	60.0	100.0	0	0	0	80.0	45.0
Random Forest	A	40ms	68.8	24.5	69.5	54.7	62.7	85.2	6.1	31.8	16.7	93.2	76.2	92.6	3.3	8.1	5.0	57.7	47.3

Table 4: Prediction accuracy ΔT seconds in the future of 16 basketball events: free throw (FT) made and missed, field goal (FG) made and missed, offensive (off) and defensive (def) rebound, etc. Methods were evaluated on the full (A) *test* split of $90k$ events, as well as a smaller, 80-example subset (H) for human performance evaluation and comparison.

6. Forecasting Events in Basketball

Predicting the ball location is just one out of many events of interest. For example, whether a teammate would screen or whether dribble or a break would take place are of great interest in basketball. In a manner similar to predicting where the ball will be at times $\{t+1, t+2\}s$, we predict which out 16 events of interest will happen in the future.

We evaluate random forest and neural network based approaches for this task. The input to the random forest are the following hand designed features, extracted from the last visible frame: player and ball coordinates and velocities, distances between each player and the ball, angles between each player and the ball, the time remaining on the shot clock, the remaining game time in the period, and the time since the most recent event for each event occurring in the visible history. In total, we used 92 features. We tested two different neural networks - (a) Overhead CNN that took as inputs the image representation of the overhead view (see Section 5.2) along with the hand designed features described above and (b) Image CNN that took as input raw RGB images. The neural network architectures and training procedure are detailed in the supplementary materials.

Table 4 reports the performance of humans and various methods described above at predicting player moves 1s, 2s and 40ms in advance. The two test splits,"*H*" and "*A*" correspond to 80 examples on which human subjects were tested and a set 90K examples on which the algorithm was evaluated. The purpose of reporting the accuracy when predicting 40ms in advance is to obtain an upper bound on performance. The results reveal that random forest outperforms CNN based approaches and both these approaches perform better than an average human. The Overhead CNN outperforms the Image CNN suggesting that extracting features relevant for forecasting from raw visuals is a hard problem. It is also noteworthy that humans are significantly better at identifying Field Goals (i.e. FG made), but worse at identifying other events.

7. Conclusion

In this work we present predicting next players' moves in basketball and water polo as benchmark tasks for measuring performance of forecasting algorithms. Instead of forecasting activities of a single human, sports require forecasting in adversarial multi-agent environments that are a better reflection of the real world. As the events we predict are discrete, our benchmark allows for a crisp and meaningful evaluation metric that is critical for measuring progress. We compare the performance of two general systems for forecasting player moves: 1) a hand-engineered system that takes raw visuals as inputs, then transforms them into an overhead view for feature extraction, and 2) an end-to-end neural network system. We find the hand-engineered system is close to (non-expert) human performance in water polo and outperforms humans in basketball. In both cases it outperforms the neural network system, which raises a very interesting question - what auxiliary tasks/unsupervised feature learning mechanisms can be used to improve prediction performance. We find that a system trained on basketball data generalizes to water polo data, showing that our techniques are capable of extracting generic game strategies.

8. Acknowledgements

We thank James Graham, of the University of Pacific, for providing the water polo game film. We thank Saurabh Gupta and Shubham Tulsiani for helpful discussions. This research was supported, in part, by Berkeley Deep Drive sponsors, and ONR MURI N00014-14-1-0671.

References

[1] P. Agrawal, A. Nair, P. Abbeel, J. Malik, and S. Levine. Learning to poke by poking: Experiential learning of intuitive physics. *arXiv preprint arXiv:1606.07419*, 2016. 1

[2] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese. Social lstm: Human trajectory prediction in crowded spaces. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 961–971, 2016. 1

[3] M. Beetz, N. von Hoyningen-Huene, B. Kirchlechner, S. Gedikli, F. Siles, M. Durus, and M. Lames. Aspogamo: Automated sports game analysis models. *International Journal of Computer Science in Sport*, 8(1):1–21, 2009. 2

[4] A. Bialkowski, P. Lucey, P. Carr, Y. Yue, S. Sridharan, and I. Matthews. Large-scale analysis of soccer matches using spatiotemporal tracking data. In *Data Mining (ICDM), 2014 IEEE International Conference on*, pages 725–730. IEEE, 2014. 2

[5] P. V. K. Borges, N. Conci, and A. Cavallaro. Video-based human behavior understanding: a survey. *Circuits and Systems for Video Technology, IEEE Transactions on*, 23(11):1993–2008, 2013. 2

[6] M. Brand, N. Oliver, and A. Pentland. Coupled hidden markov models for complex action recognition. In *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*, pages 994–999. IEEE, 1997. 2

[7] D. Fouhey and C. Zitnick. Predicting object dynamics in scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2019–2026, 2014. 2

[8] K. Fragkiadaki, P. Agrawal, S. Levine, and J. Malik. Learning visual predictive models of physics for playing billiards. *arXiv preprint arXiv:1511.07404*, 2015. 1, 2

[9] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004. 4

[10] D.-A. Huang and K. M. Kitani. Action-reaction: Forecasting the dynamics of human interaction. In *Computer Vision–ECCV 2014*, pages 489–504. Springer, 2014. 2

[11] IEEE. *Tracking multiple people under global appearance constraints*, 2011. 2

[12] A. Jain, H. S. Koppula, B. Raghavan, S. Soh, and A. Saxena. Car that knows before you do: Anticipating maneuvers via learning temporal driving models. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3182–3190, 2015. 1

[13] V. Karasev, A. Ayvaci, B. Heisele, and S. Soatto. Intent-aware long-term prediction of pedestrian motion. 2

