A Decision Forest Based Feature Selection Framework for Action Recognition from RGB-Depth Cameras *

Farhood Negin[1], Fırat Özdemir[1], Ceyhun Burak Akgül[2,3], Kamer Ali Yüksel[1], and Aytül Erçil[1,2]

[1] Sabancı University, Istanbul, Turkey,
{fnegin,firatozdemir,kamer,aytulercil}@sabanciuniv.edu
[2] Vistek ISRA Vision, Istanbul, Turkey
cb.akgul@gmail.edu,,
[3] Boğaziçi University, Istanbul, Turkey

Abstract. In this paper, we present an action recognition framework leveraging data mining capabilities of random decision forests trained on kinematic features. We describe human motion via a rich collection of kinematic feature time-series computed from the skeletal representation of the body in motion. We discriminatively optimize a random decision forest model over this collection to identify the most effective subset of features, localized both in time and space. Later, we train a support vector machine classifier on the selected features. This approach improves upon the baseline performance obtained using the whole feature set with a significantly less number of features (one tenth of the original). On MSRC-12 dataset (12 classes), our method achieves 94% accuracy. On the WorkoutSU-10 dataset, collected by our group (10 physical exercise classes), the accuracy is 98%. The approach can also be used to provide insights on the spatiotemporal dynamics of human actions.

Keywords: human motion analysis, action recognition, random decision forest

1 Introduction

The proliferation of new depth sensing technologies in recent years has positively changed the climate of the automated vision-based human action recognition problem, deemed to be very difficult due to the various ambiguities inherent to conventional video. Depth sensors, such as Microsoft Kinect [11][9] or Asus Xtion [1], and associated computer software have loudly been revolutionizing human-computer interactions by enabling users to control their virtual avatars without requiring any proxies but their own bodies. Microsoft Kinect SDK, for instance, provides markerless full-body tracking by extracting 20 joints of the user's body at 30fps and establishes a gliding wireframe skeleton. Using kinematic features

* This work has been funded in part by TÜBİTAK Project 109E134 Vipsafe.

extracted from such a powerful representation, statistical learning algorithms can interpret the user's gestures in order to control the interaction [11][5][10].

In the present work, our aim is to find the most discriminative subset of kinematic features that represent an action, which is understood here as a sequence of movements generated by a human agent during the performance of a task. Representation of the human motion via a moving skeleton plays a crucial role in the overall action recognition pipeline. Even though not so much as with conventional video, skeletal data obtained from the processing of raw data acquired from a depth camera are still prone to uncertainty due to anthropometrical differences across users. In addition, geometrical invariance issues, arising due to camera-user position variability are also present. In this work, we employ translation and rotation-invariant features to over-come the Euclidean-part of the invariance problem (Section 3.2). To alleviate user variability due to inherent action performance differences and minor scaling, we rely on discriminatively selected features.

To find the most effective and efficient subset of features for a given set of actions from a high-dimensional spatiotemporal feature space, we first discriminatively optimize a random decision forest (RDF) [3] model over a forest-specific set of hyper parameters and then, we collect all the unique features from the nodes of each tree in the optimal forest. We feed this selected feature set into a linear support vector machine training procedure to learn the final classifier.

We test our classifier on two datasets acquired using Microsoft Kinect platform. The first one is the Microsoft Research Cambridge dataset (MSRC-12) [5], where our classifier attains an average accuracy of 94% on MSRC-12. We additionally test the classifier on the WorkoutSU-10 dataset collected in our laboratory, Sabancı University VPALAB. WorkoutSU-10 contains 10 exercise gesture classes, selected by professional sport trainers for therapeutic purposes. Our classifier reaches an average accuracy of 98% on this dataset, which will be soon released publically.

Our contributions in this paper are three-fold: (1) we use a large set of invariant spatiotemporal features extracted from skeletons in motion, (2) we introduce a discriminative RDF-based feature selection framework capable of reaching impressive action recognition performance when combined with a linear SVM classifier, and (3) we present a novel therapeutic action dataset to be soon released publically (as far as we know there is no available dataset recorded via Microsoft Kinect which contains both skeleton joints position and depth information and put in therapeutic exercises).

