What makes Federer look so elegant?

Kuldeep Kulkarni and Vinay Venkataraman

Abstract—Everyday we come across thousands of sportsmen in action. Each of them have their own style of play and make varied impressions on the viewers. Despite this expected variability in the impression they make, there are certain inherent qualities of their play like their poise, the economy of their movement, flow of their movement etc. which make some of them more watchable to most of us. For example in tennis, Federer is widely regarded as the one of the most elegant players in the history. In cricket, the upright stance, the perfect balance, the precision in shot-making are some of the qualities which makes Sachin Tendulkar look more 'elegant'or 'better' than others. In this project, we wish to the measure the 'watchability' of a player by quantifying the quality of the various movements of a player. In particular, we concentrate mainly on the movements of a batsman in cricket and provide principled ways to measure the 'watchability' of a batsman in terms of the three typical movements of a batsman, viz stance, back-lift and follow-through. 'Watchability' scores can be very useful for a qualitative video summarization of sports videos, analyzing the change in style of a single players play over the course of a long career, or even to determine the amount of influence of one players style on another.

Keywords—

1 Introduction

Every day we watch various sportsmen in action on television. Each one of us have our own likings and hence our own favourite sportsmen which we wish to watch over and over again. However, despite this expected variability in our likings, there is usually universal agreement amongst the followers of a certain sport that a certain player is more 'watchable' than others or player A plays similar to player B. For example, Don Bradman, the greatest cricketer the world has seen, saw himself in a modern cricketer, Sachin Tendulkar and acknowledged that he bats much the same way that he used to bat, even though they belonged to totally different generations. Hence in this project, we take a baby step towards using computer vision techniques to automatically extract those qualities which make a certain player look more graceful or ungainly than others, as the case may be and quantify such qualities. In short, we call the set of these qualities as 'watchability' of a batsman/shot. Due to time constraints, we concentrate only on cricket clips, and in particular on the quality of shot the batsman plays.

One of the challenges in assessing the quality of a cricket shot is that there are various kinds of shots a batsman can play. As shown in the figure 1, all are played very differently from each other. For example, a straight drive is played with the full face of the bat facing camera so that the ball is directed back at angle of about 20 degrees to the direction of the delivery of the ball but a sweep shot is played by resting one knee on the ground, and directing the ball at an angle of around 135 degrees to the direction of the delivery of the ball. Hence, it is essential that strategy to measure the 'watchability' of a shot is based on the type of the shot played rather being a universal one. Hence, the naturally the first step in the pipeline is to identify the type of shot that is played in the test clip. Once, the type of shot is recognized, the quality of movement in the clip is scored based on the type of shot in the clip. The 'watchability' of a batsman is determined by how harmoniously the different parts of the body move with respect to each other. Hence, it is essential to understand the dynamics of all the important moving parts of the body, like hands, elbow, head, legs etc. with respect to each other. The ideal way to learn such dynamics is by using joint locations in every frame. However, despite significant progress, extracting joint angle locations from images remains a notoriously hard problem, and most current solutions lead to noisy outputs. To overcome this we use poselets [1] which are body part detectors, and are tightly clustered in both appearance space and configuration space. We obtain poselet activation vector [2] which implicitly encodes the joint locations for each frame, based on which we obtain a feature vector for the clip. This feature vector is used to determine the 'watchability' score of the shot played in the clip.

Related Work:

There has not been much previous work in literature related to movement quality assessment from videos. Since, action recognition is an important step in the pipeline explained in introduction, we briefly describe some of well-known action recognition methods below.

a) Action Recognition: The approaches in human activity recognition can be categorized based on the low level features. Most successful representations of human activity are based on features like optical flow, point trajectories, background subtracted blobs and shape, filter responses, etc. Mori *et al.*[3] and Cheung *et al.*[4] used geometric model based and shape based representations to recognize actions. Bobick and Davis [5] represented actions using 2D motion energy and motion history images from a sequence of human silhouettes. Laptev [6] extracted local interest points from a 3-dimensional spatiotemporal volume, leading to a concise representation of a video. For a detailed survey of action recognition, the readers are referred to [7].

2 Data Collection

As mentioned earlier, our goal of this project was to be able to assess the quality of cricket shots from Youtube videos. Our preliminary goal was to be able to recognize different cricket shots. For these experiments, we have collected a dataset with six action classes (cricket shots) - Straight Drive, Cover Drive, Cut, Flick, Pull and Sweep. Examples of these classes were collected from four batsmen (left handed) so that we have 10 examples for each class. The exemplar videos of each class is shown in Figure 1. The videos collected from Youtube.com were edited such that the start of the video is when bowler is about to bowl (throw) the ball and the end is marked just after the shot is completed. This was done consistently across shots and batsmen.

3 Cricket Shot Recognition

Due to reasons explained in section 1, we believe that to quantify the quality of movement, having an independent computational framework for each type of shot is necessary. Hence, recognizing a cricket shot is a very important step for quantification of quality of cricket shots. The first approach we take is to use bat trajectory for recognizing shots, as it is clear that the trajectory of the bat is a signature of a shot.

