Vision-based Human Action Classification Using Adaptive Boosting Algorithm

Nabil Zerrouki, Fouzi Harrou, Member, IEEE, Ying Sun and Amrane Houacine

Abstract—Precise recognition of human action is a key enabler for the development of many applications including autonomous robots for medical diagnosis and surveillance of elderly people in home environment. This paper addresses the human action recognition based on variation in body shape. Specifically, we divide the human body into five partitions that correspond to five partial occupancy areas. For each frame, we calculated area ratios and used them as input data for recognition stage. Here, we consider six classes of activities namely: walking, standing, bending, lying, squatting, and sitting. In this paper, we proposed an efficient human action recognition scheme, which takes advantages of superior discrimination capacity of AdaBoost algorithm. We validated the effectiveness of this approach by using experimental data from two publicly available databases fall detection databases from the University of Rzeszow’s and the Universidad de Malaga fall detection datasets. We provided comparisons of the proposed approach with state-of-the-art classifiers based on the neural network, K-nearest neighbor, support vector machine and naïve Bayes and showed that we achieve better results in discriminating human gestures.

Index Terms—Fall detection, cascade classifier, gesture recognition, vision computing.

I. INTRODUCTION

HUMAN action recognition and human behavior understanding are becoming increasingly important and attracted many research efforts over the past two decades [1]–[3]. Moreover, human activity analysis is a key enabler in the development of many applications, such as smart rooms, interactive virtual reality systems, people monitoring, and environment modeling [2], [4]–[6]. Approaches for classification of human activities are typically based on information captured by cameras or sensors. Two essential categories of approaches of human actions recognition can be distinguished: wearable sensors-based and vision-based methods. Wearable sensors-based approaches are mainly based on position sensors, and three-axis accelerometers. Mukhopadhyay et al. proposed a monitoring system based on physiological parameters monitoring to detect abnormal and unforeseen situations [7]. Shany et al. proposed an approach for movement and falls monitoring system based on wearable sensors [1]. Other human motion analysis systems based on adaptable wearable sensors are proposed such as inertial sensors, accelerometers [3], [8], [9]. Methods in this category can be affected by background noise, negatively impacting their performance by increasing the number of false alarms. Nowadays, small sensors are available and embedded in many daily devices such as in smartphones and smartwatches.

On the other hand, vision-based approaches focused on using information extracted from video sequences for recognizing human actions [10]. Approaches in this category have become much more important than before, due to the advancements in computer vision, pattern recognition, and image processing fields [11]. In the literature, there has been much discussion on human monitoring systems, in particular for elderly people [12]. Several systems have been developed including single and multi-camera tracking [13], [14], human identification, multi-camera activity analysis [15], [16]. Lee et al. used connected components labeling to identify the human activity by using spatial and geometric orientations features [17]. Althloothi et al. presented multiple kernel learning technique based on shape representation and kinematic structure for silhouette description [18]. Rougier et al. applied Gaussian mixture model on silhouette templates to identify gestures [19]. Wang et al. proposed neural network and SVM classifiers for recognition based on both histograms of oriented gradients (HOG) and local binary pattern (LBP) [20]. Nunez et al. proposed a vision-based solution using convolutional neural networks to detect if a video sequence contains fall incidents [21]. Indeed, human behavior understanding remains a challenging task because of the complexity of monitored environments, which are becoming more and more complicated and crowded.

This paper presents a flexible and efficient vision-based approach for human action recognition (see Figure 1). This procedure consists of four major phases namely: data collection, human body segmentation, feature extraction, and action classification. In image segmentation, the body’s silhouette is extracted from the background and then used in feature extraction step, which generates discriminative information needed as input in classification step. Specifically, in this study, we used the shape-based pose features, which are computed based on area ratios to identifying the human silhouette in images. Once the human body features are extracted from videos, different human actions are learned individually by using the training frames of each class. Each sequence will be attributed to a defined class according to their corresponding features.

Obviously, there is no one single classifier that is always the most accurate for all applications [22]. In any classification application, there are several candidate classifiers that could
be used. Generally speaking, the accuracy of some classifiers depends on tuning parameters that affect the classification performance. For instance, if a neural network classifier (multilayer perceptron) is used for classification, then different parameters should be set, such as the number of hidden layers and the number of nodes in each layer. Cross-validation technique is usually adopted to select the best classifier. Furthermore, the No Free Lunch theorem states that there is no single algorithm that in any domain that always induces the most accurate classifier [23]. The main idea of this paper is exploiting different classification algorithms to achieve improved accuracy. In this paper, classification is performed by using many algorithms that complement each other and may result in higher accuracy. Towards this end, classifiers are selected for their simplicity, not for their accuracy. In other words, we choose classifiers with reasonable accuracy and do not require them to be very accurate individually. So optimizing each classifier separately for best accuracy is not needed. Specifically, here we implemented the AdaBoost classifier using Decision Stump as a weak classifier.