2 Related Work

The use of 3D geometrical information provides a clear advantage over using 2D image-based features. The work in [12] has investigated these two categories of approaches using a wide range of features and has shown that even with high levels of noise, the recognition process benefits from using pose-based features. As skeletal kinematic models encode key parameters of the limbs, they are con-

sidered as very powerful representations for a real-time motion analysis of the human body, although such models are difficult to extract and track from conventional video. The emergence of real-time depth cameras [9][1] has greatly simplified the extraction of human skeleton models and the tracking of skeletal key points such as joints. In [10], Raptis et al. present a real-time gesture recognition platform for classification of skeletal wireframe to evaluate dance gestures. Correlation and energy profiles computed from angular features at skeleton joints have been used for evaluation of the dance gestures. They obtain an average recognition accuracy of 96% on their own dataset. In [4], the authors present an algorithm capable to cope with the latency problem in interactive action-based systems. Their proposed classifier achieves an average recognition accuracy of 88.7% on MSRC-12 dataset and 90.06% on their own dataset. In spite of all these works, defining discriminative features and relationships for human motions still stay challenging. In that sense, our work explores the potential of feature selection techniques in identifying discriminative kinematic feature sets for action recognition.

3 Methods

3.1 Feature Extraction

For a faithful representation of the skeleton in motion leading to successful recognition, we extract relational features of joints in a pose at each time point. The tracking algorithm in the current version of Microsoft Kinect platform can effectively track 20 joints of the active person. In order to characterize human motion, we calculate the so-called motion or kinematic features at the joints during the whole course of the action performed. The collection of these features will be data-mined by RDF model optimization and selected features will be fed into SVM training. A good set of kinematic features in our context should satisfy at least the following requirements:

- **Invariance to the position and orientation of the sensor.** The features introduced in the sequel are all invariant to Euclidean motion, most of them being also scale-invariant.
- **Stability.** In order to ensure stability against unavoidable noise in the form of jitter to some extent, we smooth the skeletal joint coordinate position time-series by a Gaussian filter prior to the whole feature extraction process.
- **Invariance to intra- and interpersonal variability.** While informed geometric design can cope with variability in the way an action is performed, invariance of this kind cannot be guaranteed only by feature extraction. Higher-level information gleaned from classifiers should come into play.

With these ideas in mind, we first define a torso frame, which consists of seven joints, and apply a PCA on constructed matrix of torso joint coordinate positions [10]. These joints seldom move independently, as such they are instrumental in constructing a canonical coordinate frame for several features

described in the sequel. The torso frame is established by the following joints: neck, spine, hip center, right shoulder, left shoulder, right hip, and left hip (also indicated in blue in Fig. 1b). The first two basis vectors found by PCA (**u** and **r**) and their cross product **t** form the torso frame. Let the 20 joints of the skeleton be indexed by the set $J = \{0, \ldots, 19\}$ We adopt the following partitioning of the joint index set J as $J = \{Head\} \cup J_0 \cup J_1 \cup J_2 \cup J_3$. The set J_0 indexes the

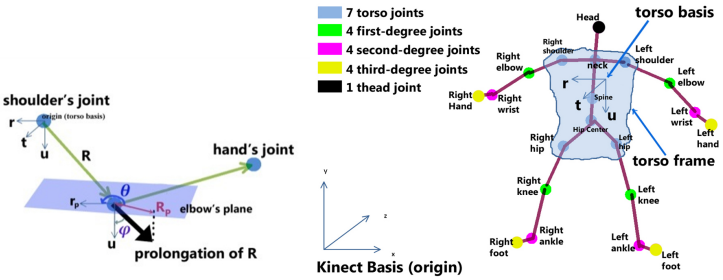

Fig. 1. (a) Joint representation and (b) skeleton model.

seven torso joints shown in the blue shaded area of Fig. 1b. The set J_1 indexes the four first-degree joints (shown in green in Fig. 1b). The set J_2 indexes the four second-degree joints (shown in pink in Fig. 1b). The set J_3 indexes the four third-degree joints (shown in yellow in Fig. 1b). This representation is similar to one that is proposed in [10][7]. We can now define the following types of features.