3.1 Shot Recognition using Bat

For a batsmen to achieve a cricket shot, the batsmen must position himself in a particular way and the bat should traverse on a particular trajectory. This makes the position of the bat to be unique for a shot at the time of impact with the ball. As our first approach, we propose to use a semi-supervised bat segmentation approach for shot recognition. The difference in position of the bat for two different shots are shown in Figure 2. The video frames are selected when the bat is about to hit the ball. We can clearly see that the shape of the bat for these shots are unique, and our aim is to extract features which are descriptive of these shapes. To get the segments of the video frame, we have used the segmentation algorithm proposed by Liu et al. [8].

3.1.1 Shape Distributions of Bat

From Figure 2, it is clear that the shape of the bat is unique for a cricket shot at the point of impact of bat with the ball. Assuming that the semi-supervised approach of bat segmentation is convincingly working to give bat segments as output for every shot, we extract discriminative shape features of these segments using an approach proposed by Osada et al. [9]. First we extract the boundary of the shape (bat) and then extract *D1, D2* and *D3* shape distributions as mentioned in [9]. These features are pictorially represented in Figure 3. In our experiments, we have used euclidean distance and nearest neighbor classifier with leave-one-out crossvalidation approach to evaluate the performance of the framework. The classification accuracy results of our semi-supervised framework for *D1, D2* and *D3* shape distributions were 63.33, 73.33 and 65 respectively. The confusion table for D2 (best performance) shape distribution is shown in Table 1.

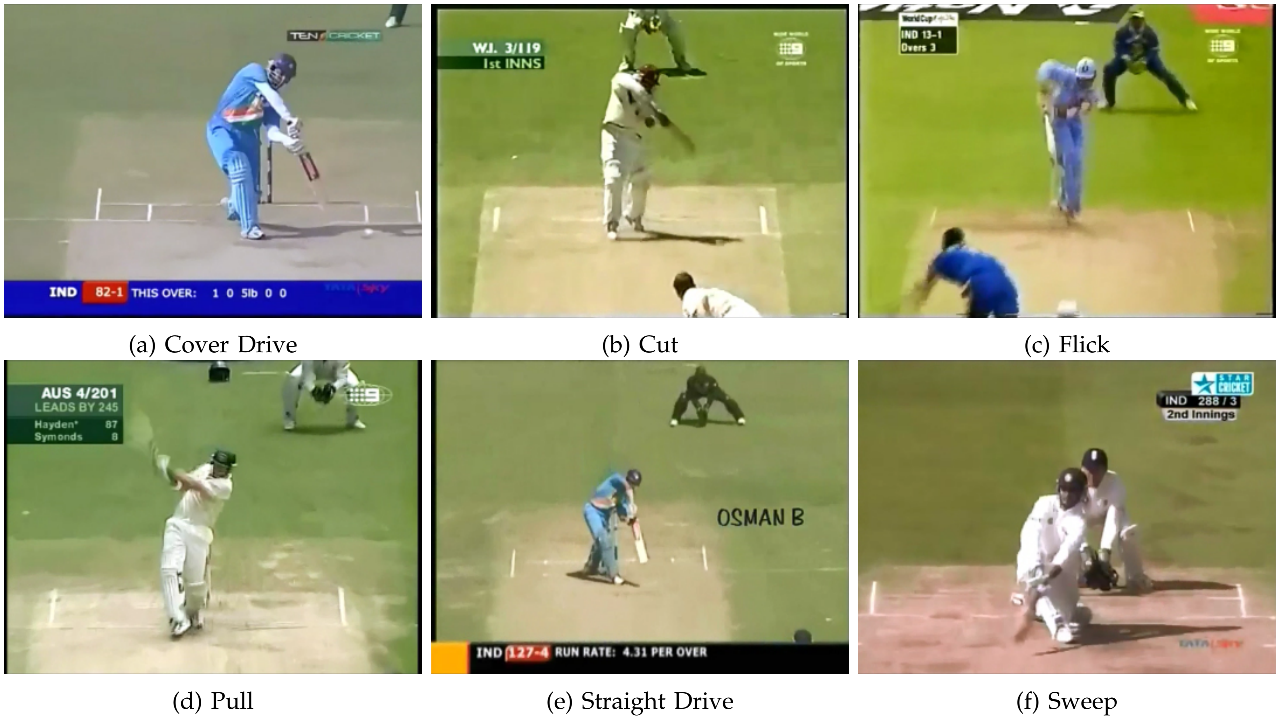
	SD	CD	FL	CT	PL	SW
SD	6	1	1	2	0	0
CD	1	8	0	1	0	0
FL	0	0	9	0	0	1
CT	3	1	0	6	0	0
PL	0	0	0	0	8	2
SW	0	0	1	0	2	7

TABLE 1: Confusion table for shot recognition using D2 shape distribution with euclidean distance as similarity measure and nearest neighbor classifier. Here, the labels SD, CD, FL, CT, PL and SW refers to Straight Drive, Cover Drive, Flick, Cut, Pull and Sweep cricket shot classes respectively.

