CGA: A New Feature Selection Model for Visual Human Action Recognition

Ritam Guha
Jadavpur University https://orcid.org/0000-0002-1375-777X

Ali Hussain Kha
Jadavpur University

Pawan Kumar Singh (✉ pawan.abc333@gmail.com)
Jadavpur University

Ram Sarkar
Jadavpur University https://orcid.org/0000-0001-8813-4086

Research Article

Keywords: Human Action Recognition, Co-operative Genetic Algorithm, Feature selection, Coalition game, Pearson Correlation Coefficient, Weizmann, KTH, UCF11, HMDB51, UCI HAR

Posted Date: May 12th, 2020

DOI: https://doi.org/10.21203/rs.3.rs-28157/v1

Version of Record: A version of this preprint was published at Neural Computing and Applications on September 8th, 2020. See the published version at https://doi.org/10.1007/s00521-020-05297-5.

CGA: A New Feature Selection Model for Visual Human Action Recognition

Abstract

Recognition of human actions from visual contents is a budding field of computer vision and image understanding. The problem with such recognition system is the huge dimensions of the feature vectors. Many of these features are irrelevant to the classification mechanism. For this reason, in this paper, we propose a novel Feature Selection (FS) model called Co-operative Genetic Algorithm (CGA) to select some of the most important and discriminating features from the entire feature set to improve the classification accuracy as well as the time requirement of the activity recognition mechanism. In CGA, we have made an effort to embed the concepts of co-operative game theory in GA to create a both-way reinforcement mechanism to improve the solution of the FS model. The proposed FS model is tested on four benchmark video datasets such as Weizmann, KTH, UCF11, HMDB51 and two sensor-based UCI HAR datasets. The experiments are conducted using four state-of-the-art feature descriptors, namely HOG, GLCM, SURF, and GIST. It is found that there is a significant improvement in the overall classification accuracy while considering nearly 50 to 60% of the original feature vector.

Keywords: Human Action Recognition, Co-operative Genetic Algorithm, Feature selection, Coalition game, Pearson Correlation Coefficient, Weizmann, KTH, UCF11, HMDB51, UCI HAR.

1. Introduction

Human Action Recognition (HAR) plays an essential role in human-to-human interaction and many interpersonal relations by providing vital information about the identity of a person, their personality and psychological state, which are generally challenging to extract [1]. The ability to automatically recognizing human's activities is one of the leading research fields of computer vision and machine learning. In the field of artificial intelligence, machine learning, and deep learning, researches aiming at understanding human actions received tremendous attention and are monotonically increasing since decades [2, 3]. This, in turn, results in a plethora of HAR techniques proposed by various researchers as well as evaluated on different benchmark datasets containing still images, video sequences, collected from accelerometer, smartwatch sensors, gyroscope sensors, gravity sensors and also by using complete length of motion curves by tracking the optical flow features.

During classification, mainly two questions arise: "*What is the action?*" (i.e., what is to be recognized) and "*Which part in the video*?" (i.e., the localization problem) [4]. It is essential to determine the kinetics of a person while classifying a particular action to make it easy for a computer to recognize the target action efficiently. Various human activities, which really arise in a natural manner such as '*Walking,*' '*Running,*' are much tricky to recognize because of the nearly same type of actions performed by a subject. HAR is still considered a challenging task when it comes to do the classification using benchmark datasets containing video sequences of variable length and resolution. It has already been mentioned that several methods have been proposed using machine learning approaches on various datasets which do not show satisfactory performances to a certain extent. However, a better algorithmic approach and further optimization are still required to be incorporated in HAR system.

A recent survey [5] published in the year 2019 depicts that a lot of researchers have focussed on the development of hand-shaped action features for solving the problem of visual HAR. In visual HAR method, the features are generally extracted from different video classes (exhibiting different human actions. Since a video consists of a finite number of frames, it may sometimes happen that the features are extracted from such frame(s) where either no human action is performed, or the human action is already completed. Moreover, in motion based human actions like "*Walking*," "*Jogging,*" or "*Running*," it is also observed that the subject is not present in the last few frames of the respective videos. In these cases, the redundant features extracted from such frames may mislead the classification process. Again, since the features are extracted from continuous video frames, the size of the feature vector increases monotonically. An alternative solution for this problem is to design a new FS model which will select the most significant feature subset among the original feature set while ignoring the redundant ones. This will, in turn, help to increase the overall recognition accuracy as well as reduce the time complexity of the entire HAR procedure.

The process of FS progresses by introducing candidate solutions, evaluating them, and improving them over successive iterations. Depending on the criteria of evaluation. FS is broadly divided into three separate categories namely filter [6–8], wrapper [9–11] and embedded [12–14] models. Filter methods use the statistical characteristics and intrinsic properties of features to evaluate candidate solutions, whereas wrapper methods take help of a learning algorithm (a classifier) to evaluate the solutions at every iteration. Embedded models, however, take advantages of both filter and wrapper methods to create more robust solutions. Although filter methods may take less time for generating solutions, solutions produced by wrapper models are of higher quality.

In this paper, a new wrapper FS procedure named Co-operative Genetic Algorithm (CGA) is designed for the feature dimensionality reduction from a combination of four state-of-the-art feature descriptors; namely, Histograms of Oriented Gradients (HOG) [15], Gray Level Co-occurrence Matrix (GLCM) [16], Speeded Up Robust Features (SURF) [17] and GIST [18]. These feature descriptors have been successfully applied for solving typical pattern recognition problems. In the present work, we have considered them for extracting feature vectors which will serve as an input to our proposed FS model. These feature descriptors are applied on four benchmark 2D RGB video datasets having different number of action classes such as Weizmann [19], KTH [20], UCF11 [21], HMDB51 [22]. These datasets are challenging because they employ complex motions, varied camera angles, and also contain camera motion. In order to test the robustness of the proposed FS model, we have additionally considered two Smartphone sensor-based datasets from UCI machine learning repository [23, 24].

CGA overcomes some major drawbacks present in coalition or co-operative game and makes it applicable to FS. Our main contributions to this paper are mentioned below:

1) Formation of a new FS procedure where the concepts of coalition game and GA is amalgamated for the first time (to the best of our knowledge), thereby making coalition game practically applicable to the domain of FS.

2) Use of Mutual Information (MI) to guide the mutation process in GA (which is otherwise done in a random manner). The new version of GA is called Enhanced GA or EGA.

3) Proposal of a new fitness measure for FS having three different components, namely classification accuracy, number of selected features, and Shapley value-based score to perform FS in a multi-objective fashion.

4) Application of the proposed FS method over some standard feature vectors extracted from various HAR datasets.

The rest of the paper is organized as follows: Section 2 gives a brief overview of the state-of-the-art machine learning based models used for HAR as well as some FS methods used till date. Section 3 reviews some of the preliminaries used for defining the CGA model, whereas Section

4 presents the detailed procedure of our proposed CGA model. Section 5 describes the feature descriptors briefly, whereas the benchmark action datasets used for evaluation of the proposed model, along with the detailed experimental results are shown in Section 6. Finally, some concluding remarks and future extensions of the current work are provided in Section 7.

2. Previous Work

In spite of the fact that a significant amount of advancements has already been done in the field of computer vision, researchers still tends to rely on the traditional pattern recognition approaches for finding the solution to HAR problem. Hand-shaped and logical methodologies are used for understanding the actions performed by the humans. Some of the works done for visual HAR are described in brief.

The work described in [25] by Luvizon *et al.* proposes 2D motion templates based on Motion History Image (MHI) of the human actions. The MHIs are described with the help of HOG feature descriptors, which are finally classified using a Nearest Neighbor (NN) classifier. The technique is applied on MuHAVi-uncut dataset having 17 action classes and achieves an accuracy of 95.4%. Yuan *et al.* 2013 [26] proposed a new global feature transform called the R transform, which captures the detailed geometrical distribution of interest points. In addition, they proposed a new technique to combine Bag of Visual Words (BoVW) with R transform so as to improve recognition accuracy. They achieved an accuracy of 95.49% on KTH dataset and 87.335% on UCF Sports dataset. Wu *et al.*, 2014 [27] proposed a method for HAR that represents each action class by hidden temporal methods. The video segment is described by a temporal pyramid model to capture the temporal structures. They achieved an overall accuracy of 84.3% on challenging datasets of Olympic Sports dataset and 47.1% on HMDB51 dataset. Sharif *et al.*, 2017 [28] proposed a framework based on uniform segmentation and combination of Euclidean distance and joint entropy-based features selection. Their work consists of four steps: segmentation of the subjects by fusing novel uniform segmentation and expectation maximization, extraction using HOG and Haralick features, feature selection using Euclidean distance and joint entropy-PCA-based method and feature classification using SVM. They have achieved an excellent accuracy of 95.80% on Weizmann dataset and 99.30% on KTH dataset. Sahoo *et al.*, 2019 [29] proposed fusion of histogram based features for action recognition called as Bag of Histogram of Optical Flow (BoHOF). After calculation of features, sobel edge filters [30] used, and median filtering is applied to suppress background noise. HOG is then extracted and mixed with BoHOF and finally, the result is fed into SVM. An overall accuracy of 96.70% has been achieved on KTH dataset. In [31], Ikizler-Cinbis et al. proposed multiple instance learning (MIL) framework based on multiple feature channels consisting of "*person-centric,*" "*object-centric,*" and "*scene-centric*" features. The framework attains the classification accuracy of 53.20% when applied to UCF11 dataset. In the work described in [32], the authors propose a new approach for the recognition of human actions based on affine-invariant shape representation and the features are classified using a SVM classifier. The approach is applied on two standard datasets such as Weizmann and KTH and attains the accuracies of 98% and 93.5% respectively. Kushwaha *et al.* in [33] design a new HAR framework which consists of three steps. Firstly, the subject is detected using background subtraction followed by extraction of contour-based pose features from human silhouettes. Finally, the human actions are classified using multi-class SVM classifier. The proposed framework yields an accuracy of 88.52% on KTH dataset.

3. Preliminaries

3.1 Genetic Algorithm

Genetic Algorithm (GA) is a popular metaheuristic approach for solving optimization problems [34–37]. It is inspired by evolution of biological features like selection, crossover, and mutation.

There are five basic steps in GA – creation of initial population, parent selection, crossover, mutation, and replacement of child chromosomes. GA starts with creating a random population of a finite number of chromosomes with each one having a fixed sized length. Initially, the individual genes of each chromosome are filled with random values. This serves as the initial population for GA. From this set of chromosomes, some are selected as parent chromosomes which are further passed for crossover and mutation to create child chromosomes. There are various approaches to select the parent chromosomes like Tournament selection [38], Roulette-wheel selection [39], etc. After child chromosomes are produced, they are evaluated using some fitness measure. If their fitness values surpass the fitness of some chromosomes in the current population, they are replaced in the population in place of the low fitness chromosomes.

The fitness function is defined over the genetic representation and measures the quality of the represented solution. These processes ultimately result in the next generation of chromosomes, which are again passed through the same process of selection, crossover and mutation to produce the subsequent generations. After certain number of generations, GA converges to a near-optimal solution. A binary version of GA is used in FS, where each chromosome is represented as vectors of '0' and '1's. A '0' means the corresponding feature is not selected, whereas a '1' means that the corresponding feature is selected.

3.2 Pearson Correlation Coefficient

Pearson Correlation Coefficient (PCC) is a correlation measure which is used in statistics [40–42] to find the strength of linear dependence between two random variables x and y. It is denoted by the following equation:

$$r_{xy} = \frac{\sum_{i=1}^{n}(x_i-\bar{x})(y_i-\bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i-\bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i-\bar{y})^2}} \quad (1)$$

where r_{xy} is the PCC value, x_i and y_i represents i^{th} sample in x and y, respectively, and n is the total number of samples.