Type-I features. We define eight pairs of azimuth and elevation (θ^j, φ^j) angles for each joint $j \in J_1 \cup J_2$ with respect to the torso frame (Fig. 1a) to render them rotation-invariant at each time point. Accordingly, there are 16 angular features in total.
Type-II features. Let $p^j \in \mathbb{R}^3$ be the 3D position of the joint j. The interjoint distance $d^{i,j}$ is then defined as the Euclidean distance between joints i and j. We will have $\binom{20}{2} = 190$ such distance features in total.
Type-III features. We also define the three coordinate components of the 3D velocity vector of an individual joint with respect to the torso frame as a new type of feature. There are 20×3 =60 velocity features in total.

Notice that since all features are calculated during the whole course of the motion, each quantity above forms a time-series. Once the pool of features is constructed, we denote the set of all features generically as $f^{(k)} = \{f^{(k)}(t_n)\}$, where $k = 1, \ldots K$ runs over the set of features $(K = 266)$ and n runs over the set of N consecutive time points. There will be KN feature values in our collection.

3.2 Matching Strategies

In order to extract a baseline classification performance to improve upon, we first employ correlation-based matching and aggregation strategies using the whole set of spatiotemporal features. Given a description for each action instance, we can compare two action instances U and V and by computing the similarity between their respective feature sets $F = \{f^{(k)}\}$ and $G = \{g^{(k)}\}$. A natural basic procedure is to first compute the normalized correlation coefficient between each corresponding pair of features $f^{(k)}$ and $g^{(k)}$ in F and G respectively, and then to aggregate the K correlation values into a unique similarity between sets F and G by averaging. Using this aggregate similarity measure, the classification procedure is to determine the label of the test instance V based on the labels of the database instances in front of the ranked list of similarities by:

- **K-NN:** Assign the majority label amongst the K-first labels in the ranked list.
- **Classwise score average:** Average the scores separately for each class then assign the label of the class having the maximum average.
- **Classwise score product:** Compute the score product for each class then assign the label of the class having the maximum score product.
- **K-first versions of classwise average and product.**
- **Borda count.**

3.3 RDF-based Feature Selection

A random decision forest (RDF) is a collection of decision trees, where each tree is grown randomly. While a RDF is in general used as a discriminative classifier by itself, in this work we employ it as a discriminative feature selection tool. More specifically, we leverage (i) one of the randomization mechanisms coming into play in RDF training, random node optimization, and (ii) the easily interpretable node structure of decision trees as will be described shortly.

There are two means of injecting randomness into the decision forest growing process. The first one is to train each tree on a different random subset of the original training set, in much the same way as in bagging [2]. In the following account, we do not use this mechanism since we did not find any noticeable positive effect as compared to making the whole training set available to each tree. The second mechanism, that we heavily rely upon, is to optimize each node in each tree over a set of parameters chosen randomly from the whole parameter set.

In standard decision tree learning, a node is optimized with respect to a purity measure such as information gain [8] or [6] in order to split the input set of instances into two subsets, each of which is as "pure" as possible in terms of class labels. The split at a node is characterized by a so-called split function, which, in our context, is instantiated as $s \stackrel{\text{def}}{=} f^{(k)}(t_n)$, where k indexes the feature time-series and t_n the time point. Given a test instance U with description $F = \{f^{(k)}\}$, the action of the split function on U is expressed as follows: if

$f^{(k)}(t_n) < \tau$, assign U to the left split; otherwise to the right split, where τ is the decision threshold. As such, the feature index k, the time point t_n and the decision threshold τ are the parameters to be optimized at each node at each tree during forest training. Random node optimization consists of testing only a fraction of these parameters instead of running an exhaustive search over the whole parameter space, which would be computationally prohibitive if performed at each node of each tree in the forest. In our case for instance, there are K=266 feature time-series and N=50 time points (totaling K×N=266×50=13300 unique features) to test; combined with the number of thresholds to test and the number of all nodes, exhaustive search would be impossible for all practical purposes. That's exactly where random node optimization becomes handy in that not all parameter configurations are tested but only a small subset of them. After training the forest with a specific configuration, one can identify and collect the features selected at each node as the most discriminative ones within the original pool of features. The natural question is then which configuration should be used to grow the forest.