3.2 Automatic Bat Detection for Shot Recognition

In the previous section, we have seen that shape of the bat at the time of impact is indicative of the shot. But the proposed framework required a person to mark the segment which had the bat. In this section, we propose a framework to automate this process of bat segmentation. The problem here is to identify the segment (outputs of the segmentation algorithm) with a bat. We assume the bat to be a rectangle and fit a rectangle to all segments. The confidence score of a segment to have a bat is then calculated as sum of distances between every point on the boundary of the segment to the nearest side of the rectangle. The segment with lowest score (corresponding to best fit) is selected as the bat segment. This procedure is done for the last 10 frames in the video where the actual cricket shot exist.

After selection of a segment for all video frames, we use shape distributions as a representative feature of the bat and concatenate the shape distributions for all the frames to form our feature vector. Similar to previous section, we use euclidean distance with nearest neighbor classifier to evaluate the performance of this framework. The results are tabulated in Table 2. The classification accuracy for *D1, D2* and *D3* shape distributions were 25, 30 and 28.33 respectively. It should be noted that the problem addressed here is difficult due to various reasons including, (a) the bat is not visible in many video frames, (b) the size and shape of the bat varies as the camera zooms in. The results achieved by this experiment sounds encouraging taking these factors into consideration and the fact that the proposed framework is completely without any user interference.



(a) Cover Drive (b) Cut (c) Flick

(d) Pull (e) Straight Drive (f) Sweep

Fig. 1: The videos of cricket shots collected from Youtube.com of four batsmen (left handed) forming six classes with 10 examples each.

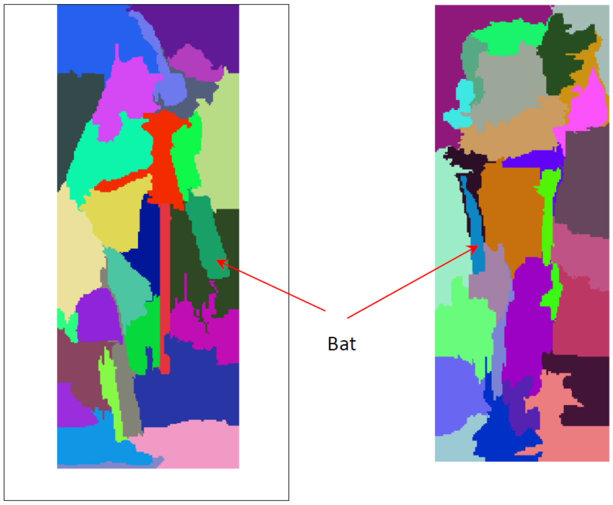


Fig. 2: An example of bat position and shape for two cricket shots (Straight Drive and Flick). The selected frame is when the bat hits the ball. From the segments (marked in various colors), the user selects a segment with bat.

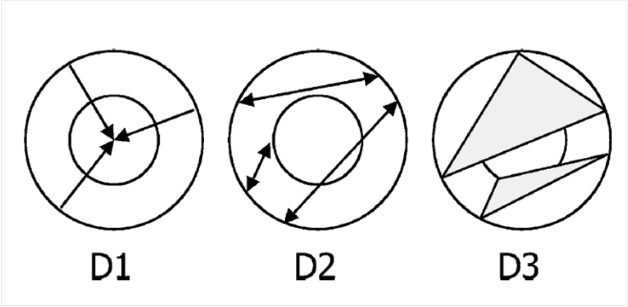


Fig. 3: Three shape distribution measures extracted from the bat segments marked by user. This image was taken from [9].

	CD	CT	FL	PL	ST	SW
CD	7	1	0	0	0	2
CT	3	1	2	0	2	2
FL	1	1	3	2	1	2
PL	2	0	2	2	3	1
ST	1	2	2	1	2	2
SW	4	1	1	1	0	3

TABLE 2: Confusion table for shot recognition using D2 shape distribution with euclidean distance as similarity measure and nearest neighbor classifier. Here, the labels SD, CD, FL, CT, PL and SW refers to Straight Drive, Cover Drive, Flick, Cut, Pull and Sweep cricket shot classes respectively.

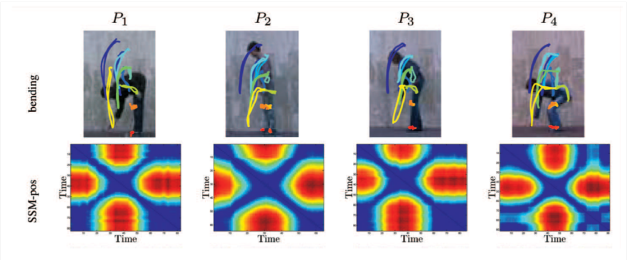


Fig. 4: Self-Similarity Matrix (SSM) extracted from body joints. This image was taken from [10].

