

Optimizing Human Action Recognition based on a Cooperative Coevolutionary Algorithm

Alexandros Andre Chaaaraoui^{a,*}, Francisco Flórez-Revuelta^b

^a*Department of Computer Technology, University of Alicante, P.O. Box 99, E-03080, Alicante, Spain*

^b*Faculty of Science, Engineering and Computing, Kingston University, Penrhyn Road, KT1 2EE, Kingston upon Thames, United Kingdom*

Abstract

Vision-based human action recognition is an essential part of human behavior analysis, which is currently in great demand due to its wide area of possible applications. In this paper, an optimization of a human action recognition method based on a cooperative coevolutionary algorithm is proposed. By means of coevolution, three different populations are evolved to obtain the best performing individuals with respect to instance, feature and parameter selection. The fitness function is based on the result of the human action recognition method. Using a multi-view silhouette-based pose representation and a weighted feature fusion scheme, an efficient feature is obtained, which takes into account multiple views and their relevance. Classification is performed by means of a bag of key poses, which represents the most characteristic pose representations, and matching of sequences of key poses. The performed experimentation indicates that not only a considerable performance gain is obtained outperforming the success rates of other state-of-the-art methods, but also the temporal and spatial performance of the algorithm is improved.

*Corresponding author: Alexandros Andre Chaaaraoui, Department of Computer Technology, University of Alicante, P.O. Box 99, E-03080, Alicante, Spain. Phone: +34 965903681, Fax: +34 965909643

Email addresses: alexandros@dtic.ua.es (Alexandros Andre Chaaaraoui), F.Florez@kingston.ac.uk (Francisco Flórez-Revuelta)

URL: <http://www.dtic.ua.es> (Alexandros Andre Chaaaraoui)

Preprint submitted to Engineering Applications of Artificial Intelligence November 3, 2013

Keywords: human action recognition, evolutionary computation, instance selection, feature subset selection, coevolution

1. Introduction

Nowadays, human behavior analysis (HBA) is gaining more and more interest in the field of artificial intelligence and machine learning. Motivated by the wide possible application areas as gaming, natural user interfaces or assistive technologies, just to name a few, great advances have been made in the learning and recognition of human behavior, especially by means of computer vision techniques (Moeslund et al., 2006). For instance, in the case of video surveillance, initially the main goal was to handle person detection, identification (and re-identification) and tracking, whereas activity recognition and scene analysis currently spark the greatest interest, not only of the researchers but also of the industry (Wang, 2013). Specifically, human action recognition (HAR) deals with the lowest level of semantic interpretations of basic human behaviors. For example, motion-based actions as walking, jumping or falling fit into this category. This essential part allows to process further recognition stages. In combination with scene analysis and event detection techniques, complex human activities and long-term behaviors or routines can be recognized (Jiang et al., 2013), for instance, in smart home environments (Rho et al., 2012; Silva et al., 2012; Chaaaraoui et al., 2012b).

In this paper, a state-of-the-art vision-based human action recognition method is used addressing multiple optimization targets. First, we seek the best possible set of instances. Due to different kinds of recording errors, noise and instance- or subject-related peculiarities (as clothes, body build, etc.), not all the available instances of a training set are equally useful. Whereas having more samples is usually valuable to learn the intra-class variance of an action class, this is not the case for random noise appearances and outlier values, which tend to spoil and overfit the learning model (Cano et al., 2005). Furthermore, filtering out the redundant instances and optimizing the set of instances to the smallest one which maintains or improves the initial recognition rate, leads to a significant spatial and temporal improvement. Second, feature subset selection is applied in order to obtain the optimal selection of elements out of a feature vector. Also in this case, redundant or noisy feature elements can be discarded, which benefits the work of the

classifier (Cantú-Paz, 2004). In our case, a human silhouette-based feature is employed whose spatial organization follows a radial fashion. This leads to the natural relation between feature elements and body parts. Certainly, depending on the action to recognize, some body parts may be more relevant than others, and some can be discarded completely. Finally, the optimal values of the algorithm’s parameters are determined in order to achieve the best empirical configuration, i.e. the one that leads to the highest recognition rate. The use of evolutionary algorithms for parameter selection is the most common, since it has been applied for decades (De Jong, 1975).

So as to compute these optimizations in an acceptable amount of time, a cooperative coevolutionary algorithm is proposed. Unsupervised selection of instances, features and parameter values is performed simultaneously by using a coevolutionary algorithm with a cooperative behavior. This choice is motivated by the fact that coevolutionary algorithms make it possible to split the domain of the problem relying on the *divide and conquer* strategy, and tackle each part of the optimization problem with respect to their different solution spaces and data types. Furthermore, the cooperative coevolution allows us to consider the intrinsic dependencies which may exist between optimization goals by using a global fitness function and evaluating the cooperation between populations (Derrac et al., 2012). As section 5 shows, a significant increase in recognition accuracy and a considerable decrease in spatial and temporal complexity are achieved with this proposal, leading to outstanding results on publicly available datasets.

The remainder of this paper is organized as follows: Section 2 summarizes recent and related work in human action recognition and data reduction techniques. A brief definition of a cooperative coevolutionary algorithm is also included. In section 3, the human action recognition method which is our object of optimization is outlined. Section 4 details the coevolutionary algorithm that is proposed for simultaneous instance, feature and parameter selection. Experimental results and a comparison with the state of the art are specified in section 5. Finally, we present conclusions and discussion in section 6.

2. Related Work

In this section, recent and related works in human action recognition and data reduction techniques are summarized. The necessary background on coevolutionary algorithms is also included.

2.1. Human action recognition

Existing human action recognition methods based on vision can be categorized by the visual features they employ in order to classify a specific image or a sequence of frames. These are either *local* (also known as *sparse*) descriptors which describe characteristics of multiple relevant points or areas in the image, or global (also known as *dense*, or *holistic*) representations which encode the image information as a whole. Whereas the former mostly rely on color and gradient-based information in order to detect and describe the points of interest, global features can rely on shape, motion and/or temporal data (Poppe, 2010).

Regarding global representations, several research works rely on human silhouettes (e.g. Bobick and Davis (2001); Blank et al. (2005); Tran and Sorokin (2008); Weinland et al. (2006); Thureau and Hlaváč (2007) and İközler and Duygulu (2007)). Human silhouettes can be obtained based on image processing techniques as background subtraction, human body detection, or using infra-red, laser or depth cameras. Commonly background subtraction is applied to remove the static background from an image and extract the foreground. Then, a blob detector can identify the part of the foreground that corresponds to the human silhouette. This reduces the problem to a single shape-based region of interest. Bobick and Davis (2001) proposed motion history and motion energy images (MHI, MEI), which respectively encode the age and the location of the motion at pixel-level over a sequence of frames. Weinland et al. (2006) extended this technique to a multi-view and viewpoint-independent motion history volume (MHV) by means of invariant motion descriptors in Fourier space. Classification has been performed combining principal component analysis (PCA) and linear discriminant analysis (LDA) for dimensionality reduction, and Mahalanobis distance for feature matching. Radial histograms of the human silhouette and the optical flow of the X and Y axis are employed in Tran and Sorokin (2008). This visual descriptor has successfully been used by other authors as, for example, in Li and Zickler (2012) for cross-view action recognition. İközler and Duygulu (2007) describe the human silhouette based on histograms of oriented rectangular patches extracted over the whole body. Then, different ways of considering the temporal domain are tested. Although best results have been achieved using dynamic time warping, frame-by-frame voting and global histogramming achieved similar results, suggesting that dynamics are not indispensable.