We consider the RDF training model selection problem in conjunction with feature selection since a specific RDF leads to a specific set of features retained. To this end, we follow a discriminative model validation approach. We set a range for the model parameters ($N_{tree}, N_{feature}, D, N_{threshold}$), and then we train a RDF for each configuration on some training set. Then, we evaluate the performance of the RDF classifier for each configuration on a separate set. Once we have all validation performances, we choose the forest in view of its validation accuracy and its feature reduction efficiency, which is defined as $efficiency = 1 - K_r/KN$, where K_r is the number of uniquely retained features in the course of training a specific forest and KN is the total number of features. As will be seen in Section 4.2, this procedure lets us select the feature set, which gives "the most bang for the buck".

4 Experiments

In our experiments, we focused on two sets of actions used in MSRC-12 and our novel dataset WorkoutSU-10. In both datasets, we used the provided 3D positions of the joints as determined by Microsoft's markerless motion capture system.

MSRC-12 dataset comprises 12 gesture classes, six of them corresponding to first-person-shooter game actions (iconic gestures) and the other six are gestures of a music player (metaphoric gestures). The gestures are performed by 30 subjects ranging from 22 to 65 years old. The dataset contains 6244 action instances while there are unequal repetitions of each action classes.

WorkoutSU-10 dataset comprises 10 gesture classes selected by professional trainers for therapeutic purposes. There are three broad category of exercises in the dataset (balance, stretching and flexibility, strengthening). We have chosen to provide participants with a combined modality before performing each exercise. The combination was an animated character performing the exercise and

a subscripted text explaining the instructions. All of the recordings have taken place in our laboratory in different days and the performances were also recorded using a video camera. In addition to skeletal joints, also the depth images of the performed exercises were recorded. 12 participants (9 male and 3 female ranging from 20 to 30 years old) were recruited from students in Sabancı University and each one performed each exercise for 10 times. In total, there are 1200 action instances in this dataset. The recording procedure has been conducted in a way that after reading the instruction and watching the video, the participants have been asked to stand in front of a green screen with a Kinect sensor in front of it and the recording has been started when the participant indicated he/she was ready.

4.1 Performance Results

For template matching using the aggregation methods described in section 3.2, after constructing of the correlation pool between each pair of action instances in dataset, we have carried out a leave-one-subject-out cross-validation (LOSOXV) test on the MSRC-12 dataset. The evaluation test has been repeated for K-NN, K-first average and K-first product method K=1, 3, 5, and 7. We have linearly interpolated and resampled all time-series instances to 50 samples. We report the performance in Table 1. The best aggregation methods turn out to be classwise K-first product and classwise K-first average with K = 3. We also note that all schemes except whole classwise average and Borda count, performances are similar within 0.8% performance points. Aggregation schemes use all the KN=13300 features with a simple but powerful NN-based classifier. As such, they provide a baseline performance that, with our RDF-based feature selection mechanism, we aim at maintaining while reducing the number of features significantly.

Table 1. LOSOXV Performance Results of Aggregation Methods on MSRC-12

Class	K-NN				Classwise K-First Product			Classwise K- Average			Whole Classwise Average	Borda Count
	K				K			K				
	1	3	5	7	3	5	7	3	5	7		
Kick	96.1	96.5	96.1	96.5	**96.5**	96.5	96.3	**96.5**	96.5	96.3	91.4	91.6
Beat Both	84.5	84.7	85.7	86.0	**85.5**	86.2	85.9	**85.5**	86.2	85.9	84.1	32.0
Change Weapon	85.7	84.9	84.1	82.5	**85.3**	84.1	83.9	**85.3**	84.1	83.9	70.1	43.8
Had enough	91.5	90.2	90.0	90.4	**90.6**	90.0	90.2	**90.6**	90.0	90.2	67.5	53.9
Throw	95.9	95.5	95.3	95.5	**95.7**	95.7	95.7	**95.7**	95.7	95.7	89.5	81.7
Bow	98.8	98.8	98.4	98.2	**99.0**	98.6	98.4	**99.0**	98.6	98.6	93.9	89.3
Shoot	86.9	85.1	84.5	83.6	**86.5**	85.9	85.5	**86.5**	85.9	85.5	41.5	16.4
Wind Up	95.8	95.7	95.2	95.1	**95.8**	95.4	95.8	**96.0**	95.5	95.8	52.1	81.0
Goggles	96.1	97.5	97.3	97.3	**97.1**	97.3	97.3	**97.1**	97.3	97.3	93.5	92.6
Push Right	91.8	92.9	92.1	91.2	**93.3**	92.7	92.0	**93.3**	92.7	92.0	58.0	46.7
Duck	99.6	99.4	99.4	99.4	**99.6**	99.6	99.4	**99.6**	99.6	99.4	98.2	90.4
Lift Arms	89.2	89.6	89.8	90.2	**90.0**	90.6	90.4	**90.0**	90.6	90.2	83.3	68.7
Average	92.7	92.6	92.4	92.2	**93.0**	92.8	92.6	**93.0**	92.8	92.6	76.3	66.0
standart deviation	5.15	5.52	5.46	5.77	**5.19**	5.30	5.37	**5.20**	5.30	5.40	18.68	26.30