In present context, we have considered different features as random variables. From the equation, we know that the PCC gives a real value between -1 to +1 for any two features. A value lesser than 0 means both the features are inversely dependent on each other, i.e. if one variable increases other will decrease. A value equal to 0 means both the features are independent to each other. A value greater than 0 signifies they both are directly dependent on each other i.e. if one feature increases, then the other will also increase.

3.3 Mutual Information

Mutual information (MI) [43, 44] is a measure between two (possibly multi-dimensional) random variables X and Y, that quantifies the amount of information obtained about one random variable, through the other random variable. The mutual information is given by

$$I(X;Y) = \sum_{x\epsilon X, y\epsilon Y} P_{xy}(x,y) log \frac{P_{xy}(x,y)}{P_x(x)P_y(y)} \quad (2)$$

where $P_{xy}(x, y)$ is the joint probability density function of X and Y, and $P_x(x)$ and $P_y(y)$ are the marginal density functions. The mutual information determines how similar the joint distribution $P_{xy}(x, y)$ is to the products of the factored marginal distributions. If X and Y are completely unrelated (and therefore independent), then $P_{xy}(x, y)$ would equal $P_x(x)P_y(y)$, and this integral would be zero.

3.4 Co-operative Game Theory

In the world of game theory, co-operative game (also known as coalition game) is a well-known game which revolves around formation of groups or coalitions of players depending on collective payoffs. Let there be n (>=2) players playing the game. N represents the entire set of players as $\{1, 2, 3\ldots n\}$. A coalition is defined as a subset $C \subset N$. So, there are 2^n ways of selecting a coalition. A formal definition of coalition can be found in [45] as:

The coalitional form of an n-person game is given by the pair (N, f), where $N = \{1, 2, \ldots, n\}$ is the set of players and f is a real-valued function, called the characteristic function of the game, defined on the set, 2^N, of all coalitions (subsets of N), and satisfying

i. $f(\emptyset) = 0$, where $\emptyset$ is the empty coalition.
ii. (super-additivity) if S and T are disjoint coalitions ($S \cap T = \emptyset$), then $f(S) + f(T) \leq f(S \cup T)$.

The value $f(C)$ is a real-value for any coalition $C \subset N$. It gives us a measure of strength or worth possessed by the coalition. So, the ultimate purpose of the game becomes to find a worthy coalition of players subject to some application related constraints.

4. Present Work

In this work, we have used co-operative game and GA in conjunction to reinforce each other with their individual qualities. When applied to the field of FS, coalition-game theory has a particular approach. So, first, we need to discuss a little about how we have adapted the concepts of coalition game for FS.

4.1 Co-operative (coalition) game for FS

Every co-operative game proceeds by creating a coalition of players. In FS, every feature acts as one of the players participating in the game. The contribution of a feature in a coalition is denoted by a special value known as the Shapley value. Shapley value considers the intrinsic properties of the features to calculate the worth with respect to their coalition.

Consider a coalition game with a total of m players. The entire set of players is defined as $M = \{1, 2, \ldots m\}$. Say in the process of the game a coalition C has been formed, which consists of n players (features). The contribution of any player f_i in C can be calculated as:

$$\Delta_i(C) = v(C \cup f_i) - v(C) \quad i = 1,2,..n$$
(3)

where $v(C)$ is the total amount of payoffs, the members of C can accumulate by co-operating. The Shapley value for a particular player is the sum of contributions of that player in all possible combinations of coalitions. Let's denote Shapley value for player i with respect to a set of players M and payoff function v as $S_i(M, v)$. The expression for S_i can be represented as:

$$S_i(M, v) = \frac{1}{2^{m-1}} \sum_{C \subseteq \mathrm{M}} \Delta_i(C) \quad (4)$$

The function v and thereby $\Delta_i(C)$ varies depending on the context where the coalition game-theory is being used. For the purpose of FS, we have used a PCC-based independence metric to find the worth of a feature in a coalition with other features. So, we can see that there are certain things we need to specify before applying coalition game in any context – Contribution of a feature in a coalition (Δ_i), Shapley value of a feature (S_i). In the following portions, we have described how we have performed the calculation of Δ_i, S_i and finally, the importance of a feature in the entire feature space.

4.1.1 Contribution of a feature in a coalition (Δ_i)

FS may be considered as an incremental process where we have a set of selected features and another set of unselected features. Now, if we want to move a feature from the unselected set to the selected one, there should be a measure by which we can say that the transfer of the feature will be fruitful. To find that measure, we need to discuss a bit about the FS process. In FS, we are trying to select the most discriminating set of features which can help in the classification process. For example, if two features are having same values, both of them are not helping in classification, and so only one of them can be used. The other one then becomes a redundant feature. Thus, our ultimate objective in FS is to remove those redundant features. Linear dependency is a great way to find a redundant feature with respect to the other features. So, if we want to transfer an unselected feature to the selected feature set, a great way is to choose a feature which is independent of the selected feature set. We have used this concept to calculate the worth of a feature in the coalition formation.

As discussed in the previous section, PCC is a correlation metric used for finding linear dependency between two random variables. Say we want to check the correlation between features f_1 and f_2. So, we first calculate the PCC value $r_{f_1 f_2}$ according to Eqn. 1 and then the final decision is made based on the following fact.

$$f_1 \text{ and } f_2 \text{ are : } \left\{ \begin{matrix} independent; if \ |r_{f_1 f_2}| < threshold \\ \\ dependent; if \ |r_{f_1 f_2}| \geq threshold \end{matrix} \right\}$$

According to the theory of PCC, two variables are correlated if the PCC value is closer to 1 than to 0. The *threshold* in this work is set to 0.5 because if the PCC value between two features is greater than 0.5, then it is closer to 1 and thus correlated else; they are not. It is to be noted that although we say when the PCC value between two features is less than 0.5, they are independent, but they are not completely independent; somewhat, their mutually dependency can be ignored to suit our purpose. In a coalition C, the contribution of a feature $\Delta_i(C)$ is determined by the number of features independent to that feature divided by the number of dependent features.

$$I_i(C) = \{set\ of\ features\ in\ C\ independent\ of\ f_i\}$$

$$D_i(C) = \{set\ of\ features\ in\ C\ dependent\ on\ f_i\}$$

$$\Delta_i(C) = \frac{|I_i(C)|}{|D_i(C)|} \qquad (5)$$

4.1.2 Shapley value calculation (S_i)

In order to calculate the Shapley value of a feature f_i, we need to find the contribution of the feature for every coalition possible in the scenario. So, finally the Shapley value of f_i becomes:

$$S_i = \sum_{C \subseteq \mathrm{M}} \Delta_i(C) \qquad (6)$$

4.1.3 Scoring of features

In our model, we have also used MI values of the features to assign a score value to every feature at each iteration. The final score value of a feature f_i is given by

$$score(i) = S_i * I(f_i; class) \tag{7}$$

where S_i is the Shapley value for f_i and $I(f_i; class)$ is the MI value of f_i with the corresponding pattern classes computed using Eqn. 2.

4.2 Co-operative GA (CGA)

The primary motivation behind proposing this model is a significant drawback present in coalition game approach followed in FS. In section 4.1.2, we have presented the equation for Shapley value calculation. It can be observed that generally, it is computed based on contributions of the feature in every possible coalition. In practice, it is impossible to calculate the contribution for every coalition because for n features, there will be 2^{n-1} possible coalitions to consider for a feature. So, for n features, the total time complexity for computing Shapley values will become $O(n * 2^{n-1})$, which is exponential in nature. Hence, now-a-days researchers are trying to provide alternative approaches to calculate Shapley values.

CGA consists of two different parts:

i. Enhanced GA (EGA)
ii. Coalition Game

In CGA, we have made an effort to reduce the computational complexity of calculating Shapley value and use that Shapley value to improve the solution of EGA. Thus, it is a bidirectional reinforcement approach between EGA and coalition game where at each iteration of CGA, Shapley value is being computed based on the solutions of EGA, and that value is again used to produce next solution of EGA.

Let us discuss both of them one-by-one:

4.2.1 EGA

EGA uses all the basic approaches used in GA, like parent selection, crossover, mutation. In addition, EGA embodies a new fitness function and a MI-guided mutation procedure. Mutation in GA is a random process where some of the features are randomly selected and their states are toggled i.e. if a feature is in state ‘0’, it is made ‘1’ and vice versa. We wanted to use some guided approach in place of this random one. Hence, we have proposed a MI-guided mutation approach where some random features are selected for mutation, but they are not directly toggled. Say, a random feature set S is selected for mutation, then the MI-guided mutation does the following:

$$f_i = \left\{ \begin{matrix} 1; if\ I(d_i, class) > rand(1) \\ 0; otherwise \end{matrix} \right\} \quad f_i \epsilon S$$

where d_i and $I(d_i, class)$ are the feature values and MI score of the feature f_i respectively.

We have also used a new fitness evaluation function. At each iteration of EGA, we need to evaluate the candidate solution or chromosomes for parent selection as well as child replacement procedures. The fitness function used for the evaluation of chromosomes in EGA is given by Eqn. 8.

$$fitness(X) = \alpha * accuracy(X) + \beta * \left(\frac{no.of\ features\ in\ X\ with\ state\ "0"}{total\ no.of\ features\ in\ X}\right) + (1 - \alpha - \beta) * \sum_{i=1}^{n}(X_i == 1) * score(i) \quad (8)$$

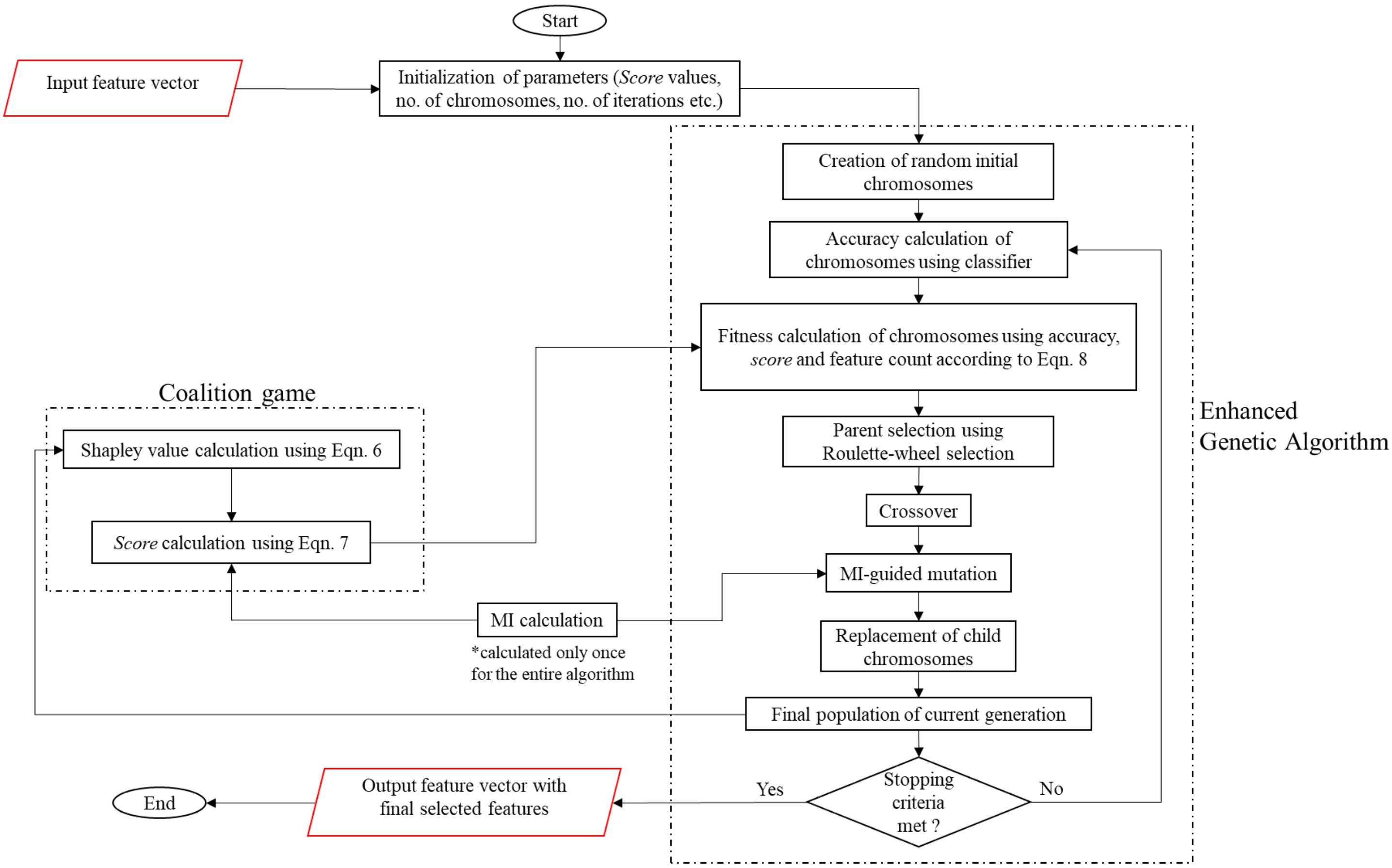
In the above equation, $accuracy(X)$ represents the classification accuracy provided by the feature vector X, X_i is the state of the i^{th} feature in X and $score(i)$ is the final score of the i^{th} feature as calculated by Eqn. 7. α, β are two parameters used to provide weightage to the three different components of the fitness function. We have used α as 0.5 and β as 0.2.