Looking at related optimizations of HAR methods, we find that feature subset selection has been applied previously with success. In Jhuang et al.

(2007), feature subset selection by means of a support vector machine (SVM) is applied to position-invariant spatio-temporal features, resulting in a reduction of 24 times of the number of features. Spatio-temporal interest points are also used by Bregonzio et al. (2010), where the global distribution information of interest points is exploited. Since the feature space dimension is very high, redundant features are eliminated. Feature selection is applied based on the relevance of each feature, i.e. the proportion of inter-class variation with respect to the intra-class variation. Kovashka and Grauman (2010) target to learn the most discriminative shapes of space-time feature neighborhoods. Multiple kernel learning is employed in order to determine the appropriate distance metrics between interest points. Entropy is used as measure of importance in Ikizler et al. (2008), so as to choose the region of the human body where most of the motion occurs. In this way, the feature size of a histogram of orientations of border lines could be reduced by a factor of three.

2.2. Data Reduction Based on Evolutionary Algorithms

As has been briefly introduced in section 1, two of our optimization targets address data reduction. These are to find the best performing selection of instances and feature subset. As stated in Liu and Motoda (2002), this can be seen as selecting rows (training instances) and columns (features) out of the training data. In this sense, a two-fold objective is pursued. First, the recognition rate can be improved by filtering noisy and outlier data (which could lead to overfitting (Wilson and Martinez, 2000)), obtaining a more consistent learning model. Second, execution time can be reduced without compromising the success rate if the redundant training data is ignored. Note at this point that obtaining suboptimal selections in acceptable execution times is pursued, since to assure condition of optimality would require exhaustive search algorithms.

Whereas a solid state of the art exists regarding instance selection (Wilson and Martinez, 2000; Jankowski and Grochowski, 2004; Grochowski and Jankowski, 2004), evolutionary algorithms (EA) for this purpose are still sparingly being used. Cano et al. elaborated a comparison between evolutionary and non-evolutionary instance and feature selection methods, and concluded that the former consistently performed better in both terms of recognition accuracy and spatial and temporal performance. A generational genetic algorithm (GA), a steady-state GA, a heterogeneous recombination and cataclysmic mutation (CHC) adaptive search algorithm, and a

population-based incremental learning specific EA have been included in the comparison (Cano et al., 2003). In García et al. (2008), a memetic algorithm is proposed for instance selection, tackling the problem of selection in large scale databases. A cooperative coevolutionary algorithm is used for instance selection in García-Pedrajas et al. (2010), where the obtained results compared favorably with standard and also recently published state-of-the-art algorithms (see García et al. (2012); Olvera-López et al. (2010) for more details).

Regarding feature subset selection, the usage of evolutionary algorithms goes back a long way (Siedlecki and Sklansky, 1989). Commonly, the approach relies on a binary selection. For example, in the case of a genetic algorithm, an individual is built up by an array of genes which indicate whether or not a specific feature is selected. Then, the feature subset is evaluated either based on intrinsic properties as distance, dependence or consistency (so called *filter* model), or using the actual learning algorithm, where the feature subset is going to be used, as a fitness function (known as the *wrapper* model) (Cantú-Paz, 2004; Casado Yusta, 2009). This last one comes with the disadvantage that each feature subset selection needs to be tested going through a complete classification process. Nonetheless, it also achieves better results (Cantú-Paz, 2004). A third possibility is to embed the feature selection in the construction of the classifier: during training the classifier selects the appropriate features to improve the results (Guyon and Elisseeff, 2003). Besides, as stated by Espejo et al. (2010), the application of genetic programming (GP) for inducing classifiers usually implies a feature selection process which is inherent to the evolution of classifiers.

Finally, some other works apply both instance and feature subset selection simultaneously. Kuncheva and Jain (1999) used a GA for this purpose overcoming the disadvantages of a consecutive approach. A similar method is presented in Ros et al. (2008), where additional heuristics are considered to promote diversity and elitism in the population. A cooperative coevolutionary algorithm is successfully employed in Derrac et al. (2010) on datasets of different data nature. McIntyre and Heywood (2011); Doucette et al. (2012) combined competitive and symbiotic (cooperative) coevolution multi-objective optimisation and GP classifiers. Competition provides a mechanism for scaling to potentially large unbalanced datasets while cooperation allows the decomposition of the training set to improve the results. Feature subset selection is embedded in the GP classifiers.

2.3. Cooperative Coevolutionary Algorithms

A coevolutionary algorithm (CEA) can be defined as one or more EA, in which the fitness value of an individual of one of the populations depends on its relationships to the individuals from the other populations (Wiegand, 2004). In other words, the search problem is divided into sub-problems, where each population handles one of them separately. Nonetheless, the proficiency of the individuals is evaluated in a correlative way (Derrac et al., 2010). Coevolutionary algorithms can be categorized by means of the type of relation between individuals. Whereas *cooperative* CEA reward individuals that work well together, *competitive* CEA follow a predator-prey relationship rewarding those individuals whose opponents perform poorly against them (Wiegand, 2004).

Coevolutionary algorithms are being used successfully in different domains as process planning and scheduling (Kim et al., 2003), multiobjective optimization (Tan et al., 2006) and clustering (Potter and Couldrey, 2010), among others.

3. Human Action Recognition Method

In this section, the human action recognition method is detailed, whose performance is our optimization objective. This multi-view human action recognition method relies on multi-view silhouette-based pose representations, and performs recognition of sequences of key poses. The present proposal builds upon previous contributions (Chaaroui et al., 2012a, 2013). In Chaaroui et al. (2012a), an action learning method based on a bag of key poses is presented, and different types of multi-view fusion are considered. In Chaaroui et al. (2013), the recognition of actions based on sequences of key poses is introduced. In the present work, these proposals are further enhanced with a novel multi-view pose representation and a weighted feature fusion scheme in which the quality of each viewpoint is learned. The complete outline of the resulting method is detailed in the following.

3.1. Multi-view Pose Representation

Our method relies on the input of multiple cameras covering the same field of view. Therefore, a multi-view pose representation is computed based on the single-view pose representations.

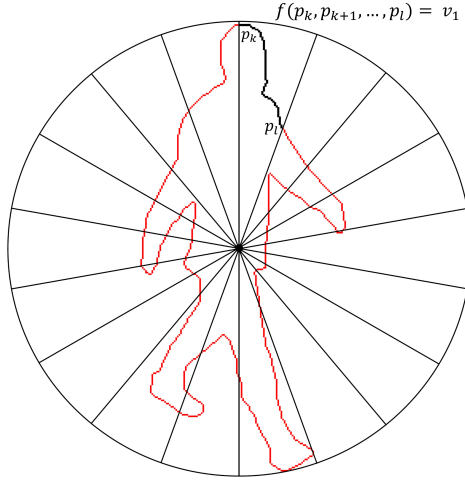


Figure 1: The single-view pose representation is obtained by applying a radial scheme to the contour points of the human silhouette and processing a summary value for each of the radial bins.

