Optimizing Activity Recognition in Video Using Evolutionary Computation

Niraj Yagnik, Chethan Sharma*

Abstract—**We live in an era where the internet is flourishing with image and video data. Several algorithms and architecture have been devised, making most of such data and have been used to solve crucial problems. The number of features in image and video data can be extremely high, and such data can reach a dimensionality of thousands making the pre-processing step of feature selection extremely important. This work proposes the use of Evolutionary Computations to optimize the problem of Video Action Recognition or Classification. The VGG-16 architecture is used for extracting features from the images. The Binary Particle Swarm Optimization algorithm is devised to perform feature selection on the image frames extracted from the video. Two separate experiments are then performed to optimize hyper-parameter selections, using Particle Swarm Optimization and another Evolution Strategy. The robustness and consistency of the proposed methodology are tested on two popular datasets. The results obtained show that the optimized implementations using Evolutionary Algorithms perform much better than the traditional technique with no optimization.**

Index Terms—**Evolutionary Algorithms, Particle Swarm Optimization, Deep Learning, Feature Selection, Hyper-parameter Optimization.**

I. Introduction

IMAGES and videos have become an essential and omnipresent component of the current era of the internet. With this boom in the amount of visual content on the internet, there is an ever-increasing need for a much compelling urge to create efficient and accurate algorithms that make the most of such kinds of data to solve problems that affect our day to day lifestyles [1]. State-of-the-art Convolutional Neural Networks based architectures have been proved to be an efficient class of models to solve problems like image/video segmentation, classification, and retrieval and have been used in an array of subject areas and have been demonstrated to be highly reliable [2].

Among all these applications, video classification and activity recognition have proved to become a necessary application that aims at automatically classifying videos or activities into a particular label based on their contents extracted frame-wise. The problem itself is a natural extension of the image classification task where a video is segmented into multiple frames for easier extraction of features [3]. Video Classification and Activity Recognition serve as the initial step towards multimedia content understanding. It allows the machine to understand the content the video is trying to deliver without a manual examination.

Manuscript received April 01, 2021; revised September 01, 2021.

Niraj Yagnik is a final year student of the Department of Information and Communication Technology, Manipal Institute of Technology, Manipal Academy of Higher Education, Manipal, Karnataka, India. (email: nirajyagnik80@gmail.com)

Chethan Sharma is a Senior Assistant Professor in Department of Information and Communication Technology, Manipal Institute of Technology, Manipal Academy of Higher Education, Manipal, Karnataka, India. (Corresponding Author, e-mail: chethan.sharma@manipal.edu)..

Evolutionary Computation includes a class of algorithms inspired by the Darwinian principles of natural selection. They are a family of algorithms that employ a vast population or space of possibilities for optimal solutions to the problem. In Evolutionary Computing, an initial set of candidate solutions are defined and generated. Then, these candidates are optimized iteratively updated using a stochastic optimization character. Explaining in genetic terminology, a population of solutions is subjected to selection and mutation processes to get the desired set of solutions in each iteration [4]. A fitness function is used to guide these simulations of natural selection and mutations towards optimal solutions. Evolutionary computation techniques have proved to produce highly reliable optimized results for various problems, thus making them a go-to optimization technique in computer science.

Evolutionary Computation has found a significant number of applications in the field of Computer Vision and Deep Learning [5]. Genetic Computation based methods have been successfully applied to training neural networks to achieve great results. They are regularly used for obtaining the best set of hyper-parameters required for training the neural network [6]. Evolutionary Algorithms have proved to be a reliable technique for feature selection for large datasets, which can be used as an essential pre-processing step for traditional machine learning techniques and neural network-based architectures. Genetic Programming is being used and experimented with to solve problems like image denoising, image restoration, and image compression achieving good results. Neuroevolution is an exciting research area, which uses evolutionary algorithms to generate Neural Network-based architectures and their corresponding rules, parameters, and topology. It finds applications in interesting topics like artificial life, automating game playing, and robotics.