	CD	CT	FL	PL	ST	SW
CD	6	3	0	1	0	0
CT	1	4	3	1	1	0
FL	1	1	6	2	0	0
PL	1	1	1	6	1	0
ST	0	1	1	2	6	0
SW	0	0	0	0	0	10

TABLE 3: Confusion table for shot recognition using LBP on SSM-OF feature with euclidean distance as similarity measure and nearest neighbor classifier. Here, the labels SD, CD, FL, CT, PL and SW refers to Straight Drive, Cover Drive, Flick, Cut, Pull and Sweep cricket shot classes respectively.

3.3 Global Features for Shot Recognition

The previous approaches based on bat segmentation for shot recognition were extracting local features. In this section, we propose a framework for shot recognition with global features. We use Self-Similarity Matrix (SSM-OF) [10] using optical flow approach for this purpose. SSM is a graphical way to study the dynamics of a system under consideration. It is based on the theory of recurrence in dynamical system and provide a way to visually analysis of this behavior. It has been previously used for action recognition in video and motion capture data [10]. The SSM matrix is found to possess this strong similarity within an action, which makes it a suitable choice for feature extraction in action recognition experiments. An example is shown in Figure 4.

The optical flow vectors computed on all n pixels were concatenated to form a long feature vector of size $2n$. SSM matrix is then given by the euclidean distance between the concatenated optical flow vectors corresponding to the two frames I_i and I_j. It was seen that these SSMs possessed unique texture patterns for a cricket shot class. Then, we use Local Binary Patterns (LBP) to extract features representative of these textures. To evaluate this framework, we use the same nearest neighbor with euclidean distance as our classifier. The results are tabulated in Table 3. We achieve a classification accuracy of 63.33% on leave-one-out crossvalidation scheme. But it is important to note that the idea of global feature extraction for shot recognition is not sufficient for assessing the quality of cricket shots. This means that local features are preferable for this application.

4 Poselet activation vector

Since, we wish to understand the dynamics of body parts explicitly, we are not interested in generating a feature vector for the entire object of interest, the batsman. Hence, we use poselets [1] which are body part detectors closely clustered both in appearance and configuration space. Based on this, Maji [2] introduced the notion of poselet activation vector (PAV), where given a bounding box, the poselet activation vector is the vector with one entry for each poselet, the entry signifies the amount presence of that particular poselet. Thus for each frame in the test clip, we track the batsman, and with the bounding box thus generated from tracking as input, we obtain a PAV for that frame. For tracking, we use off-the-shelf code released with [11] .

5 Measuring ‘watchability’

Once action is recognized, we want to score the quality of the movement of the batsman. To this end, we divide the movement temporally into the three typical movements of a batsman when he plays a shot, viz stance, back-lift, and follow-through. Stance refers to the movement of the batsman in the first few frames of the test clip. Even though it lasts only for a few (about 2 to 3) frames, the stance contributes to the ’’watchability’ of the batsman significantly. The more upright and side-on the batsman stands, greater is the ‘watchability’ of the batsman. Back-lift refers to the movement of the batsman after the stance but before the ball is delivered by the bowler. This movement often is made so that they can gather momentum which they can impart onto the ball when the shot is played, and typically lasts for about 10 frames. Some batsmen like Brian Lara make exaggerated body movements while some stay very still during back-lift. While a player like Tendulkar who makes just enough movement in his back-lift is considered to be the benchmark. The third movement of the batsman, follow-through refers to the movement of the batsman after the shot is played, and can be considered as the effect of the residual momentum after the shot is played. This movement lasts for about 15 frames. The more controlled and smooth the follow-through is, the greater is the ‘watchability’ of the batsman.

5.1 Feature vector and scoring

For each of the three movements outline above, we generate a sequence of poselet activation vectors, $[\mathbf{PAV}^i_1, ...\mathbf{PAV}^i_{N_i}[$, where $i = 1, 2, 3$ indicate Stance, Back-lift, and Follow-through respectively, and N_i is the number of frames in the i^{th} movement. As stated earlier, the poselet activation vectors implicitly the joint locations of the body. For each poselet, we construct a time series given by $[PAV_1^i(j), ...PAV_{N_i}^i(j)]$ for all $j = 1, 2, .., P$ where P is the number of poselets, and calculate the largest lyapunov exponent [12], L_i^j determining the non-linear dynamics of the time series. Now, the feature vector for the i^{th} movement of the test clip is given by the vector of lyapunov exponents for all poselets, $(f_i = [L_i^1, L_i^2, .., L_i^P]$.

Now, the 'watchability' score for each of the movements is estimated from the feature vector for that movement by using linear regression, as below.

$$v_i = w_i^T f_i \tag{1}$$

where v_i is the score, and w_i is the parameter vector for the i^{th} movement. The parameter vector w_i is estimated by minimizing the mean squared error.

$$\hat{w_i} = X_i^+ v_i \tag{2}$$

where v_i is the vector of the scores of training videos and X_i is the matrix of feature vectors for ith movement for all training videos. Using the guidelines stated earlier in the section, a score between 0 and 1 is given to each of the three movements for all training videos.