3.1.1. Single-view Pose Representation

Initially, we assume that human silhouettes are obtained out of the RGB video frames. These are usually processed by means of background subtraction or human body detection techniques. But it is also possible to obtain them using infra-red, laser or depth cameras. We process these binary segmentations in order to extract the contour points p_1, p_2, \dots, p_n of the human silhouette. Using these points, the pose representation is obtained as follows:

1. The centroid C of the silhouette is computed as:

$$C = (x_c, y_c), \quad (1)$$

$$x_c = \frac{\sum_{i=1}^n x_i}{n}, y_c = \frac{\sum_{i=1}^n y_i}{n}. \quad (2)$$

2. The Euclidean distance between each contour point and the centroid is obtained:

$$d_i = \|C - p_i\|, \quad \forall i \in [1..n]. \quad (3)$$

3. Then, a radial scheme is applied in order to spatially align the parts of the human silhouette with independence from the specific shape and resulting contour length. The radial bin s_i of each contour point

is determined as (for the sake of simplicity $\alpha_i = 0$ is considered as $\alpha_i = 360$):

$$s_i = \left\lceil \frac{S \cdot \alpha_i}{360} \right\rceil, \quad \forall i \in [1 \dots n], \quad (4)$$

$$\alpha_i = \begin{cases} \arccos\left(\frac{y_i - y_c}{d_i}\right) \cdot \frac{180}{\pi} & \text{if } x_i \geq 0, \\ 180 + \arccos\left(\frac{y_i - y_c}{d_i}\right) \cdot \frac{180}{\pi} & \text{otherwise,} \end{cases} \quad (5)$$

where S stands for the number of radial bins.

4. Finally, the contour points of each radial bin are summarized into a single value based on the range of change between their distances to the centroid:

$$v_j = \max(d_k, d_{k+1}, \dots, d_l) - \min(d_k, d_{k+1}, \dots, d_l) \\ / s_k \dots s_l = j \wedge k, l \in [1 \dots n], \quad \forall j \in [1 \dots S]. \quad (6)$$

5. These summary values are normalized and concatenated into the resulting feature vector:

$$\bar{\mathbf{V}} = \bar{v}_1 \parallel \bar{v}_2 \parallel \dots \parallel \bar{v}_S, \quad (7)$$

$$\bar{v}_j = \frac{v_j}{\sum_{j=1}^S v_j}, \quad \forall j \in [1 \dots S]. \quad (8)$$

The shape-based feature vector $\bar{\mathbf{V}}$ has the advantage of presenting a very low dimensionality and a reduced computation time. Figure 1 shows a graphical explanation of the feature extraction process.

3.1.2. Weighted Feature Fusion Scheme

In order to consider multiple views, this method uses a *feature fusion* based approach. In other words, the single-view features are joined together to a multi-view pose representation. Although other strategies have been considered (Charaoui et al., 2012a), this method proved to have the required robustness with an increased number of views.

The sought-after multi-view pose representation is obtained by feature concatenation. Additionally, a weighted feature fusion scheme is proposed in order to consider the quality of each viewpoint. Depending on the action

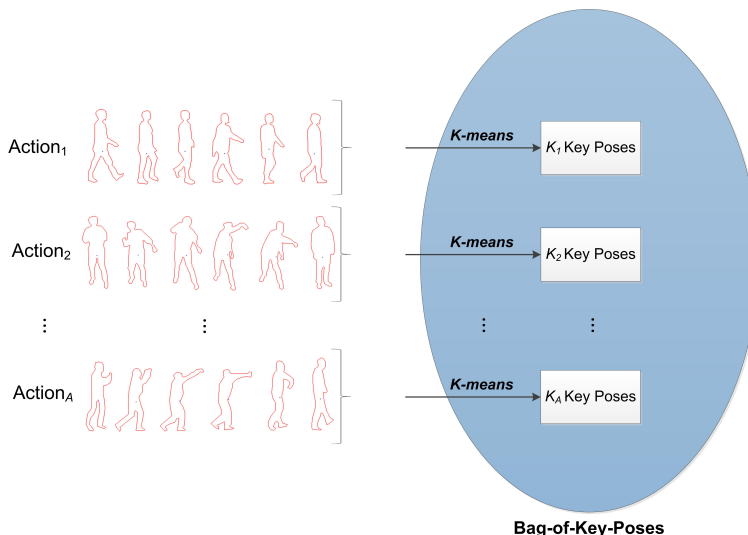


Figure 2: The bag of key poses is learned by clustering the training samples of each action class, and taking the K cluster centers as representatives because these define the most relevant poses.

class, the orientation and the location of the subject, some viewpoints may be more useful than others to recognize what a person is doing. For this reason, camera weights need to be class-specific, and should be obtained for each camera setup.

The camera weights are obtained as follows:

$$w_{m,a} = Test(m)_a, \quad \forall m \in [1...M] \quad \wedge \quad \forall a \in [1...A], \quad (9)$$

where M stands for the number of available camera views and A action classes are being learned. $Test(m)$ returns the per-class recognition results of a single-view test using view m . In this way, the camera weights for each action class are determined based on the success rates of that view. Then, these weights are normalized to unit-sum.

3.2. Bag of key poses

Once the multi-view pose representations are extracted, the most representative and common poses are learned. This way, the characteristic instances of each action class can be retained, and redundant examples can be ignored. These multi-view key poses are obtained using K -means clustering and taking the resulting cluster centers as representatives. The key poses

of each action class are joined together in the same bag of key poses. As it can be seen in figure 2, we take the per class pose representations obtained as detailed in section 3.1, and apply the clustering algorithm in order to obtain K_1, K_2, \dots, K_A key poses. As will be seen in section 4, these are the parameters which are selected by means of the evolution of the parameter population. Further insight about the bag-of-key-poses model can be looked up in previous work (Chaaroui et al., 2012a).

3.3. Sequences of Key Poses

So as to model the temporal relation between key poses, i.e. the typical order and transitions between them, sequences of key poses are built. In this manner, for each of the available training sequences, each pose representation is substituted with its *nearest neighbor* key pose out of the bag of key poses. The successive key poses make up a sequence of key poses $Seq = \{kp_1, kp_2, \dots, kp_t\}$. This step achieves to filter noise and outlier values, due to the transition to the common domain of the bag of key poses.

In order to classify a new action performance, the same steps are taken till an equivalent sequence of key poses is obtained. Then, sequence matching is employed in order to perform recognition. For this purpose, dynamic time warping (DTW) has been chosen, because it inherently supports the alignment of sequences with non-uniform speeds, which is especially needed at action performances. The best match is determined based on the lowest DTW distance $d_{DTW}(Seq, Seq')$:

$$d_{DTW}(Seq, Seq') = dtw(t, u), \quad (10)$$

$$dtw(i, j) = \min \left\{ \begin{array}{l} dtw(i-1, j), \\ dtw(i, j-1), \\ dtw(i-1, j-1) \end{array} \right\} + d(kp_i, kp'_j), \quad (11)$$

where the distance between key poses $d(kp_i, kp'_j)$ takes into account the previously obtained camera weights. For this purpose, the action class a of the known sequence Seq is considered so as to apply the appropriate weights. In other words, we suppose that the current comparison is a correct match, and use the weights that should be used in that case, since these indicate to which degree each view should be considered.

$$d(kp_i, kp'_j) = \sum_{m=0}^M w_{m,a}(\bar{\mathbf{V}}_m - \bar{\mathbf{V}}'_m), \quad (12)$$

