Adaboost-Based Algorithm For Human Action Recognition

Nabil Zerrouki[a], Fouzi Harrou[b]
[a] University of Sciences and Technology Houari Boumédienne, LCPTS, Faculty of Electronics and Computer Science, Algiers, Algeria
nzerrouki@usthb.dz , fouzi.harrou@kaust.edu.sa

Ying Sun[b], Amrane Houacine[a]
[b] King Abdullah University of Science and Technology, CEMSE Division, Thuwal, 23955-6900,Saudi Arabia
ahouacine@usthb.dz

***Abstract*—This paper presents the design and implementation of computer vision-based human action recognition. Firstly, the shape based pose features are constructed based on area ratios for identifying the human silhouette in images. Indeed, the proposed features have the interesting property of invariance to translation and scaling. Once the human body features are extracted from videos, different human actions are learned individually on the training frames of each class. The Adaboost algorithm is used for the classification process. Finally the experimental results are obtained based on "UR Fall Detection" dataset. Six classes of activities are considered namely: walking, standing, bending, lying, squatting, and sitting. For evaluation, different statistical measures have been considered such as overall accuracy and the area under ROC curve (AUC) value. Results demonstrate the efficiency of the proposed methodology.**

Keywords—fall detection; cascade classifier; gesture recognition; vision computing.

I. INTRODUCTION

The need for human action monitoring and recognition is increasing day after day. This urgency made this topic one of the most highly prioritized in the behavior understanding community. In fact, human activity analysis is very useful in many applications, such as smart rooms, interactive virtual reality systems, people monitoring, environment modeling, just to name a few. In this paper we have selected one of the most challenging problems in computer vision known as "Human action recognition". Various vision systems have been investigated, such as single charge coupled device camera [1], multiple cameras [2], specialized omnidirectional cameras [3] and stereo-pair cameras [4]. In the present work, human action classification is based on video sequences acquired from a single RGB camera.

In general, vision-based methods include four major phases namely: data collection, image segmentation, feature extraction, action classification [5]. During data acquisition, video sequences corresponding to different human activities are recorded. The segmentation phase consists of extracting the body's silhouette from the input frame while the feature extraction step determines discriminative information needed to describe the human silhouette. Finally, the classification phase is required to distinguish between different human activities. Each sequence will be attributed to a defined class according their corresponding features.

II. MOTIVATION AND CONTRIBUTION

It is well known that there is not a single classifier which is always the most accurate [6]. In any application, several classifiers could be used, and with certain algorithms, there are some parameters that affect the classification accuracy. For example, in the case of neural network classifier (multilayer perceptron), we should sit different parameters such as: the number of hidden layers, the number of nodes in each layer, etc. The classical technique is to try different parameter values and select the one that performs the best on a separate validation set. The No Free Lunch Theorem states that there is no single algorithm that in any domain always induces the most accurate classifier [7]. The main idea of this paper is to use an alternative classification way that may be accurate on these.
In this study, classification is performed by the use of multiple algorithms that complement each other so that by combining them, we attain higher accuracy. The classifiers are not chosen for their accuracy, but for their simplicity. One important note is that when we use multiple classifiers, we want them to be reasonably accurate but do not require them to be very accurate individually, so it is not needed to optimize each classifier separately for best accuracy. In this study, we implemented the Adaboost classifier using Decision Stump as weak classifier.

The choice of Adaboost algorithm is motivated by its ability to exploit the combination of many relatively weak and inaccurate classifiers to resolve complex recognition problems which can be very beneficial to the application at hand [8]. The main contributions of this paper, is to adopt Adaboost algorithm to multi class problem in order to separate between different human activities.

The organization of the rest of the paper is as follows: In Section 2 the environment modeling and people segmentation is reported. Section 3 describes the area ratio feature set as well as a shape based feature set used as reference. In Section 4 the Adaboost classification of human actions is presented. Finally, the obtained results and comparative tests are illustrated in Section 6.