4.2.2 Coalition Game

After EGA produces the final population of chromosomes in an iteration, they are passed to coalition game for computation of Shapley values for the next iteration. As we have discussed before, it is practically impossible to compute the contribution of every feature for all the possible coalitions. Hence, instead of considering the features for every coalition, we have used the final chromosomes of EGA as the feature coalitions, which we have considered for computing the Shapley values. From the perspective of EGA, we can say that the final population of the algorithm may be considered as the most crucial combination of features handpicked by the algorithm itself. So, this reduces the computational complexity of the Shapley value calculation procedure to a large extent. Normally, EGA uses very small population size like 20-50. So, in the proposed model, the Shapley value computation requires $O(n)$ time. Moreover, the chromosomes in EGA gets guided to better solutions through the fitness function involving the Shapley values.

The entire model of CGA can be described by the flowchart represented in Figure 1. From the Figure, we can see that the two segments of the model, namely Coalition game and EGA, are interacting with each other through the sharing of *score* values generated using Eqn. 7. MI values are computed only once and shared with both the segments because of their requirements.

Figure 1: Flowchart of CGA model consisting of two different segments, namely EGA and Coalition game interacting with each other.


Start
Input feature vector
Initialization of parameters (*Score* values, no. of chromosomes, no. of iterations etc.)
Creation of random initial chromosomes
Accuracy calculation of chromosomes using classifier
Fitness calculation of chromosomes using accuracy, *score* and feature count according to Eqn. 8
Coalition game
Shapley value calculation using Eqn. 6
Score calculation using Eqn. 7
Parent selection using Roulette-wheel selection
Crossover
MI-guided mutation
MI calculation
*calculated only once for the entire algorithm
Replacement of child chromosomes
Final population of current generation
Enhanced Genetic Algorithm
Stopping criteria met ?
Yes
No
Output feature vector with final selected features
End

5. Brief Overview of Datasets and Feature Descriptors

5.1 Datasets

5.1.1 KTH

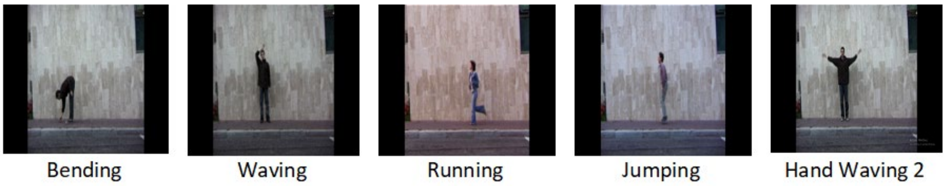
KTH dataset [11] consists of six classes, namely *Boxing*, *Hand-clapping*, *Hand-waving*, *Jogging*, *Running*, *Walking*. The dataset consists of 599 videos which are equally divided among the classes (100 videos each) except *Hand-clapping* which consists of 99 videos. KTH dataset consists of six classes of human activities in 599 videos representing natural variations of human actions in terms of camera view-point, human pose, clothing, occlusions, and scene background. Sample frames from KTH dataset is shown in Figure 2.

Figure 2: Sample frames from the KTH dataset

5.1.2 Weizmann

Weizmann dataset [10] consists of 90 videos of 10 actions (*Walking*, *Running*, *Jumping*, *Galloping sideways*, *Bending*, *One hand waving*, *Two hand waving*, *Jumping in place*, *Jumping jack* and *Skipping*). Each of these 10 actions are performed by 10 different persons. The resolution of the videos is 144*180 pixels, with 25 frames per second. Sample frames from Weizmann dataset is shown in Figure 3.

Figure 3: Sample frames from Weizmann dataset



5.1.3 UCF11

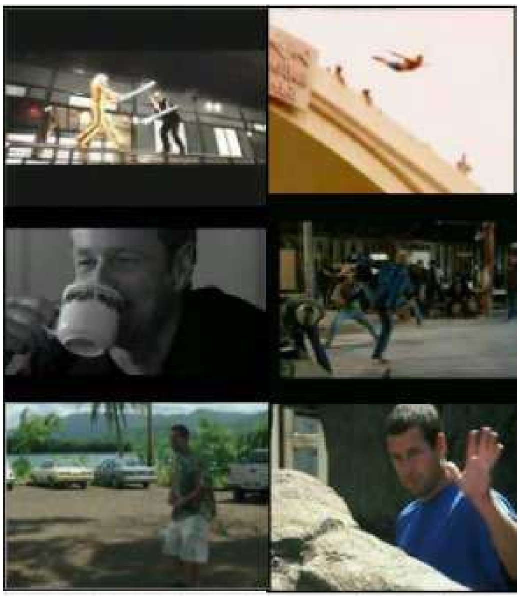
UCF11 dataset [12] consists of 11 action classes: *basketball shooting, biking/cycling, diving, golf swinging, horseback riding, soccer juggling, swinging, tennis swinging, trampoline jumping, volleyball spiking,* and *walking with a dog* for which, 25 groups with more than 4 clips in each group were there. The video clips in the same group share some standard features, such as the same actor, similar background, similar viewpoint, and so on. This is a very challenging dataset as it has significant variations in camera motion, object appearance and pose, object viewpoint, scale, background cluttering, illumination conditions, etc. Some samples of the frames taken from this dataset is shown in Figure 4.

Figure 4: Sample frames from the UCF 11 dataset

5.1.4 HMDB51

HMDB51 dataset [13] consists of 51 classes, of which each class consists of a minimum of 101 video clips. The action categories can be grouped into five types: a) *General facial actions* such as smile, laugh, talk etc. b) *Facial actions with object manipulation* such as smoke, eat, drink etc. c) *General body movements* such as cartwheel, clap hands, backhand flip. d) *Body movements with object interaction* such as brush hair, catch, draw sword etc. e) *Body movements for human interaction* such as fencing, hug, punch etc. The dataset comprises a variety of realistic videos collected from YouTube and Google videos. Some sample frames from the dataset is illustrated in Figure 5.

Figure 5: Sample frames from HMDB51 dataset



5.1.5 UCI HAR

This dataset has been prepared using a group of 30 volunteers within 19-48 years of age. Each participant performed six activities, namely WALKING, SITTING, LAYING, STANDING, WALING_UPSTAIRS and WALKING_DOWNSTAIRS wearing Samsung Galaxy SII smartphone. Data obtained from 70% volunteers were used for training and rest were used for testing. Each record in this dataset contains accelerometer's triaxial acceleration and estimated body acceleration, Gyroscope's triaxial angular velocity, 561-feature vector with time and frequency domain variables, activity label and an id of the subject who performed the experiment. It contains a total of 10299 instances and 561 attributes.

5.1.6 UCI HAR_AAL

It is an addition to UCI HAR dataset mentioned in 5.1.5. More data were collected to perform a social connectedness experiment in Ambient Assisted Living (AAL). Just like UCI HAR dataset, it contains 561 attributes but has a total of 5744 instances.

5.2 Feature Descriptors

As mentioned earlier, a set of four different feature descriptors, namely, HOG [15], GLCM [16], SURF [17] and GIST [18], are extracted from the video frames in the present work. These features are computed after performing background subtraction (using one of the previous methods described in [46]) and detection of minimum bounding box. Since these feature descriptors are well-known terms in the pattern recognition community, we have just described these descriptors below in brief.

5.2.1 HOG

The HOG is a feature descriptor used in computer vision and also image processing methods for object detection [15].The HOG feature descriptor counts occurrences of gradient orientation in localized portions of an image. At first, horizontal and vertical gradients are calculated from each minimum bounding box subtracted from the given frames of an input video. This can be quickly done by filtering the image with the kernels [-1 0 1] and $[-1\ 0\ 1]^T$. The magnitude and direction of gradient are then calculated for every pixel. The image is divided into small blocks or regions called cells of 16x16 pixels and then a histogram containing 9 bins corresponding to the angles 0-180 degrees is created. Finally, a set of 8640 features are extracted from each video frame using HOG descriptor.

After calculating HOG features from each of the 20 selected frames, 100 of the most useful features are selected from each frame using sparse filtering technique [47]. The selected features for these 20 frames are then aligned into a row vector and thus for HOG, a total of 2000 features are selected for each action class.

5.2.2 GLCM

GLCM was first proposed by Robert M Haralick, K Shanmugam, Its'hakDinstein (1973) in his work "Textural Features for Image Classification" [16] .GLCM with associated texture features is a method of extracting statistical texture features of second order.

GLCM is a matrix where the number of rows and columns is equal to the gray levels in the image. The matrix element P or Pij is the conditional probability of the intensities of two pixels with intensities i and j given that the separation between the two pixels is $(delx, dely)$. On the other hand, the matrix element $P(I, J|d, \theta)$ is the second order statistical probability for change between i and j for a given angle θ and distance d.

From the second order statistics mentioned above, one can derive some useful properties, all of them have been used for our experiment. There are a total of 13 features calculated from GLCM, each feature containing a definite amount of information about an input image. These features are as follows: Energy, Entropy, Homogeneity, Correlation, etc. GLCM also requires an 'offset' value which defines relationships of varying direction and distance and 'offset' values of [0 1; -1 1; -1 0; -1 -1] (represented in a matrix form) is used for calculating the feature values from each selected frame. As a result, a feature set consisting of 1040 elements is produced.

After calculating the GLCM based features from each of the 20 selected frames, 50 feature values are selected from each frame using sparse filtering technique [47]. The selected feature values for these 20 frames are then aligned into a row vector, and thus for GLCM feature, a total of 1000 feature values are selected for each action class.