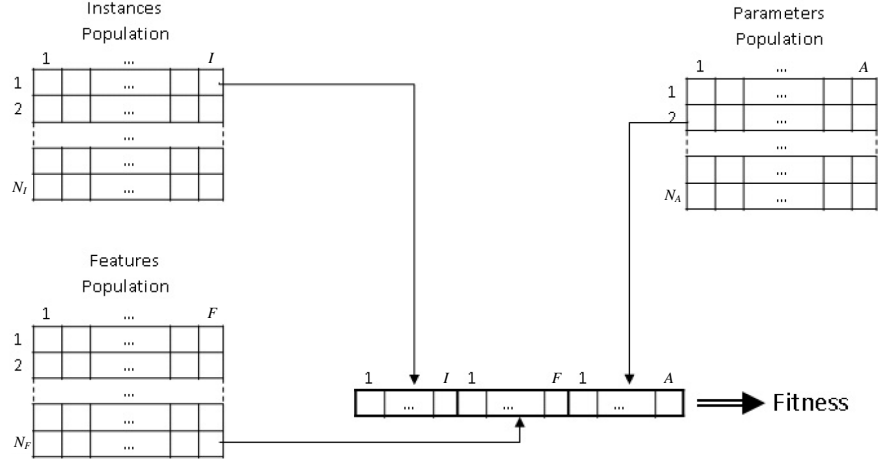


Figure 3: The fitness value is obtained by evaluating the human action recognition method with the configuration indicated as an array based on the concatenation of one individual from each population.

where $kp_i = \bar{\mathbf{V}}_1 \parallel \bar{\mathbf{V}}_2 \parallel \dots \parallel \bar{\mathbf{V}}_M$ and $kp'_j = \bar{\mathbf{V}}'_1 \parallel \bar{\mathbf{V}}'_2 \parallel \dots \parallel \bar{\mathbf{V}}'_M$.

In this way, the label of the best matching training sequence determines the result of the multi-view recognition.

4. Cooperative Coevolutionary Algorithm for Instance, Feature and Parameter Selection

In this section, the proposed cooperative coevolutionary algorithm for the simultaneous selection of instances, features and parameter values is detailed. For this purpose, we will present how the populations and the individuals are defined, and which steps are executed in the coevolution.

4.1. Population Structure

Since our problem of finding an optimal classification configuration is divided in the search of a selection of instances, a feature subset and parameter values, we use these sub-problems to define three populations. Individuals out of each population are combined to build a possible solution that can

be evaluated using the fitness function, which in our case relies on the success rate of the human action recognition algorithm using the configuration determined by the individuals (see figure 3).

4.2. Individuals' Representation

Instance and feature subset selections are regarded as binary selections. This means that a training instance is either used or not during learning (it does not make sense to apply this in the recognition stage), and that a specific feature element of the feature vector is excluded during the whole classification. Therefore, the individuals of these two populations are encoded as boolean arrays, in which each gene indicates whether or not this element is selected. The instance population's individuals have I elements, one for each training instance. And the feature population's individuals have one element for each of the F elements of the feature vector.

Parameter values are more closely related to the used HAR method. In this case, as has been seen in section 3, the method learns a parametrized number of key poses for each action class. Therefore, for A action classes, the same amount of parameters need to be indicated, leading to an individual of A integer values. Specifically, the parameters K_1, K_2, \dots, K_A which indicate the per class number of key poses are learned in order to setup and optimize the method.

4.3. Coevolutionary Algorithm

Algorithm 1 details the specific steps that are executed in the proposed coevolutionary algorithm. Following details should be considered:

1. The individuals that are used for recombination and mutation are selected using the ranking method (Jong, 2006), which means that individuals with better fitness present a higher probability to be chosen.
2. A one-point crossover recombination operator is employed.
3. In the case of the instance and feature populations, the mutation operator switches the boolean value of each gene according to a probability mut_I and mut_F respectively. Whereas in the parameter populations, two possible mutations are employed: 1) applying a random increase or decrease to the current integer value, or 2) resetting the parameter to a normal random value. These mutations are applied with a probability of mut_A .

Algorithm 1 Cooperative coevolutionary algorithm

Initialise randomly three populations for instances, features and parameters with N_I , N_F and N_A individuals respectively

Order each population by descending fitness

repeat

for all populations **do**

for number of new individuals to be created **do**

 ————— Generate a new individual —————

Create one new individual i_1 by recombination

Mutate i_1

 ————— Build a solution —————

Select by ranking two individuals i_2 and i_3 each one of them from one of the other populations

Build a solution i as combination of i_1 , i_2 and i_3

 ————— Calculate and update fitness —————

Calculate $fitness(i)$ as the classification rate

$fitness(i_1) = fitness(i)$

if $fitness(i_2) < fitness(i)$ **then**

$fitness(i_2) = fitness(i)$

end if

if $fitness(i_3) < fitness(i)$ **then**

$fitness(i_3) = fitness(i)$

end if

end for

 ————— Generate next generation's population —————

Order each population by descending fitness

Select next generations population with elitism

end for

until $generations_without_changes > gen_{max}$

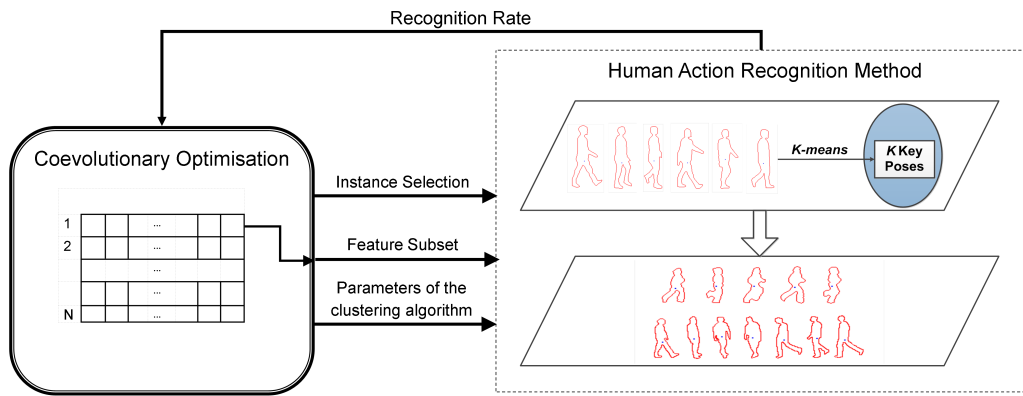


Figure 4: This figure shows how the CEA optimization interacts with the human action recognition method: The individuals are tested using their encoded instance selection, feature subset and clustering parameter values. The employed human action recognition method obtains the multi-view pose representations and learns the bag of key poses out of the training data. Classification is performed matching sequences of key poses. Then, the obtained recognition rate is used as fitness value so as to order the populations and apply elitism.

4. The fitness value is obtained by combining one individual from each population (i_1 , i_2 and i_3) and evaluating the success rate of the human action recognition method with the configuration encoded in the individuals (see section 3). As algorithm 1 shows, whereas i_1 constitutes the new individual, i_2 and i_3 are selected out of their populations based on ranking. As stated in Coello et al. (2006), to choose the best individuals would cause undersampling and excessive greediness in dependant populations.
5. The obtained fitness value is adopted by the new individual i_1 , and by i_2 and i_3 if it improves their current fitness value.
6. When populations are ordered by descending fitness value, we also apply an optimization of spatial and temporal constraints. If two individuals present the same fitness value, the most efficient one is given priority. This means that in the case of the instance and feature populations, the individual with less selected values is favored. In the parameter population, the individual with less accumulated sum is preferred, since a higher value indicates a greater amount of key poses and this results in a more costly classification.