In RDF-based feature selection, in order to find the best forest configuration, we have trained 27 forests with different tree sizes (10, 50 and 200 tree), tree

depths (4, 6 and 8), number of selected features at each node (10,100 and 1000). We have observed that the number of thresholds to test at each node had no influence in the performance. In the validation runs, we have used cross-subject cross-validation (CSXV). Accordingly, we have split each dataset (MSRC-12 and WorkoutSU-10) into two groups *A(training)* and *B(testing)*. For MSRC-12, the number of instances in sets A and B was 3600 and 2645 respectively. For WorkoutSU-10, the split has resulted in 600 instances in both sets A and B. We have reserved the set A for training the 27 RDFs and the set B to evaluate the cross-validation performance. Fig. 2 depicts CSXV validation vs. feature reduction efficiency curve on MSRC-12. We picked the configuration giving the most sensible compromise between accuracy (93.2%) and feature reduction efficiency (91%), that is, the forest with number of trees = 50, maximum tree depth = 6 and number of selected features at each node = 100. It's reassuring to see that the 93.2% CSXV performance obtained with this scheme coincides with the 93.0% LOSOXV performance obtained with the full feature set using aggregation on MSRC-12. The nodes of the RDF trained with this configuration contained 1225 unique features, corresponding to a reduction better than one tenth with respect to 13300 features in total. After this selection procedure, we have trained a linear SVM classifier using the 1225 retained features on the set A of MSRC-12. We have applied a 5-fold cross-validation to find the regularization parameter of the SVM. The classification performance on set B of MSRC is shown in Table 2-left. It can be observed that the linear SVM using the reduced feature improves the performance even further (by 1%). The results of the same procedure on WorkoutSU-10 is shown in Table 2-right, the average accuracy turns out to be 98% with a quite balanced classwise performance. Note that the RDF trained with the best configuration above has yielded 1398 unique features on WorkoutSU-10 corresponding to 89.8% reduction efficiency. Confusion matrices for individual classes in both datasets are shown in Fig. 3-left and Fig.3-right.

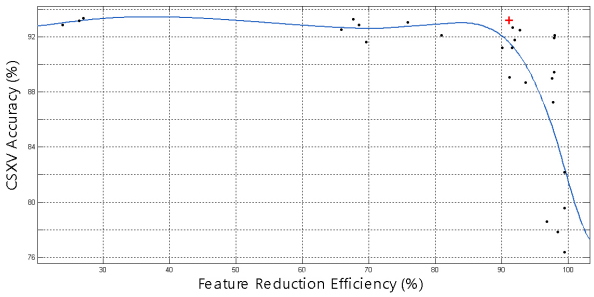

Fig. 2. Selection of the best forest configuration using the accuracy vs. feature reduction efficiency measure on MSRC-12.

4.2 Insights on Selected Features

By data mining the results, we can investigate the feature selection process in order to find out which feature type has the most effect on the performance of the classifier. We calculated the ratio of selected features in each type of features with respect to the total number of features in that type which turned out that this ratio is 11.04% for type I, 4.77% for type II and 21.40% for type III. Since we have a large initial feature set, there is considerable reduction in number of features used in each type. Feature type III (velocity of the joints) is the most selected type in tree nodes, suggesting they are the most influential features for the classifier performance. We have also looked at the ratio of the selected features in skeleton by group of joints each one belongs to. It can be observed that features from leg joints have more impact the remaining joints (with 24.58% for right and 24.09% for left leg), which have more or less the same influence(18.51% for right and 15.57% for left arm and 17.25% for torso and head together).