In this paper, the efficacy of Evolutionary Computation is augmented to optimize the problem of Video Activity Recognition. Activity recognition intends to recognize the actions of one or more agents or entities from a series of observations recorded at regular intervals. Images are fetched as frames at regular intervals, with each frame having thousands of feature values extracted using the Convolutional Neural Network. Particle Swarm Optimization is employed to perform feature selection from the images to fetch the feature values relevant to the problem and eliminate features that will not provide valuable information to the classifier. CNN (Convolutional Neural Network) is much dependent on the features selected, and the use of irrelevant features considerably dampens the accuracy of the designed model. Upon performing feature selection on the data, Particle Swarm Optimization and Evolution Strategy are used to conduct two separate experimentations to perform hyperparameter optimization and Neural Architecture Search using the refined features. The VGG-16 architecture [7] is used for the classification problem on the UCF-11 Dataset (YouTube Action Dataset) [8]. A parallel experiment is conducted on the Human Activity Recognition with Smartphones dataset, wherein the exact steps of the pipeline are repeated to test the consistency of the proposed work on multiple datasets.

The contributions of the paper include :

1) Design a robust pipeline to in-cooperate Evolutionary Computations to optimize Video Classification results using a Convolutional Neural Network.
2) Extend the power of Particle Swarm Optimization technique to perform feature selection on the Image Frame database to eliminate weak features.
3) Use Particle Swarm Optimization and Evolution Strategies for Hyperparameter Optimization.
4) The implementation which uses Particle Swarm Optimization for feature selection and hyper-parameter selection gave an accuracy score of 98.69%, boosting the accuracy score of the experiment with no EC (Evolutionary Computations) based optimization by 4.68% on the UCF-11 dataset [8].
5) The implementation which uses Particle Swarm Optimization for feature selection and hyper-parameter selection gave an accuracy score of 95.28%, boosting the accuracy score of the experiment with no EC (Evolutionary Computations) based optimization by 5.7% on the Human Activity Recognition with Smartphones dataset [9].

The remainder of the paper is organized as follows: Section 3 covers related work and literature survey of the topics, Section 4 discusses in detail the methodology of the proposed work, Sections 5 discusses the results obtained, and section 6 concludes the paper with a brief discussion on the future potential of the implementation.

II. Literature Survey and Related Work

A. Background

1) Convolutional Neural Network : Before we dive into the principal methodology and working of the proposed work, it is essential to understand the working of a Convolutional Neural Network [10] and how it has achieved state-of-the-art results in the field of computer vision. A convolutional neural network (CNN, or ConvNet) is a class of Neural Networks with single or multiple deep convoluted layers, followed by a single or multiple fully connected layers. The underlying architecture of CNN has been created to make the most out of 2D data like Images and Video Frames. They perform well with images, extracting important explicit and implicit information from a batch of images stored as an array as pixel values.

Convolutional Layers makes use of filters, which are generally a matrix to perform convolutional operations on the input image data to extract data of relevance. The filter slides horizontally and vertically to cover the image matrix's entirety and perform a scan on the whole image. The convolutional layer with the right choice of the activation function and corresponding pooling layer can achieve great results. CNN based architectures have found it is applications [11] in problems like image edge detection, image classification, object detection, and image segmentation.

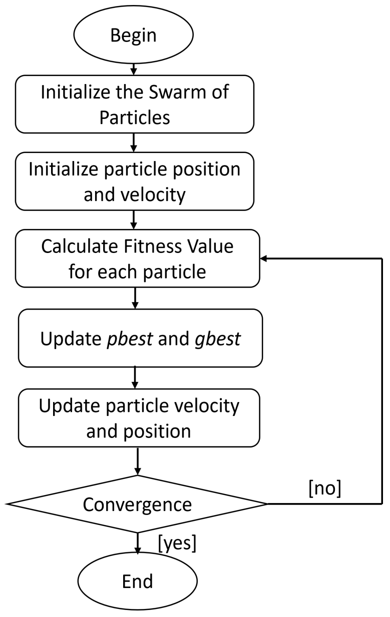
Begin
Initialize the Swarm of Particles
Initialize particle position and velocity
Calculate Fitness Value for each particle
Update *pbest* and *gbest*
Update particle velocity and position
Convergence
[no]
[yes]
End

Fig. 1. The Particle Swarm Optimization Algorithm.