Figure 4 shows how the proposed coevolutionary algorithm employs the wrapper model that has been seen in section 2.2 in order to optimize the recognition of human actions.

5. Experimentation

In this section, the performed experimentation on two publicly available datasets is detailed. These two datasets present notorious differences. Whereas the Weizmann dataset (Blank et al., 2005) constitutes one of the basic and most popular human action recognition datasets and provides automatically obtained binary segmentations, the MuHAVi dataset (Singh et al., 2010) includes multi-view data for up to 14 different action classes and comes along with a subset of manually obtained human silhouettes.

In order to test the accuracy of the recognition of the proposed method, a *leave-one-out* cross validation has been employed. In this type of test, the dataset is divided into several subsets, so as to iterate over these parts using one of them for testing, and all the others for training. This procedure is repeated for all available subsets and the accuracy scores are averaged. In this way, *leave-one-sequence-out* cross validation (LOSO) tests the robustness of the method towards instance-variance, whereas *leave-one-actor-out* cross validation (LOAO) does so for actor-variance, regarding the proficiency of the method in recognizing unseen actors. As in these cross validations training and testing sets change in each fold, an overfitted result of the optimization can be avoided.

The specific configuration details about the performed tests are indicated as follows:

1. The camera weights used at the weighted feature fusion scheme are obtained based on the success rates of the same test that is being performed, but with single-view data.
2. The size of the instance and parameter individuals is given by the specific dataset, i.e. I equals the number of training instances, and A equals the number of action classes to recognize.
3. The number of elements of the feature vector $F = S \times M$, because in the multi-view recognition S feature vector elements are employed for each of the M views.
4. The indicated results have been obtained with populations of ten individuals ($N_I = N_F = N_A = 10$) and a single offspring.

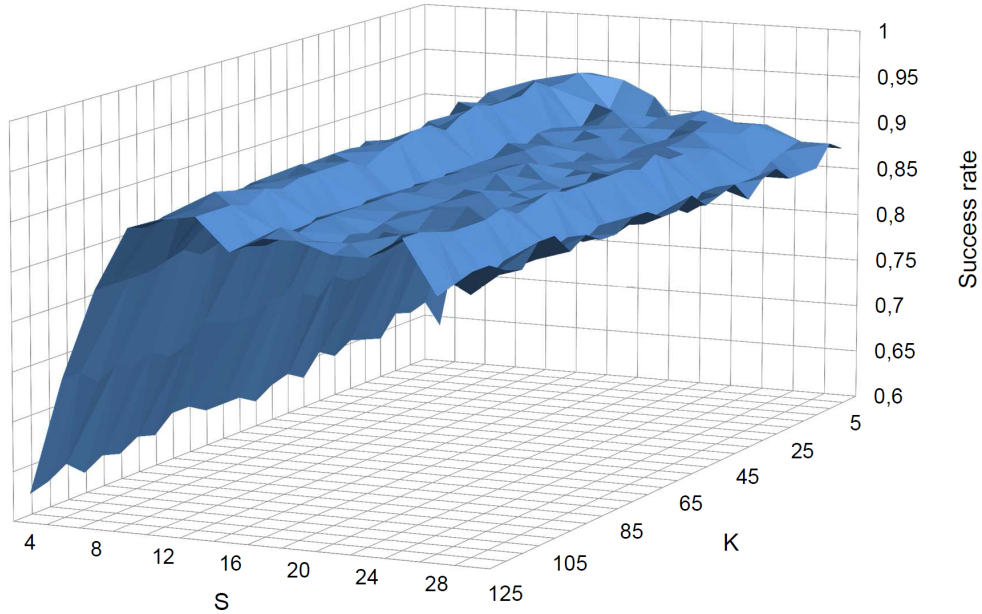


Figure 5: Median recognition rates that have been obtained for $K \in [5, 130]$ and $S \in [4, 30]$ before having applied the optimization (Weizmann LOAO cross validation test). Note that outlier values above or below $1.5 \times IQR$ are not predominant.

5. Regarding the mutation operator, random probability values mut_I , mut_F and mut_P are employed. The two possible mutations of the parameter individuals detailed in section 4.3 are chosen with a 50-50 chance. A range between 5 and 130 key poses is considered for the parameters K_1, K_2, \dots, K_A .
6. We set $gen_{max} = 250$, i.e. the evolution is considered to be stable after 250 generations without changes in its best performing individual.
7. The number of radial bins S is detailed for each test. Its value has been obtained based on a statistical analysis of the classification results of the human action recognition method from section 3. Since this parameter is needed in order to apply the feature selection, tests have been executed before applying any optimization. Besides S , the method also relies on the parameters K_1, K_2, \dots, K_A , which determine the number of key poses per action class. Although these parameters will be optimized later on, they are needed in order to analyze the sensitivity of the algorithm with respect to its parameters. Because

Table 1: Comparison of recognition rates obtained on the Weizmann dataset with other state-of-the-art approaches. In this test 14 radial bins have been used ($S = 14$).

Approach	#Actions	Rate
Tran and Sorokin (2008)	10	100%
Weinland et al. (2010)	9	100%
Naiel et al. (2011)	10	98.9%
Sadek et al. (2012)	10	97.8%
Chaararoui et al. (2013)	9	92.8%
Hernández et al. (2011)	10	90.3%
Our approach before CEA optimization	10	96.8%
Our approach after CEA optimization	10	100%

of the non-deterministic behavior of the K -means algorithm, we run ten tests with each configuration. The median success rates obtained on the Weizmann dataset are shown in figure 5, where the same K value has been established for all the classes. As it can be observed, the parameter K does not present a great influence on the recognition rate. Different action classes involve distinct kinds of motion. For some actions, more or less key poses may be required in order to capture the relevant areas of the feature space. Therefore, choosing the same value for all the action classes will favour some and hinder others, leading for this reason to similar results in average. However, it can be seen that the parameter S has a direct impact on the recognition rate. In order to obtain reproducible results, we chose the value of S based on the highest median success rate, which has been obtained with $S = 14$ in the case of the Weizmann dataset (94.6%). If multiple configurations return the same result, the lowest value will be chosen, as this reduces the feature size and the computational cost of the classification.

5.1. Benchmarks

5.1.1. Weizmann Dataset

The Weizmann dataset (Blank et al., 2005) includes 93 sequences of ten different action classes which have been performed by nine actors. Even if it is intended for single-view human action recognition only, it is commonly used as baseline benchmark. Table 1 shows a comparison of our results, before and after applying the proposed CEA optimization, and the ones that can be found among the state of the art. Several works excluded the *skip*

Table 2: Comparison of recognition rates obtained on the MuHAVi-14 dataset with other state-of-the-art approaches. In this test 12 and 10 radial bins have been employed respectively in the LOSO and LOAO cross validations.

Approach	LOSO	LOAO
Singh et al. (2010)	82.4%	61.8%
Cheema et al. (2011)	86.0%	73.5%
Eweiwi et al. (2011)	91.9%	77.9%
Chaaroui et al. (2012a)	94.1%	86.8%
Our approach before CEA optimization	98.5%	94.1%
Our approach after CEA optimization	100%	100%

Table 3: Comparison of recognition rates obtained on the MuHAVi-8 dataset with other state-of-the-art approaches. In this test 12 and 18 radial bins have been employed respectively in the LOSO and LOAO cross validations.