The main contribution of this paper is the adoption of the AdaBoost algorithm to a multi-class problem for discriminating between different human activities. The choice of the AdaBoost algorithm is mainly motivated by its capacity to exploit many relatively weak classifiers to resolve complex recognition problems [24]. We tested the proposed approach using two experimental available datasets, the UR Fall Detection [25] and the Universidad de Málaga fall detection (UAMFD) datasets [26]. We considered six classes of activities namely: walking, standing, bending, lying, squatting, and sitting. Results show that the proposed method outperformed K-nearest neighbor, neural network, naïve Bayes and support vector machine (SVM) classifiers, and showed the highest accuracy.

The organization of the rest of the paper is as follows. In Section II the environment modeling and people segmentation are presented. Section III describes the area ratio feature set used in this study. In Section IV the Adaboost classification of human actions is presented. Section V presents and discusses the results, and Section VI lists the conclusions of the study.

II. ENVIRONMENT MODELING AND PEOPLE SEGMENTATION

The aim of the segmentation task is to extract the body silhouette from an image sequence [27]. Several segmentation techniques have been proposed in the literature. In [28], Bouwmans et al. proposed a simple Gaussian background subtraction technique for human body segmentation. In this approach, the background model is obtained by using successive frame differences for registering the static regions. Pixels constituting human silhouette are marked as foreground according to a threshold value. It is worth noting that segmentation based on background subtraction is widely used in literature. However, the main limitation of this technique resides in its assumption of a stationary background. Many benchmarks consider this problem by adding some static objects during the motion detection task, such as furniture and bag. Even if these objects do not appear in the initial background, they should be considered as background’s pixels. In this study, we employed an updating background based on running average method. The background model is updated by integrating the new incoming frames. The new background pixel is then calculated from a stationary average of pixels.
This method is efficient and simple to implement, where no prior knowledge or cluster observation of pixels is needed. After the background subtraction, some noise pixels can be encountered. To reduce this noise, a filtering phase is then applied using morphological operators namely holes filling, erosion and dilatation [10]. To take into account contextual information, structuring element with the dimension of $3 \times 3$ is applied. Figure 2 illustrates some examples of human body segmentation. The first column corresponds to current images, while the background subtraction results are shown in the second column.

Fig. 2. Result of segmentation using background subtraction algorithm. The left images are the original images, (Top) sitting on a sofa, (Middle) squatting position, and (Bottom) standing up position, while the right images are their corresponding segmented images.

III. Feature extraction

Extraction of human body features is central to vision-based human action recognition [29]. The generated attributes should be invariant to body translation and scaling in the frame. Several procedures have been proposed for feature extraction based on shape information, such as the body’s center of gravity [30], bounding box’s horizontal and vertical sizes, or approximated ellipse of the human body [31]. However, many of these attributes present shortcomings in discriminating between body postures, in particular for activities with high similarity. For example, in the case of bending and sitting on a sofa positions, the approximated ellipse or bounding box present very close values of sizes and directions (Figure 3(a-b)). To bypass this shortcoming, here, we extract features based only on pixels constituting human silhouette instead of using geometrical shape. Specifically, a sample partitioning is applied to a human body, to generate five occupancy zones (i.e., each zone is represented by the number of pixels that belonged to it). The five areas are the result of five-segment partitioning based on body’s center of gravity ($X_G, Y_G$) (Figure 3(c)).

$$X_G = \frac{1}{N} \sum_{i=1}^{N} x_i, \quad Y_G = \frac{1}{N} \sum_{i=1}^{N} y_i,$$

where $N$ is the set of pixels constituting the whole body, $x_i$ and $y_i$ represent vertical and horizontal coordinates of pixels. The first vertical segment separated the first two areas $A_1$ and $A_2$ (Figure 3(c)). The next two segments towards the direction of $45^\circ$ separated areas $A_3$ and $A_5$. Finally, area $A_4$ is situated between the two remaining segments towards the direction $100^\circ$. In fact, these areas correspond to the silhouette parts in the standing posture, i.e., head ($A_4$), arms ($A_3$ and $A_5$), and legs ($A_1$ and $A_2$). Then, we normalize the computed five areas (i.e., $R_i; i = 1 \ldots 5$) by dividing each area value $A_i$; $i = 1 \ldots 5$ by the total silhouette area, $A$:

$$R_i = \frac{A_i}{\sum_{i=1}^{5} A_i}.$$

These features are invariant to translation and scaling and take into account the rotation information needed for human activity classification [32]. The extracted features are descriptive enough to represent human postures and not too computationally complex permitting a fast treatment.