6 Experiments

As shown earlier, the action recognition results were not as good as we wished them. Since, measurement of 'watchability' depends on the action/shot recognized, we ideally want perfect action results. Hence, to test the effectiveness of our methodology to measure 'watchability' of the shot, we assume that action/shot is recognized correctly. While training, the more upright and side-on are given higher scores than the front-on stances. Figure 5 shows the true and predicted scores for various stances. Figure 6 shows the true and predicted for some instances of cut shot. For back-lift of cut-shot, the more exaggerated the movement of the batsman is, the lesser is the true score. The stiller batsman stays before the ball is delivered, the higher is the score. Figure 7 shows the true and predicted scores for some instances of follow-throughs of pull shot. Figure 8 shows the true and predicted scores for some instances of follow-throughs of cover drive. The higher the right elbow is, the higher is the true score, as the general consensus among the followers of the game is that the elbow needs to be high as possible for the shot to look good. From figures 5, 6, 7 and 8, it can be seen that predicted scores do not always match the true scores. However, it is to born in mind that the training set we have is very small, and also poselets we used are not tuned to the cricket dataset we have. For example, to predict score of follow-through of cover drive, we need the poselet corresponding to the high elbow to fire in most of the frames of follow-through. However, there is no such poselet in the database we contains high elbow. With careful construction of poselets, and more training dataset, we hope to attain greater accuracy in predicted scores.

7 Contributions

We both together worked on the ideas to set-up the introduction and problem statement. Kuldeep worked on poselet activation vectors and measuring 'watchability' sections and Vinay worked on data collection and action recognition.

References

[1] L. Bourdev and J. Malik, "Poselets: Body part detectors trained using 3d human pose annotations," in *Computer Vision, 2009 IEEE 12th International Conference on*, pp. 1365–1372, IEEE, 2009.

[2] S. Maji, L. Bourdev, and J. Malik, "Action recognition from a distributed representation of pose and appearance," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pp. 3177–3184, IEEE, 2011.

[3] G. Mori, X. Ren, A. A. Efros, and J. Malik, "Recovering human body configurations: Combining segmentation and recognition," in *IEEE Conf. Comp. Vision and Pattern Recog*, 2004.

[4] G. K. M. Cheung, S. Baker, and T. Kanade, "Shape-from-silhouette of articulated objects and its use for human body kinematics estimation and motion capture," in *IEEE Conf. Comp. Vision and Pattern Recog*, 2003.

[5] A. F. Bobick and J. W. Davis, "The recognition of human movement using temporal templates," *IEEE Trans. Pattern Anal. Mach. Intell.*, no. 3, pp. 257–267, 2001.

[6] I. Laptev, "On space-time interest points," *Intl. J. Comp. Vision*, vol. 64, 2005.

[7] J. Aggarwal and M. Ryoo, "Human activity analysis: A review," *ACM Comput. Surv.*, no. 3, April 2011.

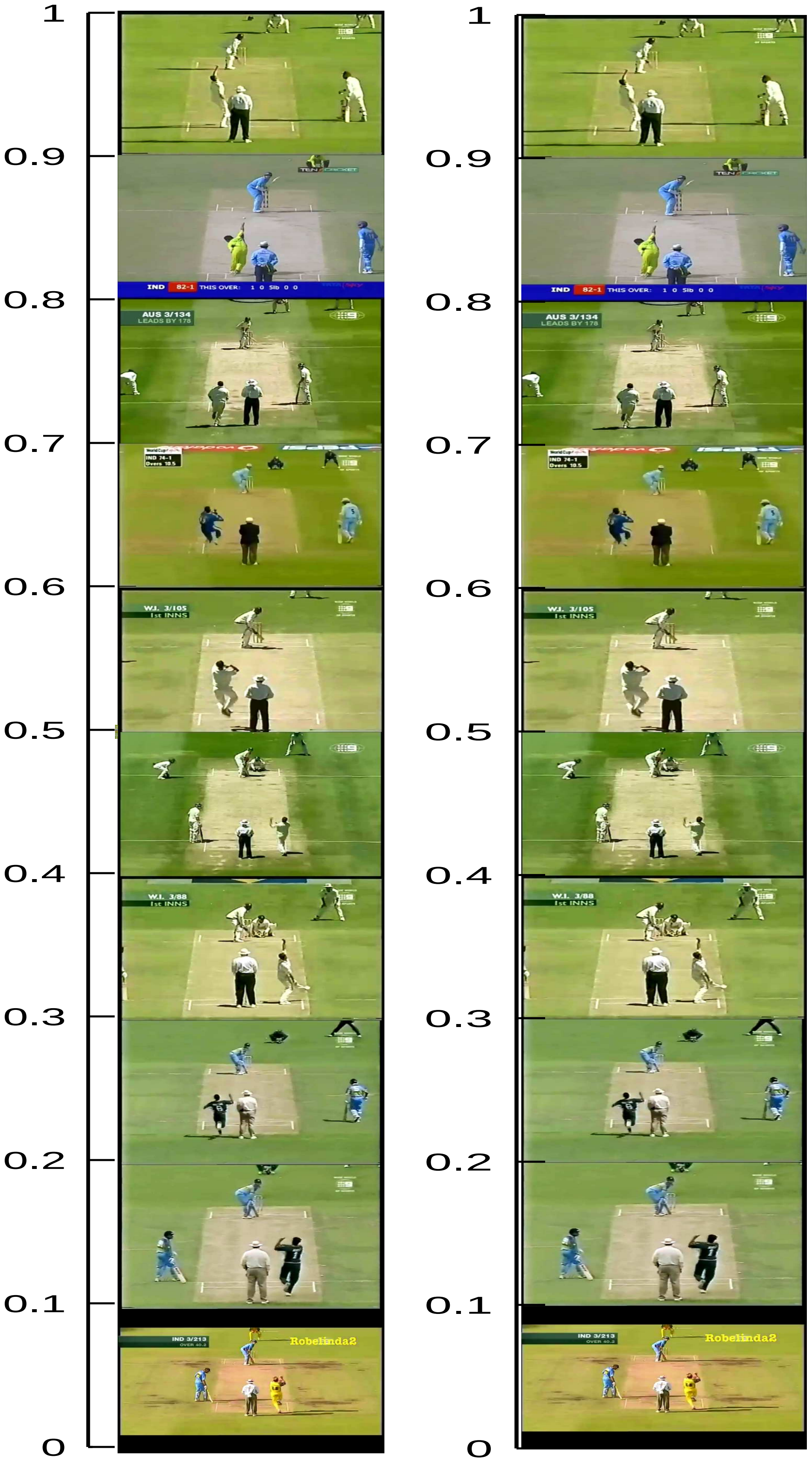
[8] M.-Y. Liu, O. Tuzel, S. Ramalingam, and R. Chellappa, "Entropy rate superpixel segmentation," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pp. 2097–2104, IEEE, 2011.