5.2.3 SURF

SURF is a local feature detector , and it was first proposed by Bay, Tuytelaars, and Gool L Van in their work [17]. It is known to be an upgraded version of Scale Invariant Feature Transform (SIFT). The SURF descriptor is scale and rotation invariant. To make the descriptor scale invariant, it is sampled over a window which is proportional in scale with the detected window size. In order to achieve rotational invariance, the orientation of the point of interest needs to be found. The Haar wavelet responses in both x- and y-directions within a circular neighborhood of radius 6S around the point of interest are computed, where S is the scale at which the point of interest was detected. The obtained responses are weighted by a Gaussian function centered at the point of interest, then plotted as points in a two-dimensional space, with the horizontal response in the abscissa and the vertical response in the ordinate. After rotation, to describe the region around the point, a rectangular region is extracted, centered on the interest point and oriented along the orientation as selected above. The size of this window is 20S.

The interest region is split into smaller 4x4 square sub-regions, and for each one, the Haar wavelet responses are extracted at 5x5 regularly spaced sample points. The responses are weighted with a Gaussian filter. Finally, a feature vector of dimension 400 is designed using the SURF feature descriptor.

After calculating SURF features from each of the 20 selected frames, 20 feature values are selected from each frame using sparse filtering technique [47]. The selected features for these 20 frames are then aligned into a row vector, and thus for SURF feature, a total of 400 feature values are selected for each action class.

5.2.4 GIST

GIST features are used for estimating various statistical characteristics of an image [18]. GIST features are calculated by using convolution of the Gabor filter with each of the selected frames

at different scales and orientation. In our experiment, we have used a Gabor filter transfer function having orientation per scale as [8 8 4]. The values for each convolution of the filter at each scale and orientation are used as GIST features for a particular frame. Thus, in this way, repetitive gradient directions having both low and high frequencies of a frame can be measured. GIST feature extraction filters the input frame first into a number of low-level visual feature channels such as color, orientation, motion, intensity and flicker. Within each sub-channel, center-surround operations are performed between filter output maps at different scales. With the above feature map, a saliency map can be yielded by using the feature map to detect clearly visible regions in each channel. Now each sub-channel extracts a GIST vector from its corresponding feature map after computing the low-level features. A 4x4 grid is used in our experiment for averaging the conspicuous sub-regions over the map which outputs a feature vector of size 6400 for each video frame.
After calculating GIST features from each of the 20 selected frames, 100 features are selected from each frame using sparse filtering technique [47]. The selected features for these 20 frames are then aligned into a row vector, and thus for gist feature, a total of 2000 features are selected for each action class.

In our work, we have reduced the total number of features per frame through sparse filtering into an available total of 5400 features (2000 features for HOG, 400 features for SURF, 1000 features for GLCM and 2000 features for GIST) representing a video as mentioned in each section of the feature descriptors.

5.3 Relation among Datasets, Feature Descriptors, and Feature Selection

At this point of the manuscript, it may become a bit hazy to grasp how datasets, features descriptros and FS are related to each other. That is why we have provided a quick overview of the existing relations amon these three parameters in this section. The proposed FS procedure is applied over four standard feature vectors. We have extracted the feature vectors from four video-based datasets, namely KTH, Weizmann, UCF11 and HMDB51 using the feature descriptors explained in section 5.2.1-5.2.4. The rest two datasets (sensor-based) namely UCI HAR and UCI HAR_AAL, are available in form of vectors in UCI machine learning repository [48]. After the feature vectors are generated, they are divided into training and testing sets. We have applied the proposed FS method named CGA over only the training feature vectors of every dataset. The FS provides the features which are relevant for the classification of activities in a particular dataset. The final set of features are then used to classify activities which are present in the testing vectors. Classification accuracy is then measured as:

$$Classification\ Accuracy = \frac{No.of\ correct\ classification\ in\ testing\ vectors}{Total\ no.of\ testing\ vectors} * 100 \qquad (9)$$

6. Result Analysis and Comparison

This section presents the results obtained by the proposed method over different feature sets. The focus has been given on two of the most crucial evaluation metrics for any FS model – classification accuracy obtained by CGA over the feature sets and the number of features used to achieve that accuracy. The accuracy and number of features selected by CGA are then compared with some benchmark as well as recently proposed meta-heuristic algorithms in the domain of FS. In order to provide a neutral environment for comparison, each of these algorithms are run on the feature sets for 5 times and Multi-layer Perceptron (MLP) [49] classifier is used to generate the classification accuracies with number of neurons set to 70 (MLP-70). Table 1 presents the best, worst, average accuracies and standard deviation achieved by CGA over the feature sets used for experimentations. From the table, we can see the difference between the best and worst accuracies are very less and the standard deviation is also negligible. Only for KTH and Weizmann, the standard deviation is more than 1 (approx. 4.8 and 7.2 respectively). For other feature sets, the standard deviation is very less (less than 1). These facts prove that CGA is able to provide stable results over different runs and there is not much deviation. For KTH, CGA is able to achieve 100 % accuracy whereas for UCI HAR, CGA has got accuracies of more than 90%. For the remaining datasets, even though CGA is not able to get 90% or above accuracy, the results are quite remarkable when compared to other methods.

Dataset	Best Accuracy	Worst Accuracy	Average Accuracy	Standard Dev
UCF11	54.644068	52.74573055	53.0587919	0.586018941
UCI HAR_AAL	86.39410188	85.1877	85.8579	0.4446
HMDB51	53.8749	51.8207	52.8852	0.8288
UCI HAR	95.7923	94.7743	95.1883	0.5215
KTH	100	82.77777778	92.97777778	4.794256229
Weizmann	48	32	40.8	7.1554

Table 1: Description of the results obtained by CGA over different feature sets. The best accuracy, worst accuracy, average accuracy, and standard deviation are provided for 5 runs of CGA.

In order to prove the effectiveness of the proposed method, the results obtained by CGA are tallied against some old and some recently proposed FS models – GA, BPSO, BGSA, BGSO, WFACOFS, and HMOGA. The comparison data is provided in Table 2 and 3. Table 2 compares the average classification accuracies obtained by different algorithms over the same set of features for 5 runs. The rank of CGA among all the algorithms is provided in a column in Table 2. From the table, we can observe that CGA has been able to get the best average classification accuracy over 5 out of 6 feature sets, and for the only other dataset, KTH, it is ranked second. These results clearly shown how good CGA is in terms of classification for different feature sets. The average accuracy (AVG) obtained by different methods for varying features are also provided in Table 2. According to the AVG values of Table 2, CGA is able to provide an average accuracy of 76.79% which is clearly 4-5% more than the second ranked algorithm which is WFACOFS.

Average Classification Accuracy (in %)								
Dataset	**CGA**	**BPSO**	**BGSA**	**GA**	**BGSO**	**WFACOFS**	**HMOGA**	**Rank**
UCF11	**53.0588**	51.4497	51.8655	51.858	52.354	51.245	51.34	1
UCI HAR_AAL	**85.8579**	74.0879	82.345	82.5268	80.63	83.467	83.244	1
HMDB51	**52.8852**	46.076	48.0541	48.5294	48.9262	51.25	49.7666	1
UCI HAR	**95.1883**	92.8952	93.4213	92.567	94.2993	92.4036	93.8242	1
KTH	92.97778	85.2929	85.2242	90.38	**94.444**	89.45	90.54	2
Weizmann	**40.8**	28.0509	28.271	20	8	26.7574	20	1

AVG	**70.128**	62.97543	64.86352	64.3102	63.109	65.7621667	64.7858	1

Table 2: Comparison of average accuracies obtained by different FS algorithms (CGA, BPSO, BGSA, GA, BGSO, WFACOFS, and HMOGA) over different feature vectors. The maximum average accuracy is made bold. The rank of CGA in terms of classification accuracy among all the algorithms over all the feature vectors are also provided in the last column of the table.

Apart from accuracy, number of features selected by a FS algorithm is also significant. The average number of features selected by each algorithm over different feature vectors is presented in Table 3. From Table 3, we can see that even though CGA does not always use the lowest set of features among the algorithms, it uses a balanced set of features to achieve the high classification accuracies presented in Table 1. CGA uses the least features for KTH and Weizmann feature sets among all the algorithms. From the table, it can also be seen that WFACOFS is very good in terms of average number of features. As it uses considerably smaller number of features in most of the cases, its classification prowess gets compromised. As a conclusion, it can be stated that as classification accuracies are always given higher priority over the number of selected features in FS problems, CGA performs the best among all the algorithms used for comparison.

Average features (in %)							
Dataset	**CGA**	**BPSO**	**BGSA**	**GA**	**BGSO**	**WFACOFS**	**HMOGA**
UCF11	30.55556	49.60477	52.76738	47.26754	47.56174	9.95364179	41.67781
UCI HAR_AAL	71.96078	46.25668	59.19786	57.29947	43.67201	6.25668449	45.811052
HMDB51	66.88148	44.84167	62.27593	50.41481	49.25926	63.4638889	46.314815
UCI HAR	68.82353	53.5205	55.10695	50.24064	50.62389	6.71122995	44.56328
KTH	18.60463	36.14352	34.80278	31.23796	36.38889	38.675	31.481481
Weizmann	44.27593	47.1537	46.98426	46.2787	51.40741	58.5240741	44.703704

Table 3: Comparison of average number of features selected by different FS algorithms (CGA, BPSO, BGSA, GA, BGSO, WFACOFS, and HMOGA) over different feature vectors.

In addition, to prove the systematic improvement in the population of the algorithm, which is the heart of any evolutionary algorithm, a comparison of algorithms are provided in terms of convergence of accuracies over iterations for different meta-heuristic algorithms. For proper evaluation, three top ranked algorithms (in terms of AVG classification accuracy), i.e. CGA, WFACOFS and BGSA, are selected from Table 2. Figures 6-11 represent the convergence of accuracies obtained by these three algorithms over UCF11, UCI HAR_AAL, HMDB51, UCI HAR, KTH and Weizmann feature sets respectively.

Figure 6: Convergence graph depicting Accuracy (using MLP-70) vs. Iteration no. for CGA, WFACOFS and BGSA over UCF11 feature set.

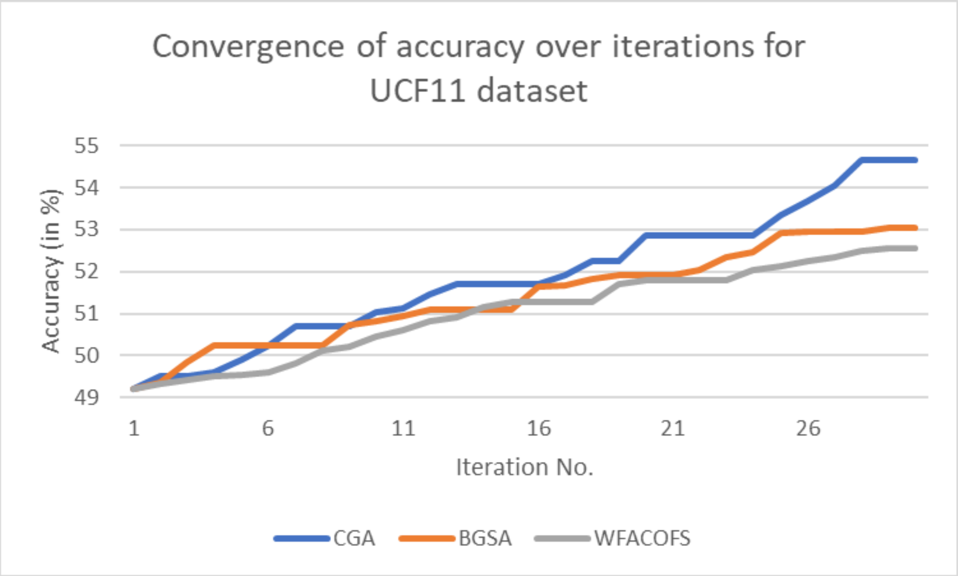


Figure 7: Convergence graph depicting Accuracy (using MLP-70) vs. Iteration no. for CGA, WFACOFS and BGSA over UCI HAR_AAL feature set.