The five ratios are computed for each image of the video sequence (see Figure 4). Figure 5 shows an example of the time evolution of the five features ($R_1, \ldots, R_5$) while performing different activities.

IV. Multi-class Adaboost algorithm for human action classification

The classification procedure comprises two phases: (1) build a representative model using training video sequences, and then (2) use the constructed model to classify the new testing
Fig. 4. The five features extracted from each image are used for action recognition in video sequence.

Fig. 5. A sample of five ratios time evolution while performing different activities: standing, bending, sitting and squatting.

samples. Numerous classification algorithms were developed in the behavior understanding community [11]. Here, we use an adaptive boosting (AdaBoost) classifier for vision-based human action classification [33], [34]. This choice is mainly for its simplicity, flexibility and high generalization performance [34]. Indeed, the AdaBoost classifier is flexible and simple to implement since it is based on several weak classifiers. Furthermore, Adaboost classifier use features of each class separately, where the use of the whole feature vector (large sets of features) at the same time may not be necessary. In addition, AdaBoost adjusts adaptively the errors of weak classifiers allowing the achievement of high accuracies with much less tuning or settings than other powerful classifiers.

The basic philosophy behind AdaBoost classifier is to combine various weak classifiers (i.e., slightly better than random) to construct a powerful classifier. Indeed, the weak classifiers, \( h_i \), are usually ranked according to their space or time complexity (see Algorithm 1). An AdaBoost procedure is implemented by cascading several classifiers in a multistage process. This means that each classifier \( h_i \) is executed only if all previous classifiers, \( h_k \) (\( k < i \)) are not satisfying enough. Towards this end, an error rate \( \varepsilon_i \) is assigned to each classifier \( h_i \) to evaluate its confidence. The classifier is used only when \( \varepsilon_i > \theta_i \), where \( \theta_i \) is the confidence threshold.

During a training step of AdaBoost, all samples are equally weighted with \( W_i \). Then, these weights are iteratively optimized by increasing weights corresponding to misclassified samples. In AdaBoost procedure, many weak learners can be combined in a weighted sum to achieve the final output of the boosted classifier. AdaBoost can reach high classification performance with less tuning of parameters compared to other commonly used classifiers such as Neural Network and SVM.

In the implementation of AdaBoost we only select: (i) a weak classifier to be used for the given classification problem; (ii) the number of boosting rounds used for training step. Many weak classifiers can be used in each round of boosting. The AdaBoost algorithm will select the weak classifier giving the best performance at that round of boosting. The main steps used to implement AdaBoost algorithm are summarized as follows: In this paper, Decision Stump, which is one node of a Decision Tree, is used as a weak classifier in the implementation of the AdaBoost procedure. Figure 6 shows an illustration of AdaBoost classification, where the process is repeated three times corresponding to \( T = 3 \). The AdaBoost algorithm was initially designed to solve binary classification. To solve multi-class problems, we are inspired by the One-Against-All approach used in the multiclass SVMs [35]. Generally speaking, the key idea is to train the first AdaBoost on data from the first class and the rest of the data from the other classes. 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 (as illustrated in Figure 7). The successive

---

Algorithm 1: AdaBoost classifier working principle

**Input:**
- A training set \( S = \{(x_1, y_1), ..., (x_n, y_n)\} \).

**Initialization**
- Define number of iterations \( T \);
- Initialize all weights over training samples \( w_i(l) = \frac{1}{n} \)

**For** \( t = 1, ..., T \) **do**

- Select classifier \( h_t: \mathbb{R}^d \rightarrow \{-1, +1\} \) using distribution \( w_t \)
- Set error \( \varepsilon_t = \sum_{i=1}^{n} w_t(l) \mathbb{I}[h_t \neq h_i(x)] \)
- Choose \( \alpha_t = \frac{1}{2} \ln \frac{1 - \varepsilon_t}{\varepsilon_t} \)
- Update weight distribution over misclassified samples

\[
 w_{t+1}(l) = \frac{w_t(l) \exp(-\alpha_t Y_t h_t(x_l))}{Z_t}
\]

Where \( Z_t = \sum_{i=1}^{n} w_t(l) \exp(-\alpha_t Y_t h_i(x_l)) \) is a normalization factor

**Output**
- Combined classifier: \( h_T(x) = \text{sign} \left( \sum_{t=1}^{T} \alpha_t \times h_t(x) \right) \)
AdaBoost classification outputs represent a binary decision for each action class, hence the naming of cascade classifier [36].