[9] R. Osada, T. Funkhouser, B. Chazelle, and D. Dobkin, "Shape distributions," *ACM Transactions on Graphics (TOG)*, vol. 21, no. 4, pp. 807–832, 2002.

[10] I. N. Junejo, E. Dexter, I. Laptev, and P. Pérez, "View-independent action recognition from temporal self-similarities," *IEEE Transactions on Pattern Analysis and Machine Intelligence,*, vol. 33, no. 1, pp. 172–185, 2011.

[11] K. Zhang, L. Zhang, and M.-H. Yang, "Real-time compressive tracking," in *Computer Vision–ECCV 2012*, pp. 864–877, Springer, 2012.

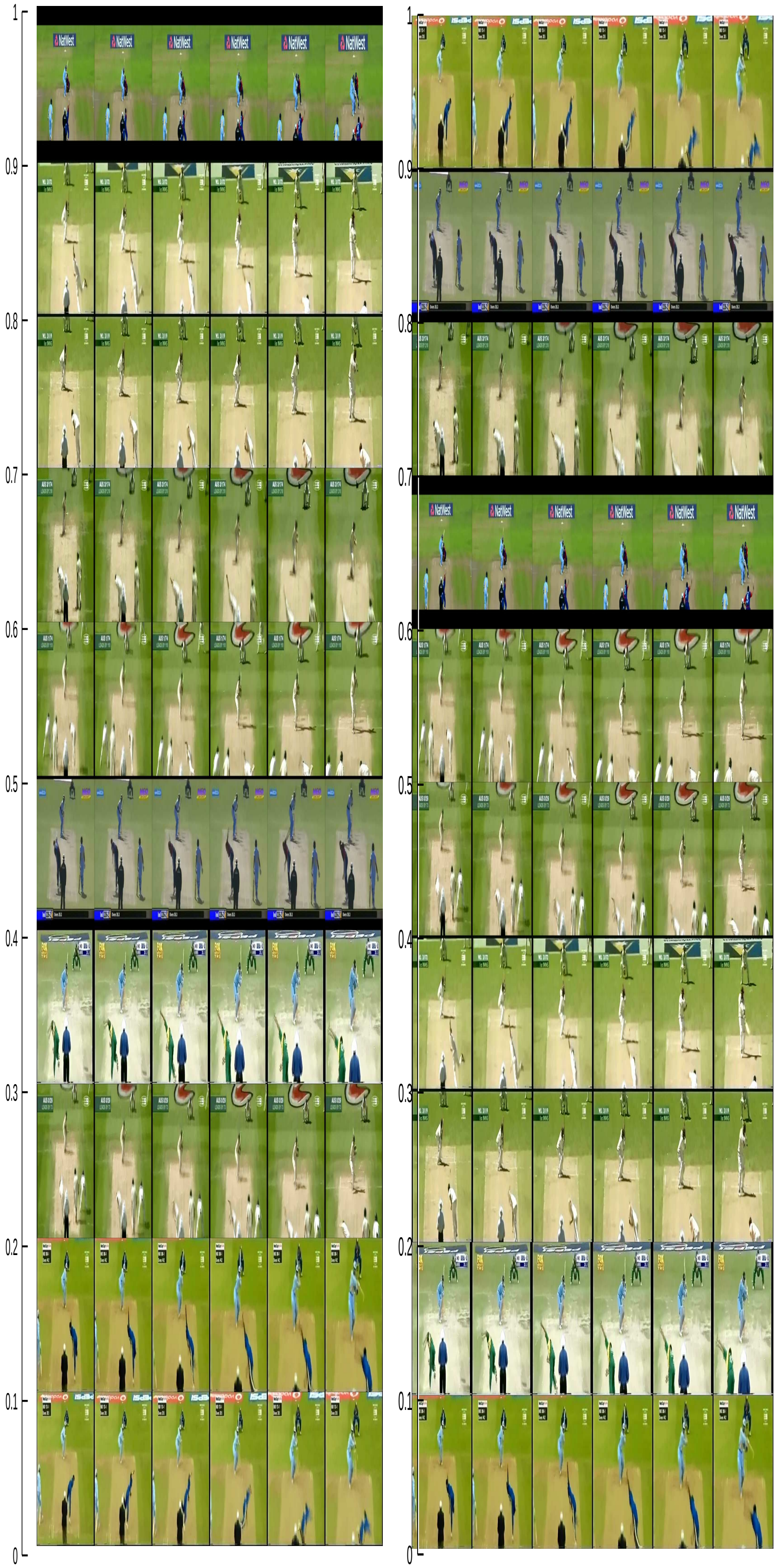
[12] M. T. Rosenstein, J. J. Collins, and C. J. De Luca, "A practical method for calculating largest lyapunov exponents from small data sets," *Physica D: Nonlinear Phenomena*, vol. 65, no. 1, pp. 117–134, 1993.