Approach	LOSO	LOAO
Cheema et al. (2011)	95.6%	83.1%
Singh et al. (2010)	97.8%	76.4%
Martínez-Contreras et al. (2009)	98.4%	-
Eweiwi et al. (2011)	98.5%	85.3%
Chaaroui et al. (2012a)	98.5%	95.6%
Our approach before/after CEA optimization	100%	100%

action because it tends to decrease the overall recognition rate, due to its inter-class similarity. Therefore, we indicated the number of actions used in each test. All of them report results for the LOAO cross validation.

It can be observed that the presented method of section 3 achieves a high recognition rate, and that by means of the applied CEA optimization of both learning and classification stages, perfect recognition is reached.

5.1.2. MuHAVi Dataset

The MuHAVi dataset (Singh et al., 2010) is a more extensive multi-view benchmark for human action recognition. In MuHAVi-MAS, a manually annotated subset of two views and either 14 (MuHAVi-14) or 8 (MuHAVi-8) action classes is provided. In the 136 available sequences, two actors performed these actions up to four times. In difference to MuHAVi-14, in MuHAVi-8 the direction in which an action is performed is ignored. This means that although fewer action classes need to be recognized, these are more difficult to learn since they present more differences.

Table 4: In this table the values of the individuals with the highest fitness value of each of the run tests are shown.

Dataset	Test	Instances	Features	Parameters
Weizmann	LOAO	71/93	11110110110111	72, 37, 25, 22, 19, 78, 25, 47, 55, 106
MuHAVi-14	LOSO	81/136	11101110010111 0111011111	6, 5, 22, 31, 5, 79, 5, 5, 5, 5, 5, 5, 5
MuHAVi-14	LOAO	99/136	10101010100111 111010	4, 4, 10, 112, 14, 29, 6, 54, 5, 95, 23, 27, 6, 20
MuHAVi-8	LOSO	87/136	00010001101101 0110110010	6, 6, 6, 6, 6, 6, 6, 6
MuHAVi-8	LOAO	77/136	10100011010101 10111011101111 001011100	22, 38, 21, 27, 11, 40, 32, 75

Table 2 and 3 show the obtained results. As is usual for this dataset, we applied both the LOSO and LOAO cross validations. In this sense, whereas LOSO is used to provide a measure for the overall recognition performance, LOAO is used to show the robustness of the method to specific actor-related differences, as clothes, body build or ways to perform an action. Regarding the state-of-the-art recognition rates on this dataset, we can observe that lower results are reported in the case of the LOAO cross validation. Our method not only outperforms all the available recognition rates on both versions of the dataset, but it also reaches perfect recognition on all four tests with the proposed optimization based on a coevolutionary algorithm. To the best of our knowledge, these are the highest results reported so far on this dataset.

5.2. Optimization Results

In table 4, the results of the proposed optimization are shown. For each of the applied tests, the values of the best performing individuals from the three populations are detailed. The *instances* column shows the number of training instances that have been selected out of the available ones, and with which the highest recognition results have been obtained. In the case of the *features* column, the feature subset selection of the multi-view pose representation is detailed. Finally, the number of key poses used to represent the action classes is indicated in the *parameters* column.

In average, the coevolutionary algorithm reduced the required training instances to $\sim 66\%$. The feature subset used during learning and classification has been reduced by $\sim 36\%$. Best results have been obtained representing each action class with ~ 26 key poses in average.

Table 5: Selection of instances in terms of action classes and actors obtained for the LOAO cross validation test on the Weizmann dataset. Note that normally each actor performed each action once, but *Lena* performed twice *run*, *skip* and *walk*.

Actor	<i>Bend</i>	<i>Jack</i>	<i>Jump</i>	<i>Pjump</i>	<i>Run</i>	<i>Side</i>	<i>Skip</i>	<i>Walk</i>	<i>Wave1</i>	<i>Wave2</i>	Total
Daria	1	1	1	0	1	1	0	1	0	1	7/10
Denis	1	1	1	1	1	0	1	1	1	1	9/10
Eli	1	1	1	1	1	0	0	1	1	1	8/10
Ido	0	1	1	0	1	1	1	1	1	1	8/10
Ira	1	1	0	1	1	1	1	1	0	1	8/10
Lena	1	1	1	1	1/2	0	2/2	2/2	0	1	10/13
Lyova	1	1	1	0	1	1	1	1	1	1	9/10
Moshe	1	1	0	1	0	1	1	0	0	1	6/10
Shahar	1	0	1	1	0	1	0	1	1	0	6/10
Total	8/9	8/9	7/9	6/9	7/10	6/9	7/10	9/10	5/9	8/9	71/93

With the purpose of providing further insight about how the obtained results correspond to the data, we analyzed the Weizmann LOAO cross validation test. In table 5, the resulting selection of instances is given in terms of both actors and action classes. It can be observed that the sequences of actors as *Moshe* and *Shahar* got selected less, which could be related to the way they performed some of the actions. For instance, the *run* sequence from *Shahar* is performed from left to right, whereas most of the others are performed from right to left. Regarding the action-wise selection, we can observe that a fairly similar distribution of actions has been selected, since samples from all actions are necessary in order to recognize them. In the case of *wave1*, less samples are necessary because all of them are performed with the right hand and mostly only the noise of the background segmentation differs between the samples.

Applying a similar analysis to the obtained feature subset, in figure 6, it can be seen which feature elements of the radial scheme have been discarded here due to redundant or noisy components.

Finally, regarding the selection of parameter values, note that the data given in table 4 follows the alphabetical order of the action classes. In this case, significant differences can be observed regarding the different number of key poses that were necessary to model the representative poses of action classes as *run* ($K = 19$) and *wave2* ($K = 106$) and obtain the best result.

To illustrate the performance gain achieved with this optimization, we performed a temporal evaluation on the Weizmann dataset. Our proposal has been implemented using the .NET Framework and the OpenCV library (Bradski, 2000). Tests have been performed on a standard PC with an Intel Core 2 Duo CPU at 3 GHz and 4 GB of RAM running Windows 7 x64, without applying any hardware-related optimizations. The learning stage of the algorithm is executed in 0.72s, whereas the testing stage requires 3.29s for the whole dataset. Applying the obtained configuration, an improvement of respectively $\sim 41\%$ and $\sim 33\%$ can be observed. The final recognition rate considering the binary silhouette images as input is 210 frames per second (FPS).

6. Discussion and future work

In this paper, a human action recognition optimization based on a cooperative coevolutionary algorithm is presented. By means of evolutionary search, the redundant or noisy training instances which confuse or unnecessarily extend the learning process are filtered. Similarly, the feature subset which includes the relevant parts of the human body in order to recognize human actions is selected considering multiple views. Last but not least, the appropriate number of key poses is sought in order to represent the different poses involved in each action class. This configuration is employed in a human action recognition algorithm, which relies on a multi-view silhouette-based pose representation and a weighted feature fusion scheme, and performs ac-

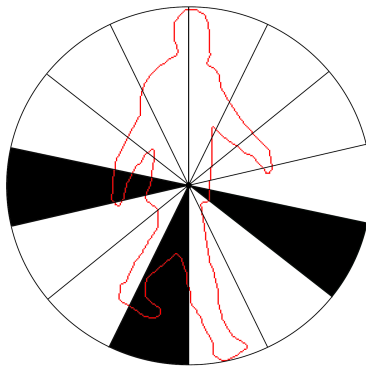


Figure 6: Feature selection that has been obtained for the Weizmann dataset. Discarded elements are shaded in black.

tion recognition by means of matching sequences of key poses.