5 Conclusion

In this paper, we proposed a discriminative RDF-based feature selection framework capable of reaching impressive action recognition performance when combined with a linear SVM classifier. Our results showed state-of-the-art performances in action classification, beating for instance the 88.7% performance of Ellis et al.'s work [4] on MSRC-12. The large, but possibly redundant set of invariant spatiotemporal features extracted from the skeleton in motion have been data-mined thanks to discriminative capabilities of RDF in order to reach comparable or even better performance with a significantly reduced number of features (one tenth of the original set). Furthermore, we have introduced a novel therapeutic action recognition dataset to be prospectively used by the action recognition community. Our future work will concentrate on expanding this dataset and to develop a joint feature extraction and selection mechanism within the RDF framework. It should be stressed that while our designed framework is planned to be used as part of a therapy application, it also could be applicable in other action recognition tasks such as assisted living, intelligent surveillance and games.

References

1. Asus `http://www.asus.com/Multimedia/Xtion_PRO_LIVE/`
2. Leo Breiman. Bagging predictors. *Machine learning*, 24(2):123–140, 1996.
3. Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.
4. Chris Ellis, Syed Zain Masood, Marshall F Tappen, Joseph J LaViola, and Rahul Sukthankar. Exploring the trade-off between accuracy and observational latency in action recognition. *International Journal of Computer Vision*, pages 1–17, 2012.

Table 2. Classification Performance of the Linear SVM with RDF-selected features

MSCR-12		WorkoutSU-10	
Class	Accuracy	Class	Accuracy
Kick	99	Hip Flexion	100
Beat Both	90.2	Trunk Rotation	93.3
Change Weapon	90.9	Lateral Stepping	98.3
Had Enough	90.3	Thoracic Rotation	98.3
Throw	92	Hip Adductor Stretch	96.6
Bow	99	Hip Stretch	100
Shoot	93.8	Curl-To-Press	100
Wind Up	92.8	Free Standing Squats	98.3
Goggles	100	Horizontal Punch	100
Push Right	81.9	Oblique Stretch	95
Duck	100		
Lift Arms	100		
Average	94.03	Average	98
standart deviation	5.62	standart deviation	2.34

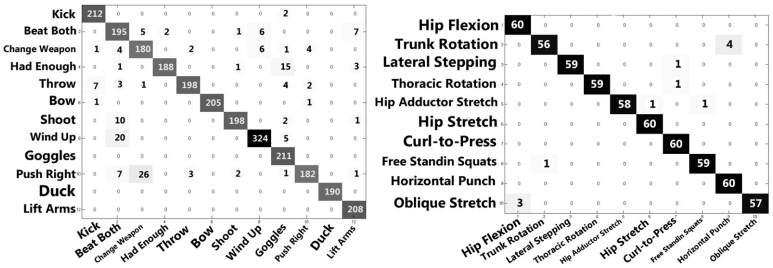

Fig. 3. Confusion matrices for MSRC-12 (left) and WorkoutSU-10 (right) using linear SVM with RDF-selected features

5. Simon Fothergill, Helena Mentis, Pushmeet Kohli, and Sebastian Nowozin. Instructing people for training gestural interactive systems. In *Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems*, pages 1737–1746. ACM.
6. C Gini. Concentration and dependency ratios. *Rivista di Politica Economica*, 87: 769–792, 1997.
7. Kinect. Microsoft kinect documentation may 2012 sdk rlease.
8. Solomon Kullback and Richard A Leibler. On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86, 1951.
9. Microsoft. Microsoft corp. redmond wa. kinect for xbox 360.
10. Michalis Raptis, Darko Kirovski, and Hugues Hoppe. Real-time classification of dance gestures from skeleton animation. In *Proceedings of the 2011 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 147–156. ACM.
11. Jamie Shotton, Andrew Fitzgibbon, Mat Cook, Toby Sharp, Mark Finocchio, Richard Moore, Alex Kipman, and Andrew Blake. Real-time human pose recognition in parts from single depth images. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1297–1304. IEEE.
12. Angela Yao, Juergen Gall, Gabriele Fanelli, and Luc Van Gool. Does human action recognition benefit from pose estimation?". In *Proceedings of the 22nd British machine vision conference-BMVC 2011*.