2) Particle Swarm Optimization : Particle Swarm Optimization (PSO) [12] is a well-known genetically inspired algorithm that attempts to optimize a problem in hand by iteratively improving an initial candidate solution taking into account the fitness function in which each candidate solution fetches. The algorithm is based on swarm intelligence. It takes inspiration from swarm-like behaviours such as birds flying in a flock or a school of fish swimming in the ocean. PSO is a meta-heuristic and optimizes the problem by making minimal assumptions about the underlying conditions and parameters.

Each potential solution is called a particle. PSO initializes a swarm of such possible solutions or particles. Each such particle has a position and velocity associated with it. The algorithm moves these defined particles around in the search space over the particle's position and velocity. The particle best position (pbest) and group best position (gbest) influence the particle's velocity in the subsequent iteration. The algorithm aims to move the swarm towards the most optimized solutions.

PSO holds a significant advantage over other evolutionary algorithms. It has a significantly lower number of parameters to select and adjust, thus allowing the researchers to focus on the problem instead of optimizing the PSO parameters algorithm.

PSO finds its application in many optimizations related issues in the field of deep learning and data mining like hyperparameter optimization [13], and feature selection [14]. **Figure 1** illustrates the flow diagram of the PSO algorithm.

3) Evolution Strategies : Evolution Strategies [15] are a class of evolutionary algorithms that uses nature-inspired phenomena like mutation, selection, and recombination. These concepts are applied to a population of particles or individuals in the solution space. Steps are taken to evolve the candidate solution to optimize the fitness function selected for the individual solution.

Each iteration in the algorithm is called a generation; creating new generations is terminated if a point of convergence of fitness function is reached. Evolution Strategy mutates and combines the best individuals across the generation by using real values of positions and not converting them into an array of bits like genetic algorithms. **Figure 2** illustrates a general pipeline of evolutionary computation algorithms.

4) Feature Selection: In an era where data is flourishing, with tons of data generated every second from multiple sources, there is a need for efficient algorithms that can refine the data being collected. Feature Selection [16] techniques are employed as a preprocessor to several machine learning and deep learning architecture to eradicate any unnecessary features or columns in the dataset. The features that would not help the model gain new information and be a computational burden are eliminated. Thus, feature Selection is defined as refining the dataset and removing irrelevant features that will not add much to the learning or even decrease the dataset's accuracy and computation efficiency.

5) Hyperparameter Optimization: Parameters that define the model architecture and how it will perform are called hyperparameters. The optimization process of searching for the best set of hyperparameter values is referred to as hyperparameter optimization. The real intention of hyperparameter optimization techniques is to search for the best possible set of architecture parameter values, giving the most accurate results for the intended input data. Hyperparameter optimization [17] is being used on a general basis for problems using Machine Learning algorithms to push the model to its most optimized state.

B. Related Work

Zha et al. [18] conducts in-depth experimentation on the use of convolutional neural networks trained for image classification for event detection in videos. The work proposed in the paper performs studies on various techniques and selections like CNN layers, pooling layers, and normalization techniques. The result gives a commendable recognition performance on the UCF-101 dataset and sets a respectful benchmark for the problem.

Zuxuan Wu [19], gives a detailed discussion about all the state-of-the-art techniques in the research areas of video classification and video captioning. The accuracy and shortcomings of every method have been described briefly.

Dan [20], proposes a method that utilizes temporal and spatial features. The process obtained the descriptor of the video and extracted the Spatio-temporal features from the video. The data fetched is trained using a Support Vector Machine to generate generalized classification results on a small sample training set. The experiment generates a reasonable accuracy.

J. Liu [8], uses the UCF11 or Youtube action dataset to present a framework for recognizing real-life actions from the videos. Motion and static features are extracted from the videos using motion statistics. PageRank is selected as a medium to extract informative static features. All this extraction is followed by the use of AdaBoost to perform classification on the data collected.