(a) True Stance scores

(b) Predicted Stance scores

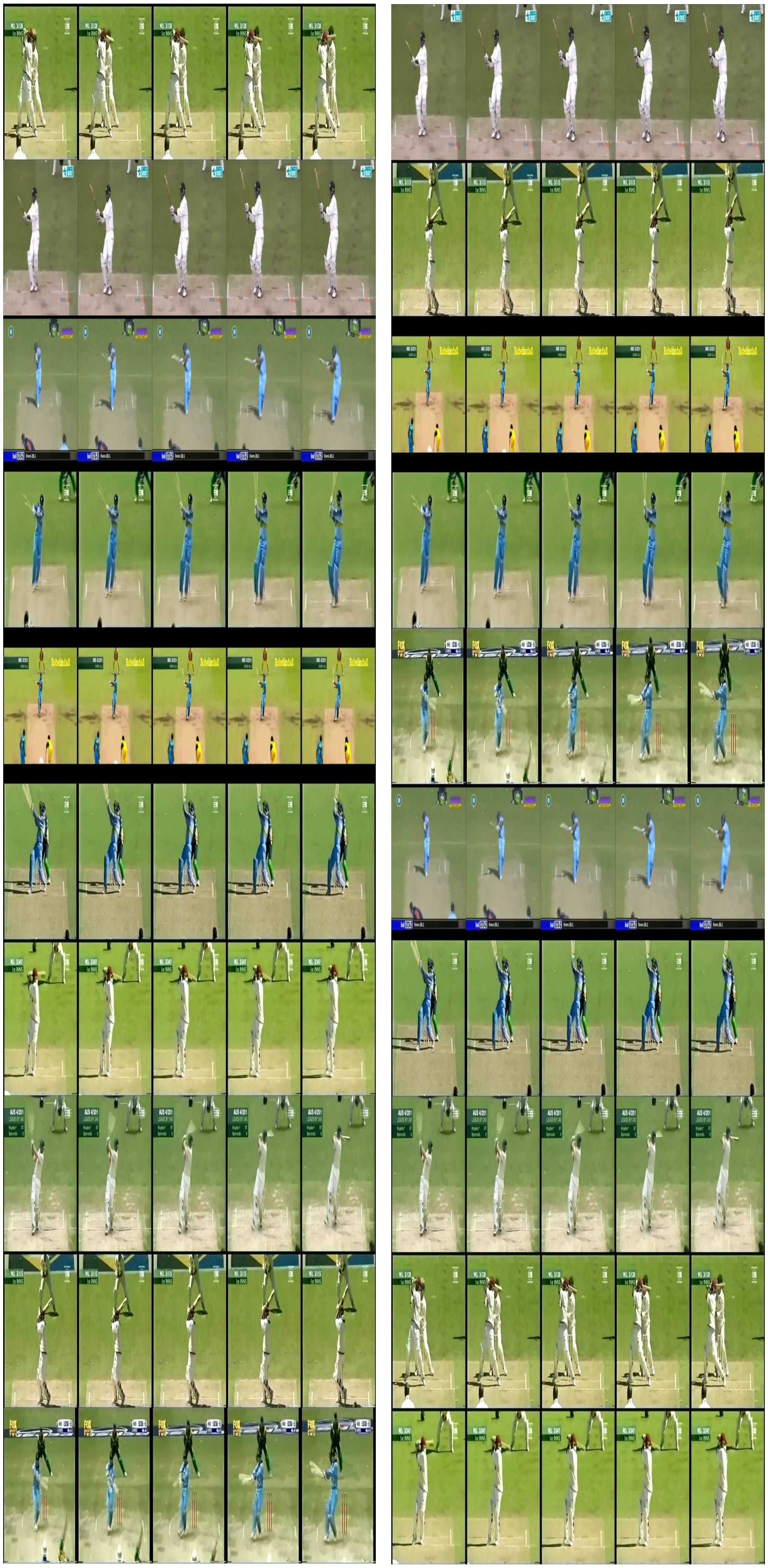
Fig. 5: True and Predicted 'watchability' scores of stances of various batsmen. The more upright and side-on the stance is, the greater is the score.