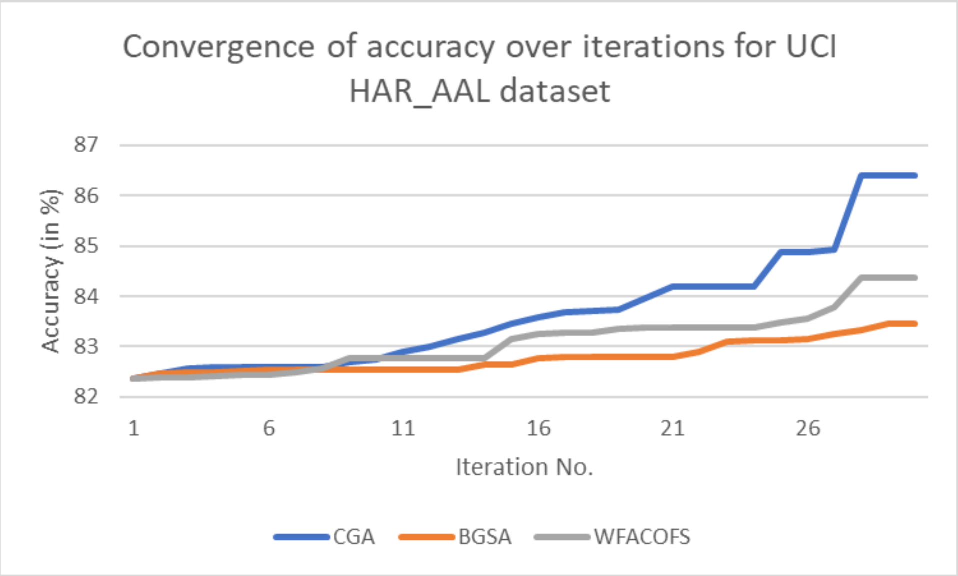


Figure 8: Convergence graph depicting Accuracy (using MLP-70) vs. Iteration no. for CGA, WFACOFS and BGSA over HMDB51 feature set.

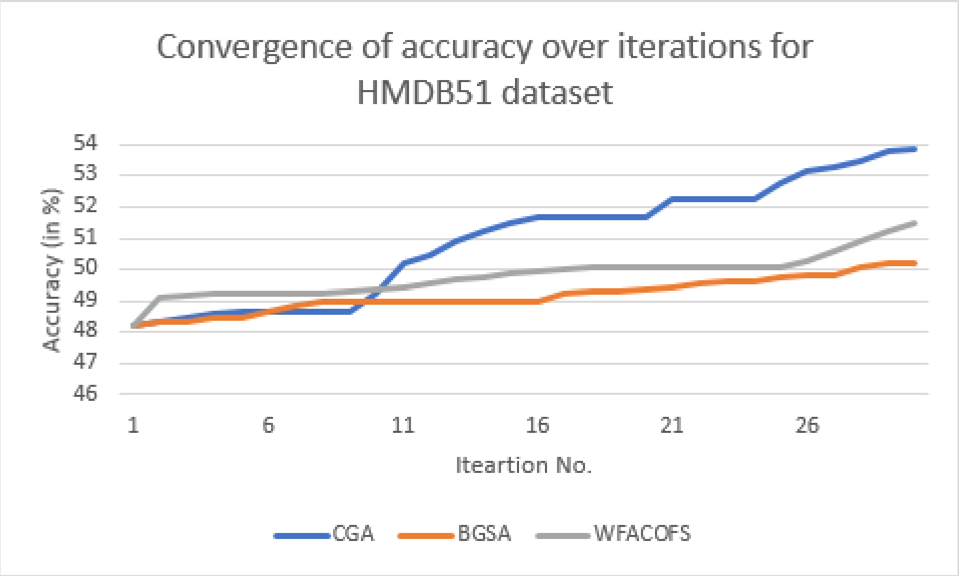


Figure 9: Convergence graph depicting Accuracy (using MLP-70) vs. Iteration no. for CGA, WFACOFS and BGSA over UCI HAR feature set.

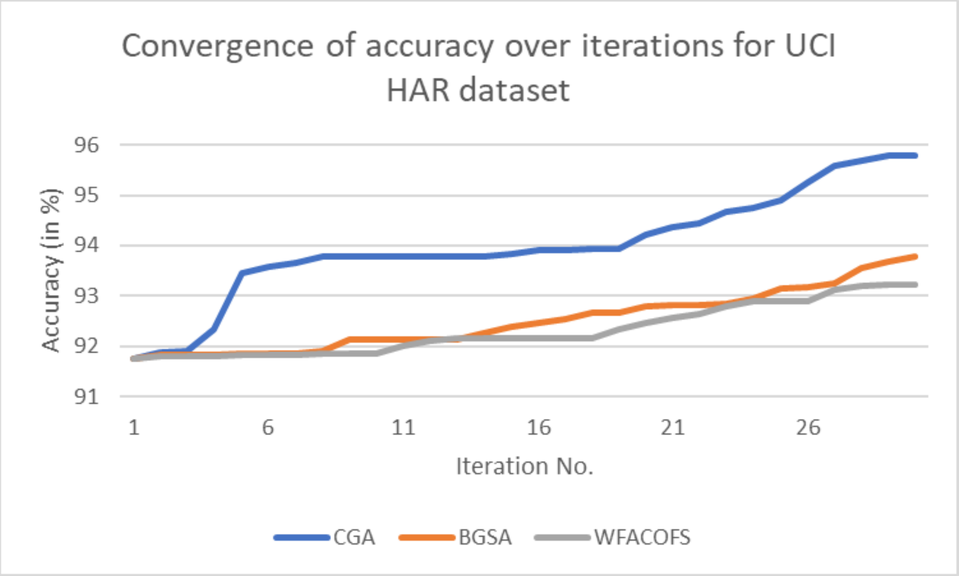


Figure 10: Convergence graph depicting Accuracy (using MLP-70) vs. Iteration no. for CGA, WFACOFS and BGSA over KTH feature set.

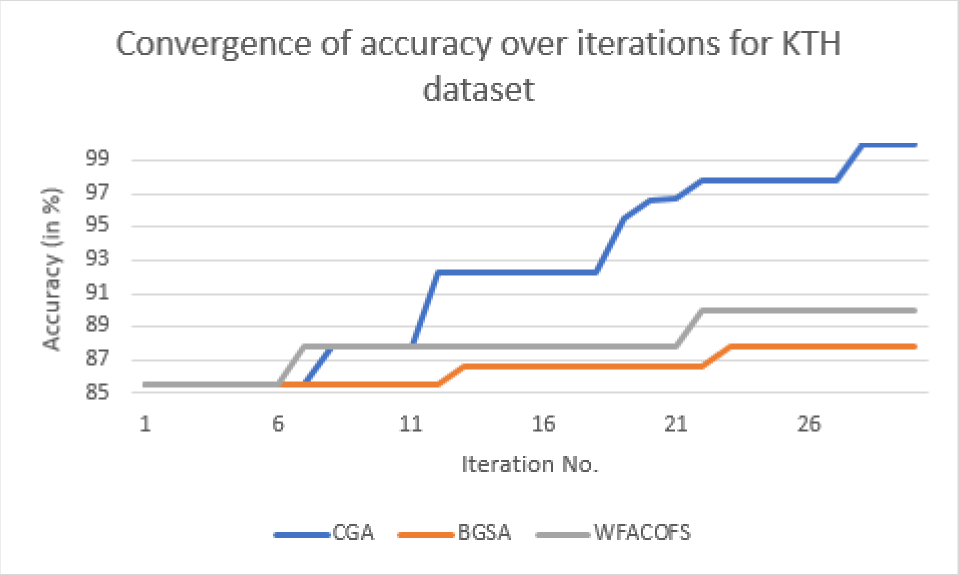
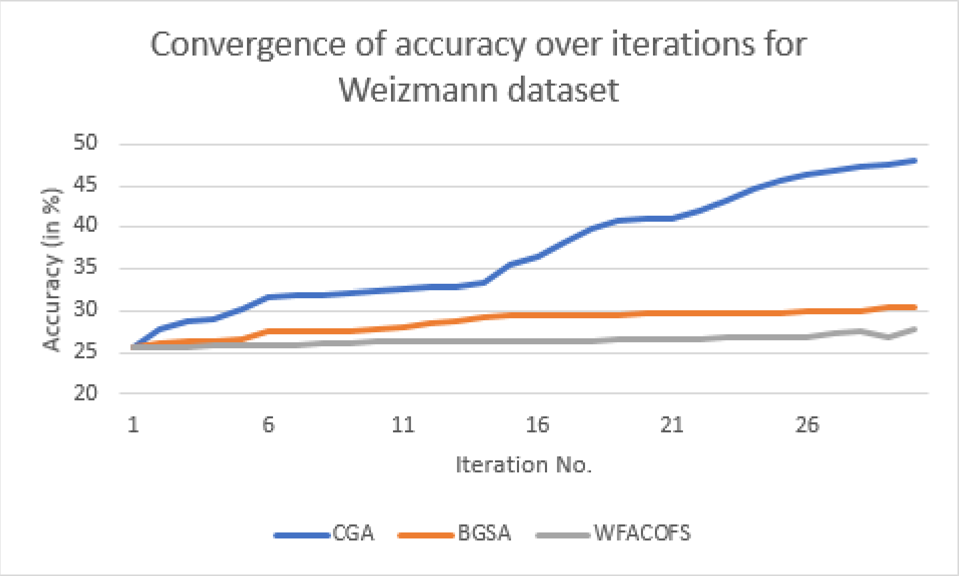


Figure 11: Convergence graph depicting Accuracy (using MLP-70) vs. Iteration no. for CGA, WFACOFS and BGSA over Weizmann feature set.



From the convergence graphs, it is visible that CGA is able to get reasonable convergence rate in terms of classification accuracy compared to the other algorithms. This establishes the success of CGA as an evolutionary algorithm because it is able to improve the solutions over the iterations stably. We can clearly state that CGA is an excellent evolutionary algorithm which is applicable to solve FS problems which is evident from the discussion of the results.

7. Conclusion

From the results and associated discussion, we can see that CGA has exceptional abilities of FS. CGA is able to address a significant drawback of coalition game – huge time requirement for processing worth of a single feature. Thus, it is a novel attempt to make coalition game applicable to the domain of FS. From the experiments, we can state that there are many redundant features in the HAR datasets. CGA is able to produce very high accuracy using lesser than 70% of the features. This level of reduction can hugely affect the HAR tasks by making them considerably faster. In future, we want to apply this method on various new fields other than HAR. We also want to explore other application areas of coalition game like engineering optimization or image enhancement.