B. Qolomany [13], uses the Particle Swarm Optimization technique to optimize parameters of deep learning architecture. PSO provides an efficient way of tuning and fetching the optimal number of hidden layers and the number of neurons in each layer required for the model training. In addition, the PSO optimization technique improves the initial accuracy value.

B. Xue [14], employs Particle Swarm Optimization to perform feature selection on the dataset. The paper discusses two implementations of PSO and how they can be utilized for feature selection. Experiments that involved feature selection using both the implementations of PSO perform better than the initial experimentation, which does not involve any feature selection.

Sun [21] uses genetic algorithms to automate CNN architecture design to address image classification tasks effectively. The algorithm proposed is validated on several widely used image classification datasets. Furthermore, the proposed algorithm outperforms the vanilla CNN architecture and outscores existing algorithms that can automate the CNN architecture design.

Q. Shen [22], employs genetic algorithms to obtain the most accurate architecture by optimizing the neurons' weights and the connections between the networks. The experiment improves the classification percentage and gives a satisfactory mean square error. Vieira [23] proposes a modified Binary PSO for feature selection and optimizing SVM kernel parameters by using it as a wrapper. The results outperform other PSO based feature selection techniques. Finally, young [24] uses Multi-node Evolutionary Neural Networks for Deep Learning (MENNDL) as a method for hyper-parameter optimization on computational clusters.

Wei-Zhong Sun [25] proposes a method of colony image feature extraction and essential dimension reduction based on digital image processing technology. A colony indicates a group of microorganisms generated and produced by the growth, mutation and reproduction of a single microorganism species. Texture features of colony images are extracted along with their correlation dimension estimator.

Wei-Zhong Sun [26] proposes a binary particle swarm optimization (BPSO) algorithm based on the Z-shaped probability transfer function to solve the 0-1 knapsack problem. A penalty function strategy is employed to deal with the violation of the constraint solutions. The simulation experiments show that the proposed method improves the convergence speed and optimization accuracy of the BPSO algorithm.

Islam T. Elgendy [27] presents a GA-based approach and supporting tool for data-flow test data generation for web applications developed in the ASP.NET framework. The employed GA conducts its searches by constructing new test data using previously generated test data that have been validated as effective data. Chromosome in this GA is a collection of user interface control objects, and each control is considered a gene. Chethan et.al [28] presents a comparative analysis of various object detection algorithms on Video dataset.

III. Materials and Methods

A. Dataset

The proposed work makes use of the Youtube Action Dataset [8] (also called the UCF11 dataset). The dataset contains videos belonging to 11 action categories, which are basketball shooting, biking, diving, golf swinging, horse riding, soccer juggling, swinging, tennis swinging, trampoline jumping, volleyball spiking, and walking with a dog.