Results obtained during the experimentation have shown that our proposal achieves to improve the initial recognition rates, reaching perfect recognition for all the applied benchmarks, and considerably outperforming the available rates of the MuHAVi dataset. Furthermore, the execution time and memory needs of the learning algorithm are reduced, since the evolutionary algorithm implicitly prioritizes the more efficient configurations.

In conclusion, the presented work achieves to successfully optimize both the recognition, and the temporal and spatial performance of human action recognition. Since it only has to be applied as a pre-classification stage, it is also suitable for online human action recognition, where the obtained configuration can be applied.

In future works, further experimentation on larger and more complex datasets is going to be performed in order to analyze which margins of optimization can be reached. So as to find the *optimal* configuration of a HAR method intended to perform online in a real world application, a compound of tests could be used assigning the appropriate weights to consider the specific needs of the system. Furthermore, if so required by the application, not only the recognition rate, but also the execution time could be used as optimization function.

Acknowledgements.

This work has been partially supported by the European Commission under project “caring4U - A study on people activity in private spaces: towards a multisensor network that meets privacy requirements” (PIEF-GA-2010-274649) and by the Spanish Ministry of Science and Innovation under project “Sistema de visión para la monitorización de la actividad de la vida diaria en el hogar” (TIN2010-20510-C04-02). Alexandros Andre Chaaaraoui acknowledges financial support by the Conselleria d’Educació, Formació i Ocupació of the Generalitat Valenciana (fellowship ACIF/2011/160). The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript. We sincerely thank the reviewers for their constructive and insightful suggestions that have helped to improve the quality of this paper.

This article was originally published in *Chaaaraoui, A.A., Flrez-Revuelta, F., 2013. Optimizing Human Action Recognition Based on a Cooperative Coevolutionary Algorithm. Engineering Applications of Artificial Intelligence. Advances in Evolutionary Optimization Based Image Processing. DOI 10.1016/j.engappai.2013.10.003.*

References

Blank, M., Gorelick, L., Shechtman, E., Irani, M., Basri, R., 2005. Actions as space-time shapes, in: Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference

- on, pp. 1395–1402 Vol. 2.
- Bobick, A., Davis, J., 2001. The recognition of human movement using temporal templates. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 23, 257–267.
- Bradski, G., 2000. The OpenCV Library. *Dr. Dobb’s Journal of Software Tools* .
- Bregonzio, M., Li, J., Gong, S., Xiang, T., 2010. Discriminative topics modelling for action feature selection and recognition, in: *Proceedings of the British Machine Vision Conference, BMVA Press*. pp. 8.1–8.11. Doi:10.5244/C.24.8.
- Cano, J., Herrera, F., Lozano, M., 2003. Using evolutionary algorithms as instance selection for data reduction in KDD: an experimental study. *IEEE Transactions on Evolutionary Computation* 7, 561–575.
- Cano, J., Herrera, F., Lozano, M., 2005. A study on the combination of evolutionary algorithms and stratified strategies for training set selection in data mining, in: Hoffmann, F., Kppen, M., Klawonn, F., Roy, R. (Eds.), *Soft Computing: Methodologies and Applications*. Springer Berlin Heidelberg. volume 32 of *Advances in Soft Computing*, pp. 271–284.
- Cantú-Paz, E., 2004. Feature subset selection, class separability, and genetic algorithms, in: Deb, K. (Ed.), *Genetic and Evolutionary Computation GECCO 2004*. Springer Berlin / Heidelberg. volume 3102 of *Lecture Notes in Computer Science*, pp. 959–970.
- Casado Yusta, S., 2009. Different metaheuristic strategies to solve the feature selection problem. *Pattern Recognition Letters* 30, 525–534.
- Chaaroui, A.A., Climent-Pérez, P., Flórez-Revuelta, F., 2012a. An efficient approach for multi-view human action recognition based on bag-of-key-poses, in: Salah, A.A., Ruiz-del Solar, J., Meriçli, c., Oudeyer, P.Y. (Eds.), *Human Behavior Understanding*. Springer Berlin Heidelberg. volume 7559 of *Lecture Notes in Computer Science*, pp. 29–40.
- Chaaroui, A.A., Climent-Pérez, P., Flórez-Revuelta, F., 2012b. A review on vision techniques applied to human behaviour analysis for ambient-assisted living. *Expert Systems with Applications* 39, 10873–10888.
- Chaaroui, A.A., Climent-Pérez, P., Flórez-Revuelta, F., 2013. Silhouette-based human action recognition using sequences of key poses. *Pattern Recognition Letters* , -DOI 10.1016/j.patrec.2013.01.021.
- Cheema, S., Eweiwi, A., Thureau, C., Bauckhage, C., 2011. Action recognition by learning discriminative key poses, in: *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pp. 1302–1309.

- Coello, C.A.C., Lamont, G.B., Veldhuizen, D.A.V., 2006. *Evolutionary Algorithms for Solving Multi-Objective Problems (Genetic and Evolutionary Computation)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA.
- De Jong, K.A., 1975. *An Analysis of the behavior of a class of genetic adaptive systems*. Doctoral dissertation. University of Michigan.
- Derrac, J., García, S., Herrera, F., 2010. IFS-CoCo: Instance and feature selection based on cooperative coevolution with nearest neighbor rule. *Pattern Recognition* 43, 2082 – 2105.
- Derrac, J., Triguero, I., Garca, S., Herrera, F., 2012. A co-evolutionary framework for nearest neighbor enhancement: Combining instance and feature weighting with instance selection, in: Corchado, E., Snel, V., Abraham, A., Woniak, M., Graa, M., Cho, S.B. (Eds.), *Hybrid Artificial Intelligent Systems*. Springer Berlin Heidelberg. volume 7209 of *Lecture Notes in Computer Science*, pp. 176–187.
- Doucette, J., McIntyre, A., Lichodziejewski, P., Heywood, M., 2012. Symbiotic coevolutionary genetic programming: a benchmarking study under large attribute spaces. *Genetic Programming and Evolvable Machines* 13, 71–101.
- Espejo, P., Ventura, S., Herrera, F., 2010. A survey on the application of genetic programming to classification. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on* 40, 121–144.
- Eweiwi, A., Cheema, S., Thureau, C., Bauckhage, C., 2011. Temporal key poses for human action recognition, in: *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pp. 1310–1317.
- García, S., Cano, J.R., Herrera, F., 2008. A memetic algorithm for evolutionary prototype selection: A scaling up approach. *Pattern Recognition* 41, 2693 – 2709.
- García, S., Derrac, J., Cano, J., Herrera, F., 2012. Prototype selection for nearest neighbor classification: Taxonomy and empirical study. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 34, 417–435.
- García-Pedrajas, N., Romero del Castillo, J.A., Ortiz-Boyer, D., 2010. A cooperative coevolutionary algorithm for instance selection for instance-based learning. *Machine Learning* 78, 381–420.
- Grochowski, M., Jankowski, N., 2004. Comparison of Instance Selection Algorithms II. Results and Comments, in: Rutkowski, L., Siekmann, J., Tadeusiewicz, R., Zadeh, L. (Eds.), *Artificial Intelligence and Soft Computing - ICAISC 2004*. Springer Berlin Heidelberg. volume 3070 of *Lecture Notes in Computer Science*, pp. 580–585.
- Guyon, I., Elisseeff, A., 2003. An introduction to variable and feature selection. *J. Mach. Learn. Res.* 3, 1157–1182.