![Image of AdaBoost separation](image)

**V. EXPERIMENTS AND RESULTS**

**A. Data description**

This section reports on the effectiveness of the proposed AdaBoost-based approach for human action recognition. We tested the effectiveness of our approach by using publicly available benchmarks: the University of Rzeszow’s fall detection dataset (URFD) [37] and UMAFD dataset [26]. The URFD dataset consists of 70 videos (30 sequences correspond to fall incidents and 40 sequences represent daily gestures). To simulate daily gestures close to reality and ensure a variety of gestures different scenarios and environments are considered in this database (e.g., bedroom, classroom, and office). The speed of the image flow is 25 frames per second, and with the resolution of $640 \times 480$ pixels. The second dataset, UMAFD, contains 530 video-clips describing activities of daily living (ADL) and falls. The frame rate is 29 frames per second and the resolution is $854 \times 480$ pixels. These video samples described the mobility traces generated by a group of 17 experimental subjects. Each actor performed several predetermined activities, such as walking, squatting, body bending, hopping, climbing, lying down, and sitting. To use this database for posture classification, we annotated the ground truth considering 6 classes: standing, sitting, bending, lying, squatting and kneeling. We selected more than 5000 frames to evaluate the effectiveness of the proposed system. Each selected frame should contain one of the predefined postures. These sample of frames were then split into training and testing sets. To allow a fair and robust evaluation, a 3-fold cross-validation process was employed.

**B. Classification Results**

In this section, we are interested in studying the performance of the proposed human recognition approach. We also compare the classification quality of the proposed approach to that of K-NN, neural network, naïve naïve and SVM classifiers. The parameters of the machine learning approaches studied in this paper are selected during the training phase, corresponding to the best classification rate. The experimental parameters of the machine learning algorithms studied in this paper are presented in Table I.

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>KNN</td>
<td>The number of neighbors was varied from 1 to 30 to choose k=3.</td>
</tr>
<tr>
<td>Naïve Bayes</td>
<td>Gaussian model as distribution model with values of mean and standard deviation are 3.43 and 0.38, respectively.</td>
</tr>
<tr>
<td>Neural network</td>
<td>RBF is selected as kernel, where σ (the width of Gaussian kernel) = 0.125 and C the parameter of the cost function=128.</td>
</tr>
<tr>
<td>SVM</td>
<td>Decision tree is selected as weak classifier with number of boosting iterations $T = 60$.</td>
</tr>
</tbody>
</table>

**TABLE I**

VALUES OF PARAMETERS USED IN THE STUDIED SCHEMES.

In the AdaBoost algorithm, we selected decision tree as a weak classifier and the number of boosting iterations is $T = 60$. The efficiency of the proposed system has been quantified based on confusion matrix, and the performance metrics such as the recognition rate and the kappa coefficient. Table II and Table III illustrate the confusion matrices and the performance metrics for the proposed method applied to URFD and UMAFD respectively. From Table IV, it can be seen that most of the postures are successfully classified and lying positions are correctly identified. This is mainly due to the efficiency of the AdaBoost formalism in discriminating different actions. This means that the AdaBoost classifier can successfully address the problem of human action classification even if it is based on weak classifiers. We also note that the information captured through area ratios allows an appropriate description for most human postures.

**TABLE II**

HUMAN POSTURE CLASSIFICATION RESULTS APPLIED TO URFD DATASET.