III. ENVIRONMENT MODELING AND PEOPLE SEGMENTATION

The segmentation consists in extracting the body silhouette from image sequence. This latter is a very essential initial step for many vision-based applications. And it remains a challenging task to achieve automatic and robust action recognition in cases where no prior information is available on background, lighting changes, environment constraints, .etc [9]. Several segmentation techniques have been proposed in the literature. The simple Gaussian background subtraction method was used by Bouwmans et al. [10]. In this paper, human body segmentation is based on the background subtraction technique. The background model is obtained by using successive frame differences for registering the stationary pixels. A threshold is needed to decide whether a pixel belongs to the foreground or the background.

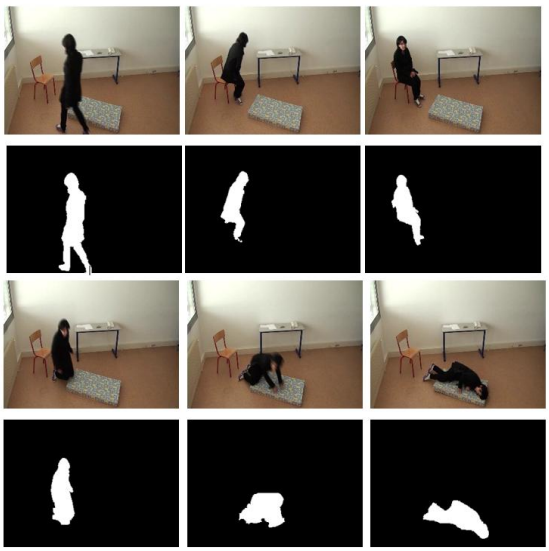

Fig. 1. Result of background subtraction algorithm

After the background subtraction, some noise regions can be encountered. To eliminate this noise, the morphological operator based on erosion and dilation operators with 3×3 structuring elements is applied. An example of background subtraction technique is shown in Fig.1. The input images correspond to the first raw, while the second raw illustrates the background subtraction results.

IV. HUMAN BODY FEATURE EXTRACTION

Accurate feature extraction is necessary for video-based human action classification, where extracted features have a direct impact on classification accuracy. The extracted features have to be invariant to image translation when the position of human body changes in the image and to scaling when the dimensions of the silhouette change as the distance between the monitored person and camera varies. Several previous works have performed feature extraction by focusing on shape information to detect and classify falls; for example, the body's center of gravity [11], horizontal and vertical dimensions of the bounding box or approximated ellipse of the silhouette [12]. However, these features cannot always distinguish among body postures, especially when there is a high degree of similarity between activities (e.g., dimensions and orientations of the ellipse or the bounding box are the same for both bending and sitting postures, as shown in Fig. 2 a–c). For this reason, instead of using the body's geometrical shape we base the extracted features on non-zero pixels that constitute the human body. More specifically, we use five partial occupancy areas (i.e., the number of pixels in each area of an image) of the body to detect and classify falls. These areas typically correspond to the action of body parts when in a standing posture. As shown in Fig. 2 d, the body is divided into five portions. These areas were determined using the body's center of gravity (x_G, y_G), which is simply the barycenter of the pixels.

$$X_G = \frac{1}{N}\sum_{i=1}^{N} X_i, \quad Y_G = \frac{1}{N}\sum_{i=1}^{N} Y_i, \tag{1}$$

where N is the number of pixels representing the human body, and x_i and y_i denote the horizontal and vertical coordinates of the pixels composing the human body, respectively. Partitioning is performed by tracing five segments from the body's center of gravity. The first segment is vertical, the second and third segments are located at 45° on either side of the vertical segment and the fourth and the fifth segments are situated at 100° on either side of the third and fourth segments (as shown in Fig. 2 d). The center of gravity and the five areas are computed for each image to represent the feature frame that we expect to be a precise characterization for human gesture classification.