[14] K. Kitani, B. Ziebart, J. Bagnell, and M. Hebert. Activity forecasting. *Computer Vision–ECCV 2012*, pages 201–214, 2012. 1, 2

[15] J. F. P. Kooij, N. Schneider, F. Flohr, and D. M. Gavrila. Context-based pedestrian path prediction. In *Computer Vision–ECCV 2014*, pages 618–633. Springer, 2014. 2

[16] H. S. Koppula and A. Saxena. Anticipating human activities using object affordances for reactive robotic response. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 38(1):14–29, 2016. 1, 2

[17] H. Kretzschmar, M. Kuderer, and W. Burgard. Learning to predict trajectories of cooperatively navigating agents. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 4015–4020. IEEE, 2014. 2

[18] H. W. Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2:83–97, 1955. 4

[19] T. Lan, T.-C. Chen, and S. Savarese. A hierarchical representation for future action prediction. In *Computer Vision–ECCV 2014*, pages 689–704. Springer, 2014. 2

[20] I. Laptev, M. Marszałek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008. 2

[21] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015. 5

[22] P. Lucey, A. Bialkowski, P. Carr, S. Morgan, I. Matthews, and Y. Sheikh. Representing and discovering adversarial team behaviors using player roles. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2706–2713, 2013. 2

[23] A. Maksai, X. Wang, and P. Fua. What players do with the ball: A physically constrained interaction modeling. *arXiv preprint arXiv:1511.06181*, 2015. 2

[24] M. Mathieu, C. Couprie, and Y. LeCun. Deep multi-scale video prediction beyond mean square error. *arXiv preprint arXiv:1511.05440*, 2015. 1, 2

[25] R. Mottaghi, H. Bagherinezhad, M. Rastegari, and A. Farhadi. Newtonian image understanding: Unfolding the dynamics of objects in static images. *arXiv preprint arXiv:1511.04048*, 2015. 2

[26] J. Oh, X. Guo, H. Lee, R. Lewis, and S. Singh. Action-conditional video prediction using deep networks in atari games. *NIPS*, 2015. 1

[27] V. Ramanathan, J. Huang, S. Abu-El-Haija, A. Gorban, K. Murphy, and L. Fei-Fei. Detecting events and key actors in multi-person videos. *arXiv preprint arXiv:1511.02917*, 2015. 2

[28] M. Ranzato, A. Szlam, J. Bruna, M. Mathieu, R. Collobert, and S. Chopra. Video (language) modeling: a baseline for generative models of natural videos. *arXiv preprint arXiv:1412.6604*, 2014. 1, 2

[29] E. Rehder and H. Kloeden. Goal-directed pedestrian prediction. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 50–58, 2015. 2

[30] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in Neural Information Processing Systems*, pages 568–576, 2014. 2

[31] K. Soomro, A. R. Zamir, and M. Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012. 2

[32] STATS. https://www.stats.com/sportvu-basketball/. 3

[33] R. Urtasun, D. J. Fleet, and P. Fua. 3d people tracking with gaussian process dynamical models. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 1, pages 238–245. IEEE, 2006. 2

[34] C. Vondrick, D. Patterson, and D. Ramanan. Efficiently scaling up crowdsourced video annotation. *International Journal of Computer Vision*, pages 1–21. 10.1007/s11263-012-0564-1. 3

[35] C. Vondrick, H. Pirsiavash, and A. Torralba. Anticipating the future by watching unlabeled video. *arXiv preprint arXiv:1504.08023*, 2015. 1, 2

[36] C. Vondrick, H. Pirsiavash, and A. Torralba. Generating videos with scene dynamics. In *Advances In Neural Information Processing Systems*, pages 613–621, 2016. 1

[37] J. Walker, C. Doersch, A. Gupta, and M. Hebert. An uncertain future: Forecasting from static images using variational autoencoders. In *European Conference on Computer Vision*, pages 835–851. Springer, 2016. 1

[38] J. Walker, A. Gupta, and M. Hebert. Patch to the future: Unsupervised visual prediction. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 3302–3309. IEEE, 2014. 1, 2

[39] X. Wei, P. Lucey, S. Morgan, and S. Sridharan. Predicting shot locations in tennis using spatiotemporal data. In *Digital Image Computing: Techniques and Applications (DICTA), 2013 International Conference on*, pages 1–8. IEEE, 2013. 2

[40] X. Wei, P. Lucey, S. Vidas, S. Morgan, and S. Sridharan. Forecasting events using an augmented hidden conditional random field. In *Computer Vision–ACCV 2014*, pages 569–582. Springer, 2014. 2

[41] S.-K. Weng, C.-M. Kuo, and S.-K. Tu. Video object tracking using adaptive kalman filter. *Journal of Visual Communication and Image Representation*, 17(6):1190–1208, 2006. 2

[42] S. Yeung, O. Russakovsky, N. Jin, M. Andriluka, G. Mori, and L. Fei-Fei. Every moment counts: Dense detailed labeling of actions in complex videos. *arXiv preprint arXiv:1507.05738*, 2015. 2

[43] A. Yilmaz, O. Javed, and M. Shah. Object tracking: A survey. *Acm computing surveys (CSUR)*, 38(4):13, 2006. 2

[44] T. Zhou, S. Tulsiani, W. Sun, J. Malik, and A. A. Efros. View synthesis by appearance flow. In *European Conference on Computer Vision*, pages 286–301. Springer, 2016. 1

[45] Y. Zhou and T. L. Berg. Temporal perception and prediction in ego-centric video. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4498–4506, 2015. 2