(a) True Stance scores

(b) Predicted Stance scores

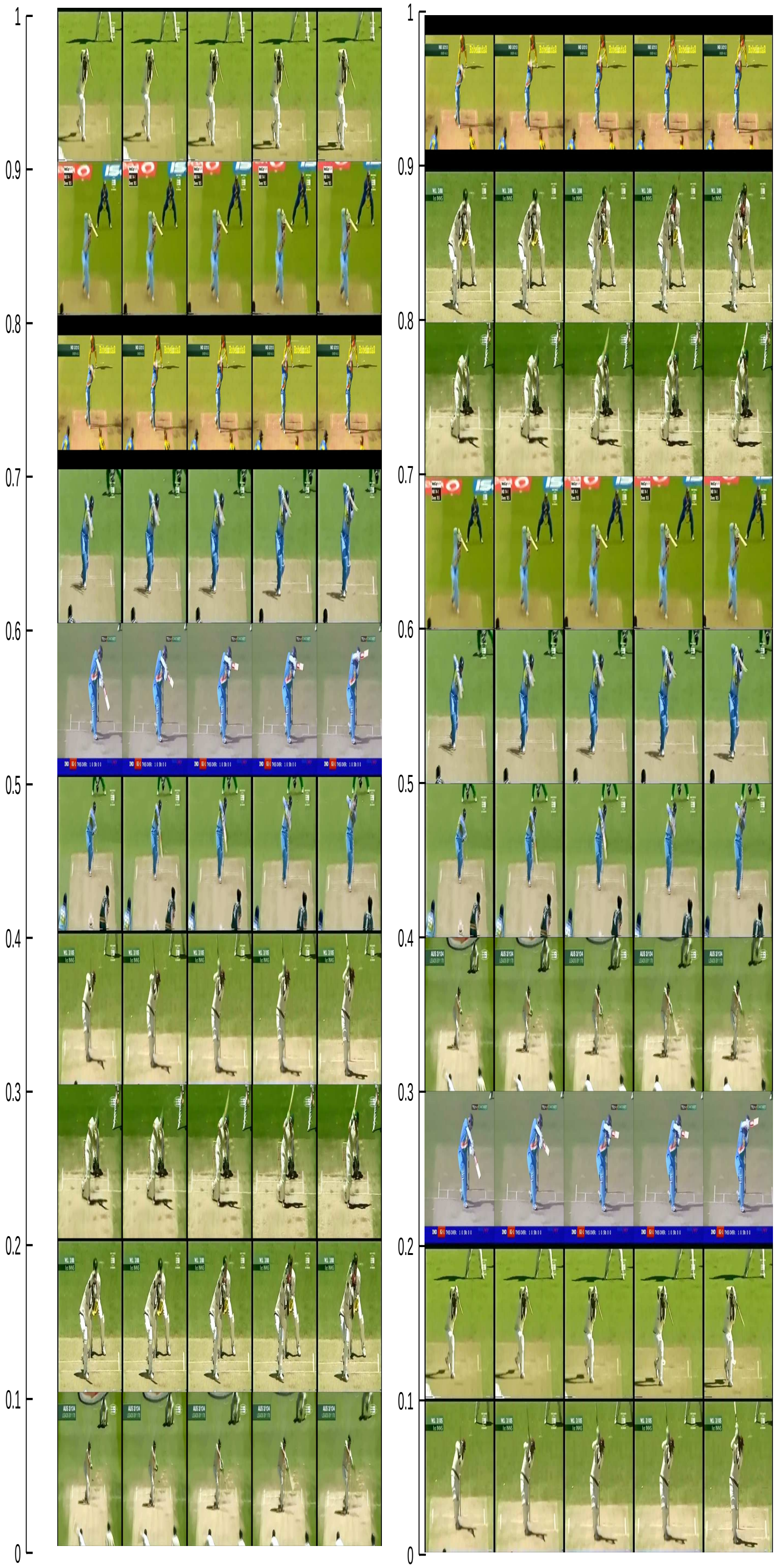
Fig. 6: True and Predicted 'watchability' scores of back-lifts for cut shot.



(a) True Stance scores

(b) Predicted Stance scores

Fig. 7: True and Predicted‘watchability’ scores of follow-throughs for pull shot.



(a) True Stance scores

(b) Predicted Stance scores

Fig. 8: True and Predicted 'watchability' scores of follow-throughs for cover drive.