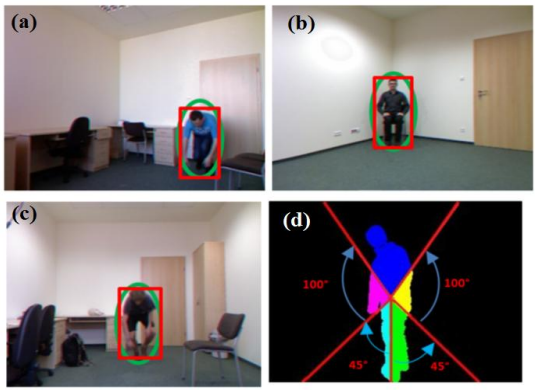

Fig. 2. Human body feature extraction

A normalization phase, where each area value is divided by the global body area, is performed to negate any concerns related to scaling. Given the total number of pixels making up the body area, A, and the number of pixels making up the partial areas Ai ; i = 1 . . . 5, the normalized partial areas.

$$R_i = \frac{A_i}{\sum_{i=1}^{5} A_i}, \tag{2}$$

Thus, the use of the area ratios permits a significant separation between different activities. Note that these area ratios are invariant to translation and scaling, and that they take into account the rotation information necessary for human activity classification [13].

V. HUMAN ACTION CLASSIFICATION USING MULTI-CLASS ADABOOST CLASSIFIER (MCAC)

The classification process consists of two steps: (i) assign the system certain video sequences as training samples, and (ii) classify the rest of sequences according to their feature spaces via a trained classifier model. Several types of classifiers have been deployed in the behavior understanding community [14]. As previously mentioned, we have selected an Adaboost classifier due to its high generalization performance. Even with a simple principle using weak classifiers, Adaboost algorithm can compete with many powerful classifiers. The idea behind Adaboost classifier is to use a sequence of weak classifiers in order to form a powerful classifier. The weak classifiers d_i are generally sorted according their space or time complexity. Cascading is a multistage process, where the classifier d_i is only used if all previous classifiers, d_k ($k < j$) are not satisfying enough. A weight value W_i is associated with each classifier d_i to evaluate its confidence. The classifier is only used if $W_i > \theta_i$ where θ_i is the confidence threshold.

$$y_i = d_i \ \ if \ \ w_i > \theta_i \tag{3}$$

$$and \ \ \forall \ \ k < j, \ w_k > \theta_k \tag{4}$$

In training phase, Adaboost sets an equal weight W_i for all weak classifiers. These weights are then iteratively optimized, where weights of misclassified samples are increased, while of correctly classified samples are decreased in order to focus more on misclassified samples. Unlike other powerful classifiers, such as Neural Network (NN) or Support vector machine (SVM), AdaBoost can achieve similar classification results with much less tuning of parameters or settings. The user only needs to choose: (1) which weak classifier might work best to solve their given classification problem; (2) the number of boosting rounds that should be used during the training phase. Several weak classifiers can be used at each round of boosting. The AdaBoost algorithm will select the weak classifier that works best at that round of boosting.

It is worth noting that the AdaBoost algorithm was initially designed to solve binary classification. To solve multi-class problems, we are inspired from the approach used in the multi-class SVMs, which is One-Against-All (OAA). This approach is the earliest and the widely used to solve a problem with N classes [15].

In this study, we have implemented the AdaBoost algorithm using Decision Stump as weak classifier (which is just one node of a Decision Tree). The main idea is to train the first AdaBoost on data from the first class on the one hand and the rest of the data from the other classes on the other hand. The second AdaBoost will then train on the second class and the rest of the data with the exception of the first class (which is already considered in the first AdaBoost), and other AdaBoost classifiers will be trained for the rest of the classes. The successive AdaBoost classification outputs represent a binary decision for each action class, hence the naming of cascade classifier [15].

VI. EXPERIMENTAL RESULTS

A. Evaluation Measures

1) Overall accuracy:

The percentage of actions being correctly classified and recognized constitutes the overall accuracy. This latter is computed as a ratio from the confusion matrix represented by the mean of the diagonal cells. It is worth noting that the accuracy is the most widely used empirical measure, it is expressed as:

$$acuuracy = \frac{tp + tn}{tp + fp + fn + tn}, \tag{5}$$

where tp is true positive, tn is true negative, fp is false positive, and fn is false negative. However, to obtain an unbiased global accuracy, we have conducted a k-fold cross-validation procedure.

2) Kappa coefficient

Another measure which can be extracted from a confusion matrix is the Kappa coefficient. It is a statistical measure of inter-raters agreement [16]. This measure is more robust than the accuracy measure since it subtracts the agreement occurring by chance. This coefficient is expressed as:

$$Kappa = \frac{P(a) - P(e)}{1 - P(e)} \tag{6}$$

where P(a) is the probability of relative observed agreement among raters, and P (e) is the probability of by chance agreement. The range of the kappa coefficient is [-1, 1].