Declarations

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Aslan MF, Durdu A, Sabanci K (2019) Human action recognition with bag of visual words using different machine learning methods and hyperparameter optimization. Neural Comput Appl 1–13

2. Sahoo SP, Ari S (2019) On an algorithm for human action recognition. Expert Syst Appl 115:524–534

3. Saggese A, Strisciuglio N, Vento M, Petkov N (2019) Learning skeleton representations for human action recognition. Pattern Recognit Lett 118:23–31

4. Zhang P, Lan C, Xing J, et al (2019) View adaptive neural networks for high performance skeleton-based human action recognition. IEEE Trans Pattern Anal Mach Intell

5. Zhang H-B, Zhang Y-X, Zhong B, et al (2019) A comprehensive survey of vision-based human action recognition methods. Sensors 19:1005

6. Liu H, Motoda H (2007) Computational Methods of Feature Selection. CRC Press

7. Mitra P, Murthy CA, Pal SK (2002) Unsupervised feature selection using feature similarity. IEEE Trans Pattern Anal Mach Intell 24:301–312

8. Shang W-Q, Qu Y-L, Huang H-K, et al (2006) Fuzzy knn text classifier based on gini index. J Guang xi Norm Univ Nat Sci Ed 24:87–90

9. Dorigo M, Birattari M (2011) Ant colony optimization. In: Encyclopedia of machine learning. Springer, pp 36–39

10. Eberhart R, Kennedy J (1995) A new optimizer using particle swarm theory. In: Micro Machine and Human Science, 1995. MHS'95., Proceedings of the Sixth International Symposium on. IEEE, pp 39–43

11. Malakar S, Ghosh M, Bhowmik S, et al (2019) A GA based hierarchical feature selection approach for handwritten word recognition. Neural Comput Appl 1–20

12. Ghosh M, Begum S, Sarkar R, et al (2019) Recursive Memetic Algorithm for gene selection in microarray data. Expert Syst Appl 116:172–185

13. Ghosh M, Adhikary S, Ghosh KK, et al (2019) Genetic algorithm based cancerous gene identification from microarray data using ensemble of filter methods. Med Biol Eng Comput 57:159–176

14. Ghosh M, Malakar S, Bhowmik S, et al (2019) Feature Selection for Handwritten Word Recognition Using Memetic Algorithm. In: Advances in Intelligent Computing. Springer, pp 103–124

15. Dalal N, Triggs B (2010) Histograms of Oriented Gradients for Human Detection. IEEE Comput Soc Conf Comput Vis Pattern Recognit 886–893

16. Robert M. Haralick ; K. Shanmugam ; Its'Hak Dinstein (1973) Textural Features for Image Classification. IEEE Trans Syst Man Cybern SMC-3:610–621

17. Bay H, Tuytelaars T, Van Gool L (2006) Surf: Speeded up robust features. In: European conference on computer vision. Springer, pp 404–417

18. Oliva A, Torralba A (2001) Modeling the shape of the scene: A holistic representation of the spatial envelope. Int J Comput Vis 42:145–175. https://doi.org/10.1023/A:1011139631724

19. Blank M, Gorelick L, Shechtman E, et al (2005) Actions as space-time shapes. In: Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1. IEEE, pp 1395–1402

20. Schuldt C, Laptev I, Caputo B (2004) Recognizing human actions: a local SVM approach. In: Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004. IEEE, pp 32–36

21. Liu J, Luo J, Shah M (2009) Recognizing realistic actions from videos in the wild. Citeseer

22. Kuehne H, Jhuang H, Garrote E, et al (2011) HMDB: a large video database for human motion recognition. In: 2011 International Conference on Computer Vision. IEEE, pp 2556–2563

23. Anguita D, Ghio A, Oneto L, et al (2013) A public domain dataset for human activity recognition using smartphones. In: Esann

24. Anguita D, Ghio A, Oneto L, et al (2012) Human activity recognition on smartphones using a multiclass hardware-friendly support vector machine. In: International workshop on ambient assisted living. Springer, pp 216–223

25. Luvizon DC, Tabia H, Picard D (2017) Learning features combination for human action recognition from skeleton sequences. Pattern Recognit Lett 99:13–20

26. Yuan C, Li X, Hu W, et al (2013) 3D R transform on spatio-temporal interest points for action recognition. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. pp 724–730

27. Wu J, Hu D, Chen F (2014) Action recognition by hidden temporal models. Vis Comput 30:. https://doi.org/10.1007/s00371-013-0899-9

28. Sharif M, Khan MA, Akram T, et al (2017) A framework of human detection and

action recognition based on uniform segmentation and combination of Euclidean distance and joint entropy-based features selection. Eurasip J Image Video Process 2017:. https://doi.org/10.1186/s13640-017-0236-8

29. Sahoo SP, Silambarasi R, Ari S (2019) Fusion of histogram based features for Human Action Recognition. In: 2019 5th International Conference on Advanced Computing and Communication Systems, ICACCS 2019. IEEE, pp 1012–1016

30. Gupta S, Ghosh Mazumdar S (2013) Sobel Edge detection algorithm. Int J Comput Sci Manag Res 2:1578–1583

31. Ikizler-Cinbis N, Sclaroff S (2010) Object, scene and actions: Combining multiple features for human action recognition. In: European conference on computer vision. Springer, pp 494–507

32. Sadek S, Al-Hamadi A, Krell G, Michaelis B (2013) Affine-invariant feature extraction for activity recognition. ISRN Mach Vis 2013:

33. Kushwaha AKS, Srivastava R A Framework for Human Activity Recognition using Pose Feature for Video Surveillance System. Int J Comput Appl 975:8887

34. Holland JH (1992) Genetic Algorithms. Sci Am 1:66–73

35. Abualigah LMQ, Hanandeh ES (2015) Applying genetic algorithms to information retrieval using vector space model. Int J Comput Sci Eng Appl 5:19

36. Ghosh M, Guha R, Mondal R, et al (2018) Feature selection using histogram-based multi-objective GA for handwritten Devanagari numeral recognition

37. Nezamabadi-Pour H (2015) A quantum-inspired gravitational search algorithm for binary encoded optimization problems. Eng Appl Artif Intell 40:62–75. https://doi.org/10.1016/j.engappai.2015.01.002

38. Miller BL, Goldberg DE (1995) Genetic algorithms, tournament selection, and the effects of noise. Complex Syst 9:193–212

39. Lipowski A, Lipowska D (2012) Roulette-wheel selection via stochastic acceptance. Phys A Stat Mech its Appl 391:2193–2196. https://doi.org/10.1016/j.physa.2011.12.004

40. Mukaka MM (2012) A guide to appropriate use of correlation coefficient in medical research. Malawi Med J 24:69–71

41. Lawrence I, Lin K (1989) A concordance correlation coefficient to evaluate reproducibility. Biometrics 255–268

42. Shrout PE, Fleiss JL (1979) Intraclass correlations: uses in assessing rater reliability. Psychol Bull 86:420

43. Estévez PA, Tesmer M, Perez CA, Zurada JM (2009) Normalized mutual information feature selection. IEEE Trans Neural Networks 20:189–201

44. Amiri F, Yousefi MR, Lucas C, et al (2011) Mutual information-based feature selection for intrusion detection systems. J Netw Comput Appl 34:1184–1199

45. Ferguson TS (2000) Game Theory, Optimal Stopping, Probability and Statistics: Papers in Honor of Thomas S. Ferguson. IMS

46. Elgammal A, Duraiswami R, Harwood D, Davis LS (2002) Background and foreground modeling using nonparametric kernel density estimation for visual surveillance. Proc IEEE 90:1151–1163

47. Ngiam J, Koh P, Chen Z, et al (2011) Sparse Filtering. Nips 1–9

48. Dua, D. and Graff C (2019) UCI Machine Learning Repository. In: Irvine, CA Univ. California, Sch. Inf. Comput. Sci. http://archive.ics.uci.edu/ml. Accessed 7 Jan 2019

49. Basu S, Das N, Sarkar R, et al (2005) Handwritten ‘ Bangla ’ alphabet recognition using an MLP based classfier. In: 2nd National Conf. on Computer Processing of Bangla-2005. pp 285–291

Figures

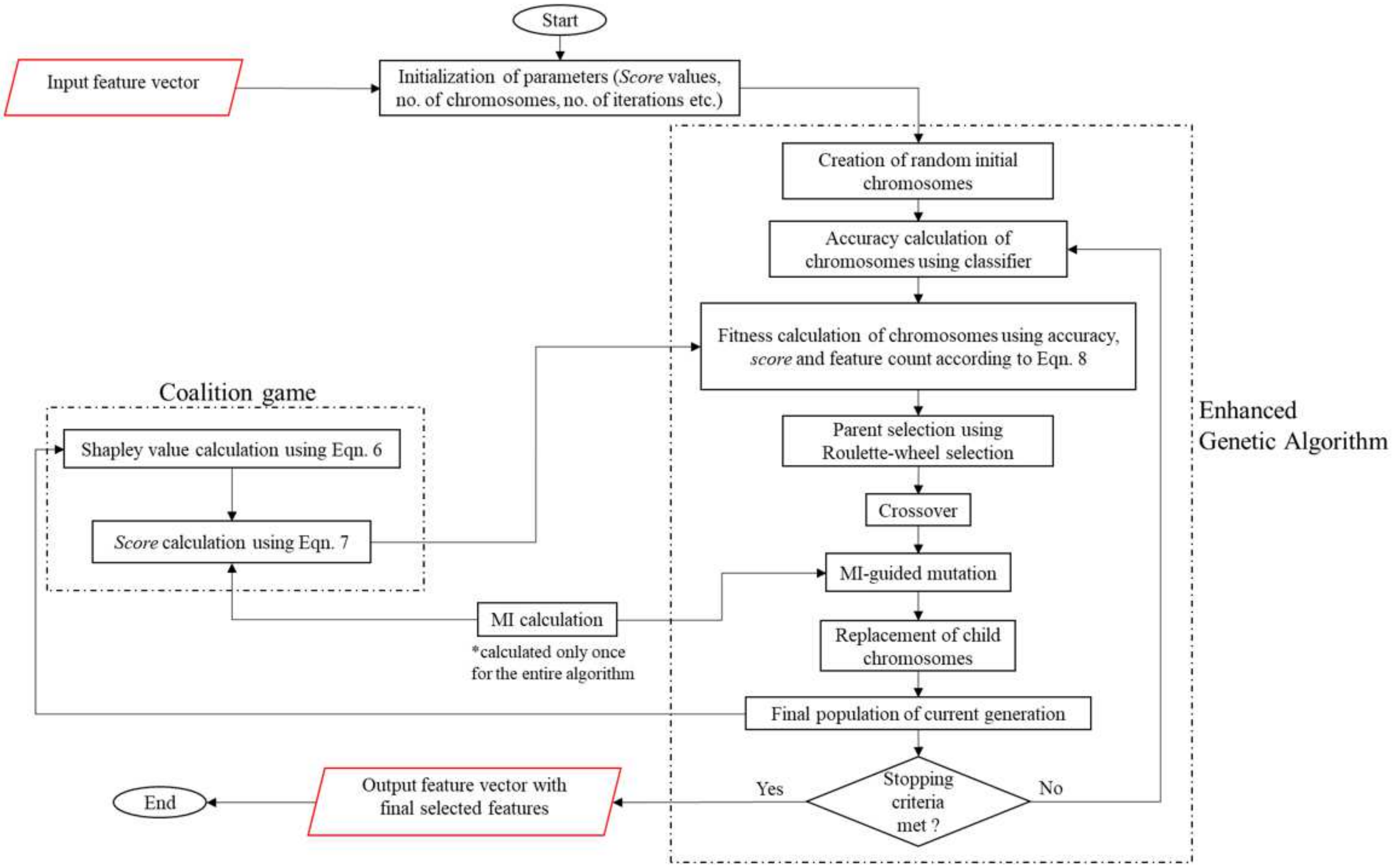

Start
Input feature vector
Initialization of parameters (*Score* values, no. of chromosomes, no. of iterations etc.)
Creation of random initial chromosomes
Accuracy calculation of chromosomes using classifier
Fitness calculation of chromosomes using accuracy, *score* and feature count according to Eqn. 8
Coalition game
Shapley value calculation using Eqn. 6
Score calculation using Eqn. 7
Enhanced Genetic Algorithm
Parent selection using Roulette-wheel selection
Crossover
MI-guided mutation
MI calculation
*calculated only once for the entire algorithm
Replacement of child chromosomes
Final population of current generation
Stopping criteria met ?
Yes
No
Output feature vector with final selected features
End

Figure 1

Flowchart of CGA model consisting of two different segments, namely EGA and Coalition game interacting with each other.