- Hernández, J., Montemayor, A., José Pantrigo, J., Sánchez, A., 2011. Human action recognition based on tracking features, in: Ferrández, J., Álvarez Sánchez, J., de la Paz, F., Toledo, F. (Eds.), *Foundations on Natural and Artificial Computation*. Springer Berlin / Heidelberg. volume 6686 of *Lecture Notes in Computer Science*, pp. 471–480.
- Ikizler, N., Cinbis, R., Duygulu, P., 2008. Human action recognition with line and flow histograms, in: *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, pp. 1–4.
- Ikizler, N., Duygulu, P., 2007. Human action recognition using distribution of oriented rectangular patches, in: Elgammal, A., Rosenhahn, B., Klette, R. (Eds.), *Human Motion Understanding, Modeling, Capture and Animation*. Springer Berlin / Heidelberg. volume 4814 of *Lecture Notes in Computer Science*, pp. 271–284.
- Jankowski, N., Grochowski, M., 2004. Comparison of Instances Seletion Algorithms I. Algorithms Survey, in: Rutkowski, L., Siekmann, J., Tadeusiewicz, R., Zadeh, L. (Eds.), *Artificial Intelligence and Soft Computing - ICAISC 2004*. Springer Berlin Heidelberg. volume 3070 of *Lecture Notes in Computer Science*, pp. 598–603.
- Jhuang, H., Serre, T., Wolf, L., Poggio, T., 2007. A biologically inspired system for action recognition, in: *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pp. 1–8.
- Jiang, Y.G., Bhattacharya, S., Chang, S.F., Shah, M., 2013. High-level event recognition in unconstrained videos. *International Journal of Multimedia Information Retrieval* 2, 73–101.
- Jong, K.A.D., 2006. *Evolutionary computation - a unified approach*. MIT Press.
- Kim, Y.K., Park, K., Ko, J., 2003. A symbiotic evolutionary algorithm for the integration of process planning and job shop scheduling. *Computers & Operations Research* 30, 1151–1171.
- Kovashka, A., Grauman, K., 2010. Learning a hierarchy of discriminative space-time neighborhood features for human action recognition, in: *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pp. 2046–2053.
- Kuncheva, L.I., Jain, L.C., 1999. Nearest neighbor classifier: Simultaneous editing and feature selection. *Pattern Recognition Letters* 20, 1149–1156.
- Li, R., Zickler, T., 2012. Discriminative virtual views for cross-view action recognition, in: *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pp. 2855–2862.
- Liu, H., Motoda, H., 2002. On issues of instance selection. *Data Min. Knowl. Discov.* 6, 115–130.

- Martínez-Contreras, F., Orrite-Urunuela, C., Herrero-Jaraba, E., Ragheb, H., Velastin, S., 2009. Recognizing human actions using silhouette-based HMM, in: *Advanced Video and Signal Based Surveillance, 2009. AVSS '09. Sixth IEEE International Conference on*, pp. 43–48.
- McIntyre, A.R., Heywood, M.I., 2011. Classification as clustering: A pareto cooperative-competitive gp approach. *Evol. Comput.* 19, 137–166.
- Moeslund, T.B., Hilton, A., Krüger, V., 2006. A survey of advances in vision-based human motion capture and analysis. *Comput. Vis. Image Underst.* 104, 90–126.
- Naiel, M., Abdelwahab, M., El-Saban, M., 2011. Multi-view human action recognition system employing 2DPCA, in: *Applications of Computer Vision (WACV), 2011 IEEE Workshop on*, pp. 270–275.
- Olvera-López, J., Carrasco-Ochoa, J., Martínez-Trinidad, J., Kittler, J., 2010. A review of instance selection methods. *Artificial Intelligence Review* 34, 133–143.
- Poppe, R., 2010. A survey on vision-based human action recognition. *Image and Vision Computing* 28, 976 – 990.
- Potter, M.A., Couldrey, C., 2010. A cooperative coevolutionary approach to partitional clustering, in: Schaefer, R., Cotta, C., Kołodziej, J., Rudolph, G. (Eds.), *Parallel Problem Solving from Nature, PPSN XI. Springer Berlin Heidelberg. volume 6238 of Lecture Notes in Computer Science*, pp. 374–383.
- Rho, S., Min, G., Chen, W., 2012. Advanced issues in artificial intelligence and pattern recognition for intelligent surveillance system in smart home environment. *Engineering Applications of Artificial Intelligence* 25, 1299 – 1300.
- Ros, F., Guillaume, S., Pintore, M., Chrtien, J., 2008. Hybrid genetic algorithm for dual selection. *Pattern Analysis and Applications* 11, 179–198.
- Sadek, S., Al-Hamadi, A., Michaelis, B., Sayed, U., 2012. A fast statistical approach for human activity recognition. *International Journal of Intelligence Science* 2, 9–15.
- Siedlecki, W., Sklansky, J., 1989. A note on genetic algorithms for large-scale feature selection. *Pattern Recognition Letters* 10, 335 – 347.
- Silva, L.C.D., Morikawa, C., Petra, I.M., 2012. State of the art of smart homes. *Engineering Applications of Artificial Intelligence* 25, 1313 – 1321. *Advanced issues in Artificial Intelligence and Pattern Recognition for Intelligent Surveillance System in Smart Home Environment.*
- Singh, S., Velastin, S., Ragheb, H., 2010. Muhavi: A multicamera human action video dataset for the evaluation of action recognition methods, in: *Advanced Video and Signal Based Surveillance (AVSS), 2010 Seventh IEEE International Conference on*, pp. 48–55.

- Tan, K., Yang, Y., Goh, C., 2006. A distributed cooperative coevolutionary algorithm for multiobjective optimization. *IEEE Transactions on Evolutionary Computation* 10, 527–549.
- Thureau, C., Hlaváč, V., 2007. n -grams of action primitives for recognizing human behavior, in: Kropatsch, W., Kampel, M., Hanbury, A. (Eds.), *Computer Analysis of Images and Patterns*. Springer Berlin / Heidelberg. volume 4673 of *Lecture Notes in Computer Science*, pp. 93–100.
- Tran, D., Sorokin, A., 2008. Human activity recognition with metric learning, in: Forsyth, D., Torr, P., Zisserman, A. (Eds.), *Computer Vision - ECCV 2008*. Springer Berlin Heidelberg. volume 5302 of *Lecture Notes in Computer Science*, pp. 548–561.
- Wang, X., 2013. Intelligent multi-camera video surveillance: A review. *Pattern Recognition Letters* 34, 3 – 19. Extracting Semantics from Multi-Spectrum Video.
- Weinland, D., Özuysal, M., Fua, P., 2010. Making action recognition robust to occlusions and viewpoint changes, in: Daniilidis, K., Maragos, P., Paragios, N. (Eds.), *Computer Vision - ECCV 2010*. Springer Berlin / Heidelberg. volume 6313 of *Lecture Notes in Computer Science*, pp. 635–648.
- Weinland, D., Ronfard, R., Boyer, E., 2006. Free viewpoint action recognition using motion history volumes. *Comput. Vis. Image Underst.* 104, 249–257.
- Wiegand, R.P., 2004. An analysis of cooperative coevolutionary algorithms. Ph.D. thesis. George Mason University. Fairfax, VA, USA.
- Wilson, D., Martinez, T., 2000. Reduction techniques for instance-based learning algorithms. *Machine Learning* 38, 257–286.