3) ROC curve

ROC curve plots the true positives (sensitivity) vs. false positives (1−specificity) for a binary classifier system when the discrimination threshold is modified. In ROC coordinate system, X axis is marked by false positives (1−specificity), and Y axis is marked by true positives (sensitivity) [16]. The frame of coordinate axes describes the relation between income and cost of the model outputs. The area under curve (AUC) associated to the ROC evaluation method is proportional to the objective model performance. It represents a means to quantify the ROC curve performance using a single value. Since, a random method depicts the first bisector; it has an AUC value equal to 0.5. Efficient classifier's areas should have an AUC value larger than 0.5. It is well known that the higher the AUC value, the more efficient is the classifier.

B. Dataset and experiments

To analyze the efficiency of the proposed model, URFD fall detection dataset has been conducted [17] that comprises 70 video sequences of several actions performed in diverse ways (30 falls + 40 activities of daily living ADL) sequences of several actions performed in diverse ways. The images are acquired at 25 frames per second, and with a resolution of 640

× 480 pixels. In this database, videos are recorded from different environments and contain variable illumination like shadows and reflections that can be detected as moving objects. To investigate whether the proposed classification is capable of identifying human postures; we manually denoted the ground truth of data samples for 6 posture classes. To evaluate the proposed method, we have selected 5000 images. The selected images should contain one of the six postures previously defined. These images were then divided into training and testing samples using a 3-fold cross-validation procedure.

C. Results and interpretation

For the proposed human action classification, a confusion matrix is computed (see Table I), along with the overall accuracy and the Kappa coefficient. It is clear from these results that the proposed classification enables robust recognition in very challenging situations. It is also clear (from Table 1) that the bending class remains a challenge for the proposed approaches where it is characterized by the lowest classification accuracy (91.79%). In fact, the bending class is slightly confused with the sitting class. This confusion is mainly due to: (i) the important similarity presented by the two classes, (ii) dark environment and degraded illumination conditions, and (iii) segmentation errors generally induced by the presence of shadows; confusion of body limbs with environment objects. It is well-established that segmentation conducted in numerous areas is always error prone.

TABLE I. HUMAN POSTURE CLASSIFICATION RESULTS

		Reference data					
		Standing	***Lying***	***Bending***	***Sitting***	***Squatting***	***Knelling***
Classified data	***Standing***	***95.21***	*0*	*1.19*	*0*	*0*	*0*
	Lying	*0*	***100***	*0.91*	*0*	*0*	*0*
	Bending	*2.87*	*0*	***91.79***	*5.02*	*0*	*0.94*
	Sitting	*1.92*	*0*	*6.11*	***94.98***	*0*	*0*
	Squatting	*0*	*0*	*0*	*0*	***100***	*1.12*
	Knelling	*0*	*0*	*0*	*0*	*0*	***97.4***
	Overall accuracy = 96.56% ; Kappa coefficient = 0.96						

On the contrary, the standing, and kneeling classes are correctly classified in the most of cases. Furthermore Lying and squatting classes are perfectly accurate with 100% of its reference images; this result is explained by the efficiency of the area ratios as human body features and the adaptation of Adaboost classifier to the application at hand.

TABLE II. PERFORMANCE COMPARISON BETWEEN ADABOOST, KNN, NAÏVE BAYES, NEURAL NETWORK, AND SVM ALGORITHMS.

	Accuracy (%)	***Kappa coefficient***	***F-measure***
KNN	*91.09*	*0.895*	*0.91*
Naïve Bayes	*92.21*	*0.904*	*0.92*
Neural Network	*94.67*	*0.938*	*0.94*
SVM	*95.26*	*0.942*	*0.95*
Adaboost	***96.56***	***0.953***	***0.96***

As second evaluation, we compared the proposed classification with some powerful machine learning algorithms namely: KNN, Neural Network, Naive Bayes and SVM classifiers. To allow a fair comparison, the same video sequences are used for evaluation. Table 2 compares the proposed classification strategy with that using the other classifiers, where overall accuracy, F-measure, and the Kappa coefficient are calculated. The results shown in Table 2 demonstrate that Adaboost algorithm permits a robust classification. Compared to all of these classifiers, Adaboost remains significantly better, since it presents the highest accuracy.