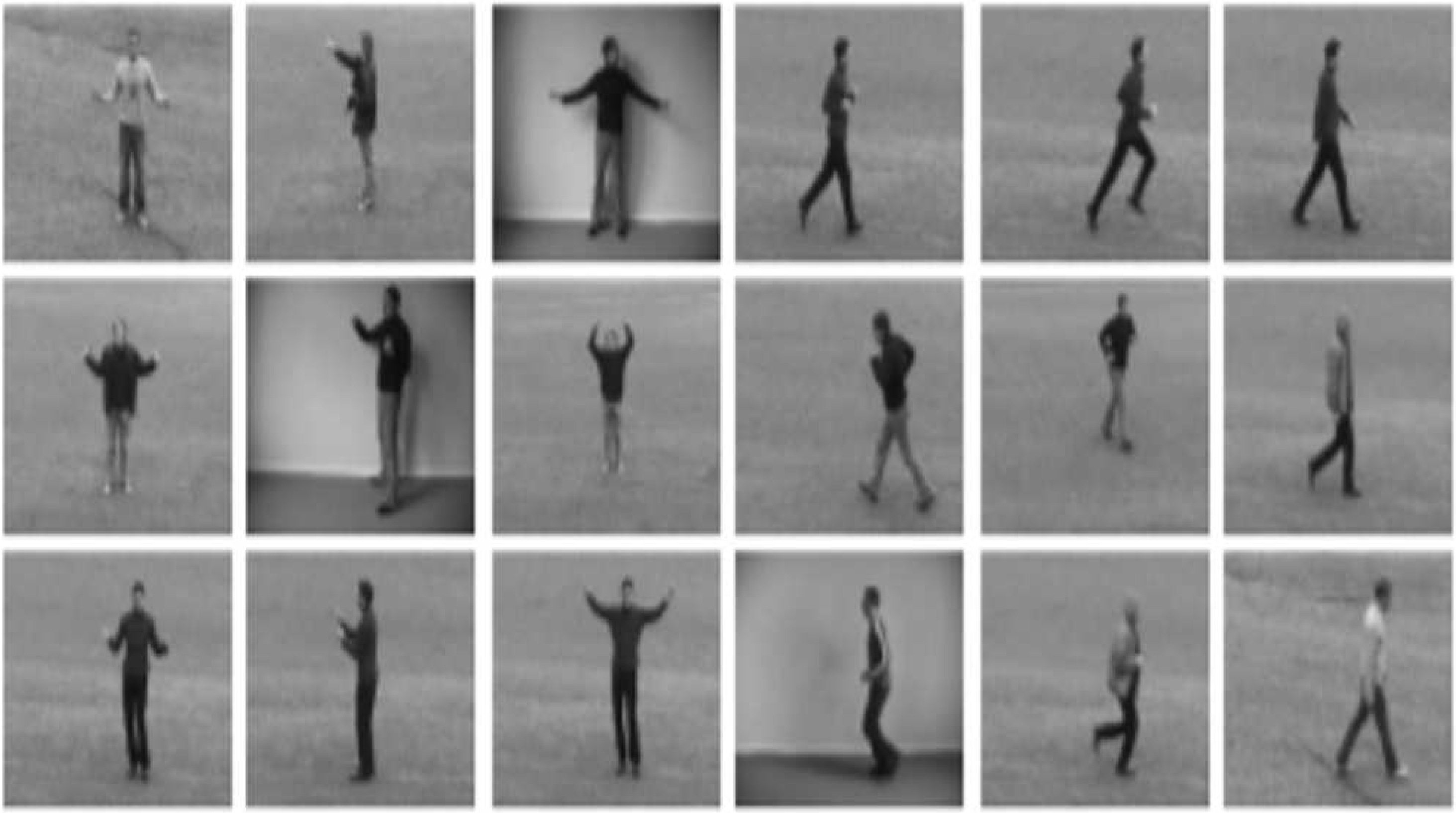


Figure 2

Sample frames from the KTH dataset

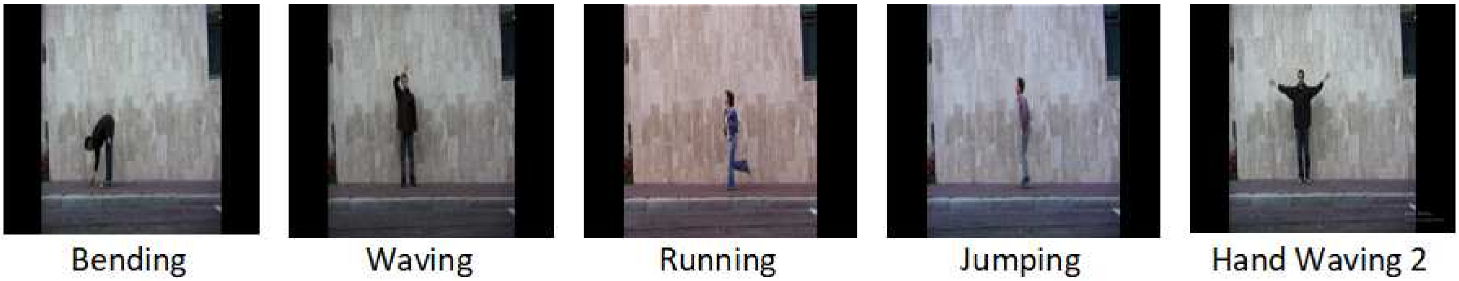


Figure 3

Sample frames from Weizmann dataset

Figure 4

Sample frames from the UCF 11 dataset

[redacted]

Figure 5

has been redacted due to potential copyright concerns.

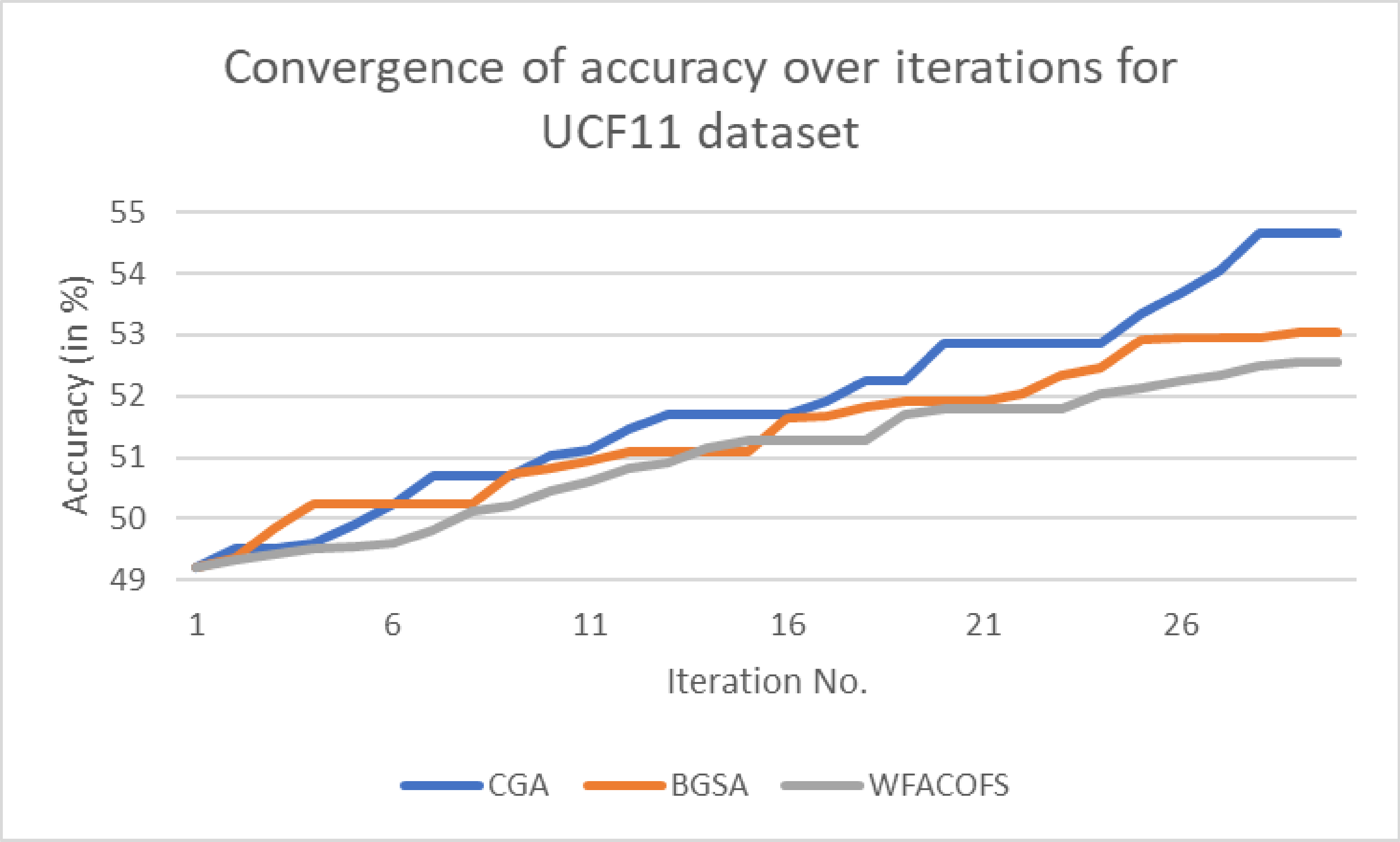


Figure 6

Convergence graph depicting Accuracy (using MLP-70) vs. Iteration no. for CGA, WFACOFS and BGSA over UCF11 feature set.