<table>
<thead>
<tr>
<th>Classification data</th>
<th>Reference data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standing</td>
<td>Lying</td>
</tr>
<tr>
<td>Standing</td>
<td>95.21</td>
</tr>
<tr>
<td>Lying</td>
<td>0</td>
</tr>
<tr>
<td>Bending</td>
<td>2.87</td>
</tr>
<tr>
<td>Sitting</td>
<td>1.92</td>
</tr>
<tr>
<td>Squatting</td>
<td>0</td>
</tr>
<tr>
<td>Kneeling</td>
<td>0</td>
</tr>
</tbody>
</table>

**Overall accuracy=96.56%; Kappa coefficient=0.96**

In Table IV, we compare the performance of the proposed method with results from some commonly used machine learning algorithms, including K-NN, neural network, naïve Bayes and SVM classifiers. Table IV presents the results of a
comparison when the studied classifiers are applied to URFD dataset. The obtained results showed that the proposed method outperformed the other algorithms used in this study, and exhibited the highest accuracy.

<table>
<thead>
<tr>
<th>Classification data</th>
<th>Standing</th>
<th>Lying</th>
<th>Bending</th>
<th>Sitting</th>
<th>Squatting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standing</td>
<td>92.35</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Lying</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Bending</td>
<td>3.76</td>
<td>0</td>
<td>89.91</td>
<td>7.21</td>
<td>2.58</td>
</tr>
<tr>
<td>Sitting</td>
<td>1.54</td>
<td>0</td>
<td>6.76</td>
<td>90.33</td>
<td>0.42</td>
</tr>
<tr>
<td>Squatting</td>
<td>2.32</td>
<td>0</td>
<td>3.33</td>
<td>2.46</td>
<td>97.00</td>
</tr>
</tbody>
</table>

**Overall accuracy = 93.91%**

**Kappa coefficient = 0.91**

**TABLE III**

**HUMAN POSTURE CLASSIFICATION RESULTS APPLIED ON UMAFD DATASET.**

By comparing results in this study, we noted that the classification efficiency greatly enhanced by using the AdaBoost algorithm. Also, results demonstrate that the AdaBoost classifier, by cascading several weak classifiers (decision tree), achieved a better classification of gestures. In AdaBoost classification, each decision tree classifier is focused only on a part of features which allows an acceptable distinction between classes. This makes it easy to understand and implement. Table VI presents the average processing times in seconds for each classification technique. The processing time corresponds to the time needed for features extraction and classification. Processing time is a helpful tool for comparing the complexity of classifiers. The fact that AdaBoost algorithm is based on simple classifiers (Decision Stump as weak classifiers) makes processing much faster than other studied classifiers (Table VI), and suitable for this type of applications.

<table>
<thead>
<tr>
<th>Method</th>
<th>Average training time (ms)</th>
<th>Average testing time (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>KNN</td>
<td>114</td>
<td>440</td>
</tr>
<tr>
<td>Naïve Bayes</td>
<td>3270</td>
<td>280</td>
</tr>
<tr>
<td>SVM</td>
<td>1310</td>
<td>190</td>
</tr>
<tr>
<td>AdaBoost</td>
<td>680</td>
<td>87</td>
</tr>
</tbody>
</table>

**TABLE VI**

**AVERAGE PROCESSING TIMES (MS) FOR EACH CLASSIFICATION METHOD.**

In summary, it can be noticed that the AdaBoost algorithm based on area ratios as the input features exhibited high efficiency in discriminating between postures. Moreover, the AdaBoost algorithm is less susceptible to the overfitting problem and can be implemented with low computational cost compared to the other studied machine learning algorithms, which make its implementation suitable in real-time application [38].

**VI. CONCLUSION**

In this paper, an effective human action recognition method using video camera monitoring and adapting AdaBoost classifier has been proposed. We demonstrate the effectiveness of the proposed approach using the URFDD dataset and the Universidad de Málaga fall detection dataset. Furthermore, we provided a comparison of the proposed model with other commonly used classification algorithms namely K-NN, neural network, naïve Bayes and SVM and showed that we achieve better results.

A direction for future improvement is to use a camera equipped with infrared or thermal imager which makes human action recognition possible even in dusky environments. It is obvious that in dark environments, thermal or infrared imagers can extract more clearly human silhouettes than RGB cameras. Hence in this case, the proposed human actions recognition strategy can be used in similar manner as in the case of RGB cameras.

The automatic changing of background makes the human action recognition challenging and can generate errors and false classification. We plan also to use automatic background...
updating methods that can be more accurate than a simple average to further enhance the performance of the proposed approach for classifying human activities under a dynamical background.

ACKNOWLEDGMENT

This publication is based upon work supported by the King Abdullah University of Science and Technology (KAUST) Office of Sponsored Research (OSR) under Award No: OSR-2015-CRG4-2582. We are grateful to the two referees, the Associate Editor, and the Editor-in-Chief for their comments.

REFERENCES