VII. CONCLUSION

In this work we have presented an approach for human action recognition using video camera monitoring and adapting adaboost classifier. The experimental validation of the algorithm that was conducted on realistic image sequences of daily activities showed that the algorithm allows reliable human action recognition with low false positives ratio. This classification is evaluated using various complementary statistical measures. The experiments revealed that AdaBoost classifier achieved the most accurate results with much less tuning of parameters or settings than other powerful classifiers.

REFERENCES

[1] [1] D. Anderson, J. Keller, M. Skubic, X. Chen, and Z. He, "Recognizing falls from silhouettes," in 28th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS'06. IEEE, 2006, pp. 6388–6391.

[2] [2] R. Cucchiara, A. Prati, and R. Vezzani, "A multi-camera vision system for fall detection and alarm generation," Expert Systems, vol. 24, no. 5, pp. 334–345, 2007.

[3] [3] B. Jansen and R. Deklerck, "Context aware inactivity recognition for visual fall detection," in Pervasive Health Conference and Workshops. IEEE, 2006, pp. 1–4.

[4] [4] H. Liu and C. Zuo, "An improved algorithm of automatic fall detection," AASRI Procedia, vol. 1, pp. 353–358, 2012.

[5] [5] Zerrouki, N., Harrou, F., Sun, Y., Houacine, A. Accelerometer and camera-based strategy for improved human fall detection. Journal of medical systems, 40(12), 284, 2016.

[6] [6] Mitchell, T. M. (1999). Machine learning and data mining. Communications of the ACM, 42(11), 30-36.

[7] [7] ALPAYDIN, E., Introduction to Machine Learning, Second Edition, The MIT Press, 2010

[8] [8] Duda, R. O., Hart, P. E., & Stork, D. G.. Pattern classification. John Wiley & Sons,, 2012.

[9] [9] Zerrouki, N., Houacine, A. Automatic classification of human body postures based on curvelet transform. Int Conf Image Analysis and Recognition, pp. 329-337, Springer Int Pub, 2014.

[10] [10] Bouwmans, T., El Baf, F., Vachon, B. Background Modeling using Mixture of Gaussians for Foreground Detection - A Survey. Recent Patents on Computer Science, Bentham Science Publishers, 2008, 1 (3), pp.219-237.

[11] [11] B. Kwolek and M. Kepski, "Fuzzy inference-based fall detection using kinect and body-worn accelerometer," Applied Soft Computing, vol. 40, pp. 305–318, 2016.

[12] [12] H. Foroughi, B. Aski, and H. Pourreza, "Intelligent video surveillance for monitoring fall detection of elderly in home environments," in 11th International Conference on Computer and Information Technology, ICCIT 2008. IEEE, 2008, pp. 219–224.

[13] [13] Zerrouki, N., Harrou, F., Houacine, A., & Sun, Y. (2017, January). Fall detection using supervised machine learning algorithms: A comparative study. In 2016 8th International Conference on Modelling, Identification and Control (ICMIC). Institute of Electrical and Electronics Engineers (IEEE).

[14] [14] M. Yu, R. Miao, N. Adel, S. Mohsen, L. Wang, and J. Chambers, “A posture recognition-based fall detection system for monitoring an elderly person in a smart home environment,” IEEE Transactions on Information Technology in Biomedicine, vol. 16, no. 6, pp. 1274–1286, 2012.

[15] [15] Wang, Yubo, et al. "Real time facial expression recognition with adaboost." Proceedings of the 17th International Conference on Pattern Recognition, Vol. 3., 2004.

[16] [16] Hand, D.J.: Assessing the performance of classification methods. International Statistical Review 80, 400–414, 2012.

[17] [17] B. Kwolek and M. Kepski, “Human fall detection on embedded platform using depth maps and wireless accelerometer,”Computer methods and programs in biomedicine, vol. 117, no. 3, pp. 489–501, 2014.