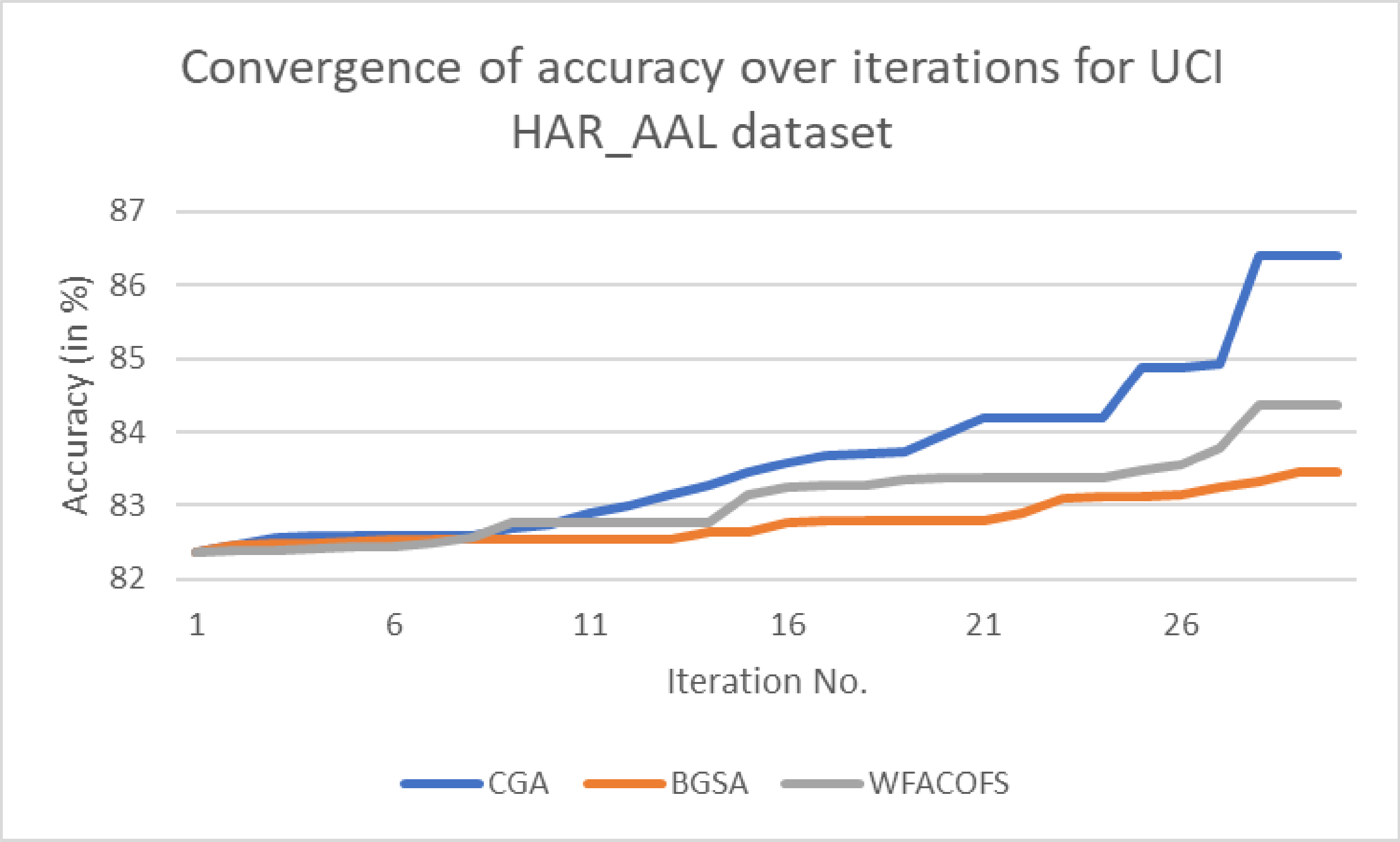


Figure 7

Convergence graph depicting Accuracy (using MLP-70) vs. Iteration no. for CGA, WFACOFS and BGSA over UCI HAR_AAL feature set.

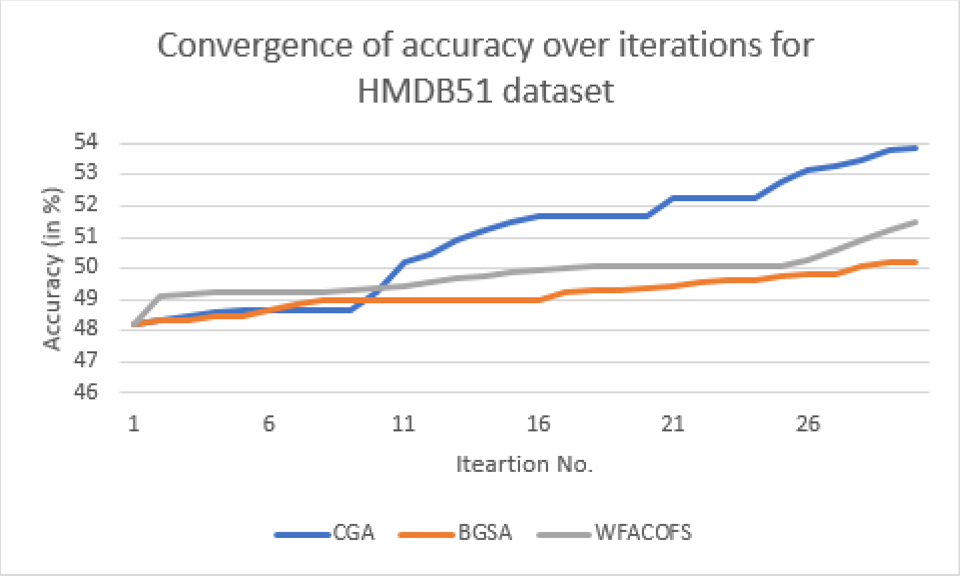


Figure 8

Convergence graph depicting Accuracy (using MLP-70) vs. Iteration no. for CGA, WFACOFS and BGSA over HMDB51 feature set.

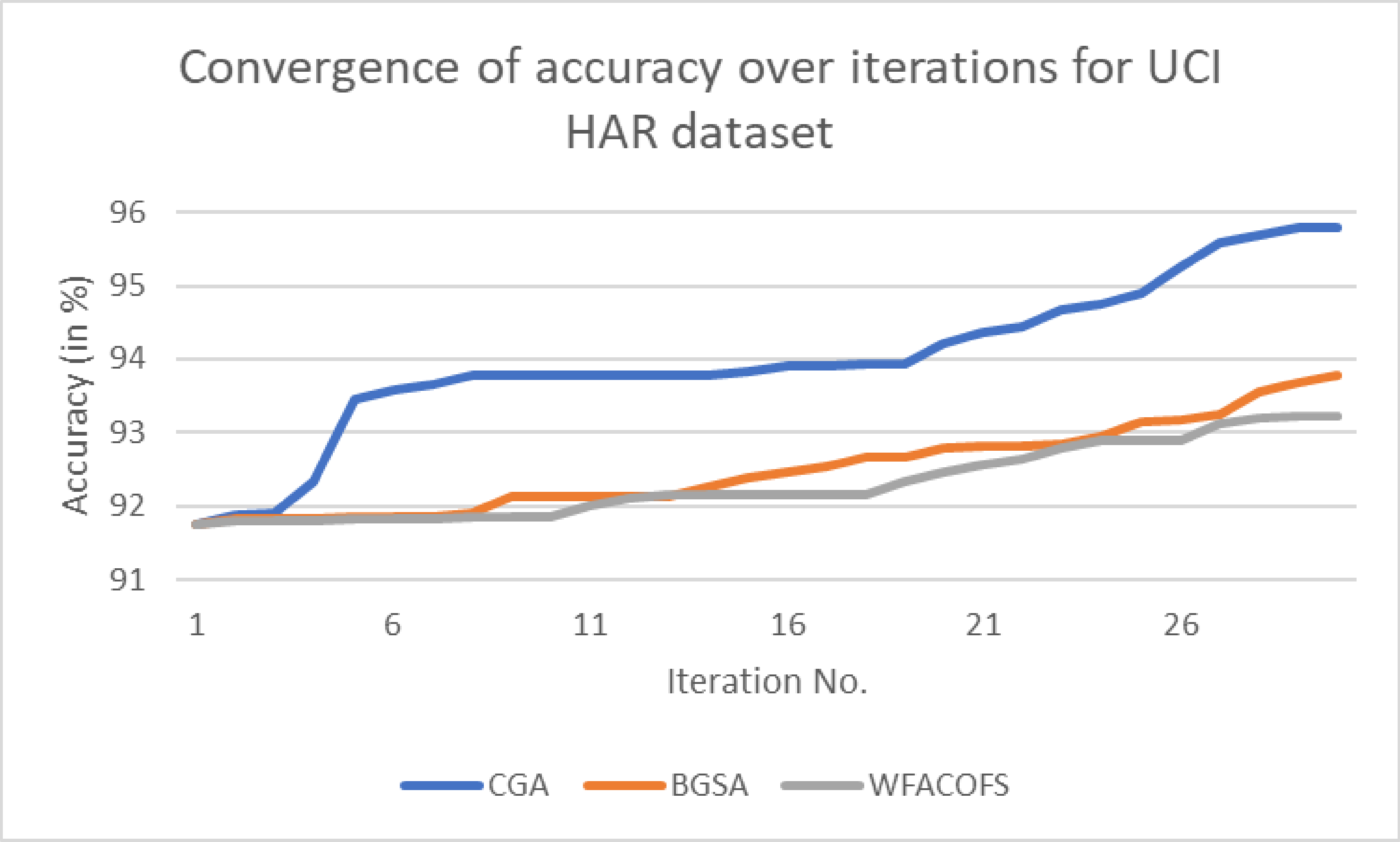


Figure 9

Convergence graph depicting Accuracy (using MLP-70) vs. Iteration no. for CGA, WFACOFS and BGSA over UCI HAR feature set.

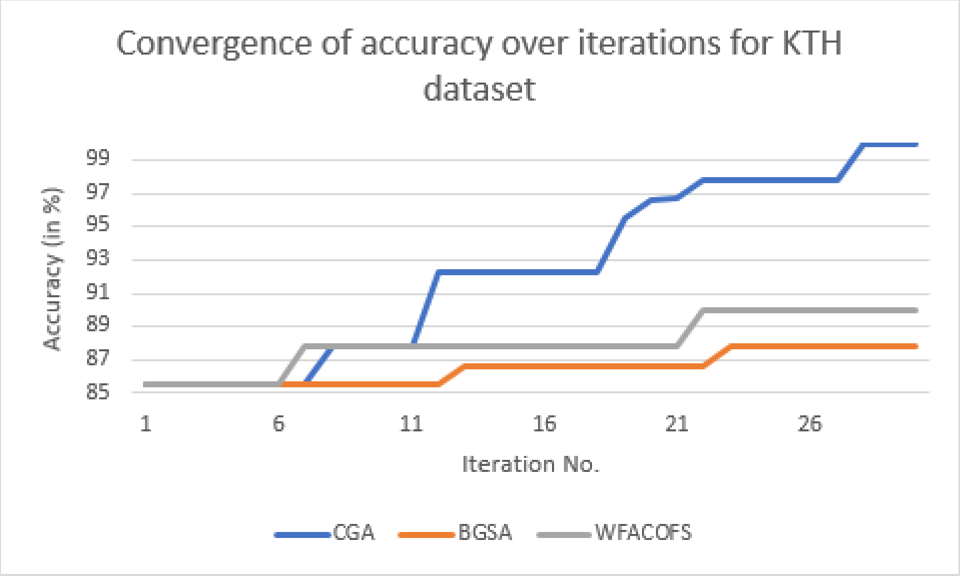


Figure 10

Convergence graph depicting Accuracy (using MLP-70) vs. Iteration no. for CGA, WFACOFS and BGSA over KTH feature set.

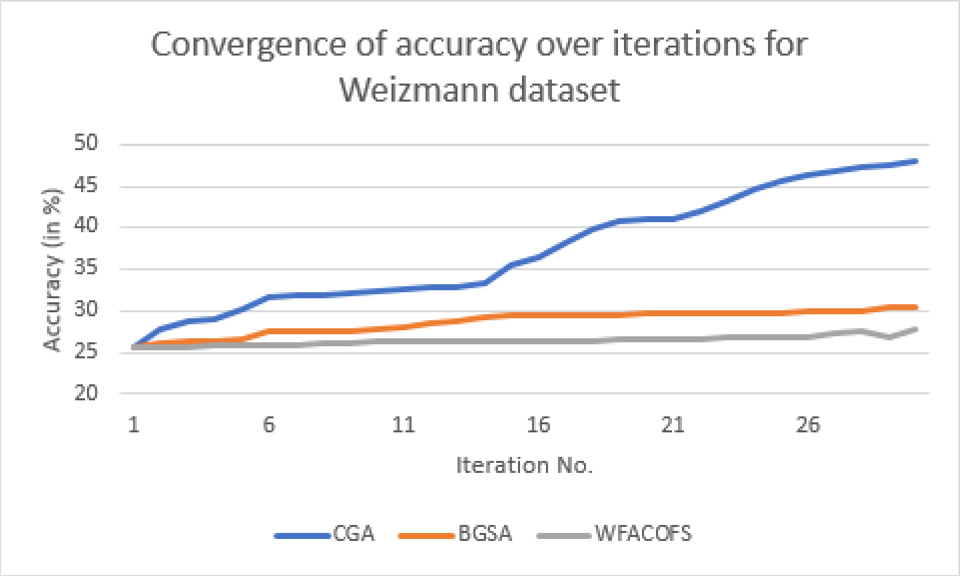


Figure 11

Convergence graph depicting Accuracy (using MLP-70) vs. Iteration no. for CGA, WFACOFS and BGSA over Weizmann feature set.