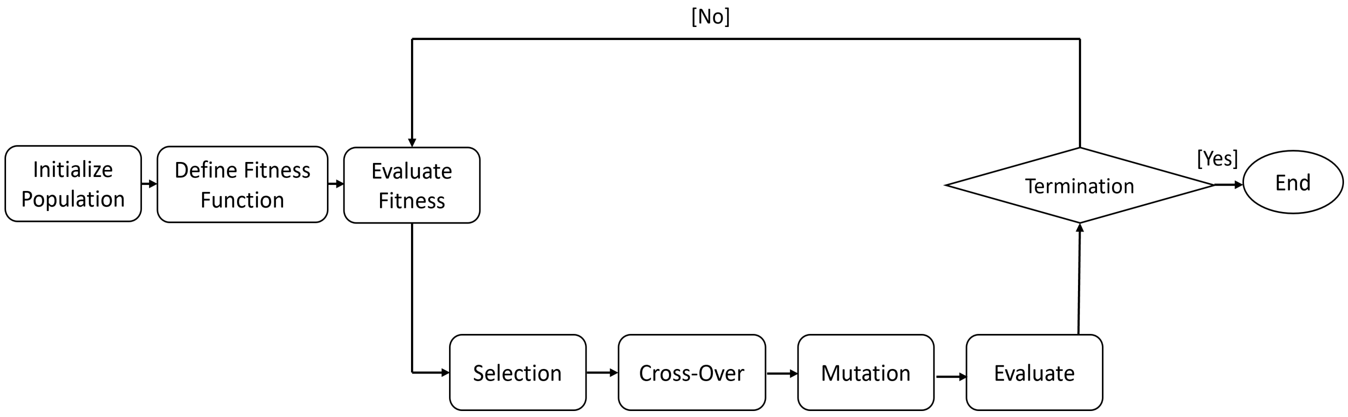

Fig. 2. Flowchart of Evolutionary Algorithms.

The videos belonging to the same category in the dataset are grouped into 25 with more than four action clips about the class. The video clips belonging to the same group share similar defining features such as similar backgrounds, similar viewpoints, and the same actors.

The efficacy of the proposed methodology is parallelly tested on the Human Activity Recognition with Smartphones dataset[9]. The database is built from the recordings of 30 study volunteers performing activities of daily living (ADL) while handling a smartphone with inertial sensors attached to their waist. The goal is to categorize each activity into one of the six categories. The activity performed by the volunteer is captured using the embedded gyroscope and accelerometer of the sensor. 3-axial linear acceleration and 3-axial angular velocity of the actions performed captured at a steady rate of 50Hz.

B. Preprocessing

Each video in the database is transformed into images. These images are retrieved from the video [18] at a preset frame rate. Each image frame is associated with the class label corresponding to the movie from which the frames are being retrieved at a constant frame rate. A data frame is created with a path to the frames and the class labels as the two columns, which will be utilized for the classification problem. **Figure 3** illustrates the preprocessing step on a single input video.

C. Feature Selection Using Binary Particle Swarm Optimization

PSO (Particle Swarm Optimization) is an Evolutionary Computation based optimization algorithm inspired by natural concepts like fish-schooling and bird flocking. It has garnered immense popularity in research for its efficacy to perform feature selection on large datasets.

For Binary PSO, the position of the particles is expressed as a binary, either 1 or 0 (on or off). The position of the particle for each dimension of the dataset is seen as either on or off. Given a dataset with d features, each feature is assigned as a dimension of a particle. Thus upon initialization of the particles, the BinaryPSO (Binary Particle Swarm Optimization) is implemented. The best positions obtained as a binary array can be used to interpret whether or not a feature should be considered for training or not.

An objective function f is defined in Eq. (1), which maps the search space to the function space [23]. **Figure 1** gives a pictorial representation of the PSO algorithm.

$$f(x) = \alpha(1-P) + (1-\alpha)\frac{Nf}{Nt} \qquad (1)$$

Where :
P is the performance of the classifier.
Nf is the size of the feature subset.
Nt is the total number of features in the dataset before any feature selection is initiated.
α is a hyperparameter that influences the tradeoff between P and N.
Algorithm 1 indicates the pseudo code for PSO. **Algorithm 2** demonstrates the pseudo code for Feature Selection using PSO [29] [30].

Algorithm 1 PSO Algorithm

```
1: Initialize the position $x$ and velocity $v$ of each particle
2: Fitness Function Initialization
3: initialize pbest and gbest
4: while stopping criterion not met do
5:     if fitness $x_i > pbest_i$ then
6:         $pbest_i = x_i$
7:     end if
8:     if fitness $pbest_i > gbest_i$ then
9:         $gbest_i = pbest_i$
10:    end if
11:    Update Velocity of partile $i$
12:    Update Position of particle $i$
13: end while
14: return Return $gbest$ and its fitness values.
```

Algorithm 2 Feature Selection Using PSO

```
Require: Image Matrix input $x$, output label $y$
1: dimension ← dimension of array x
2: classifier ← initialize supervised classification model
3: f ← define objective function using accuracy metric of
   the classifier
4: Initialize Swarm with features as particles
5: Call Instance of PSO
6: Initiate Optimization using PSO for input number of
   generations
7: return Retained Features
```

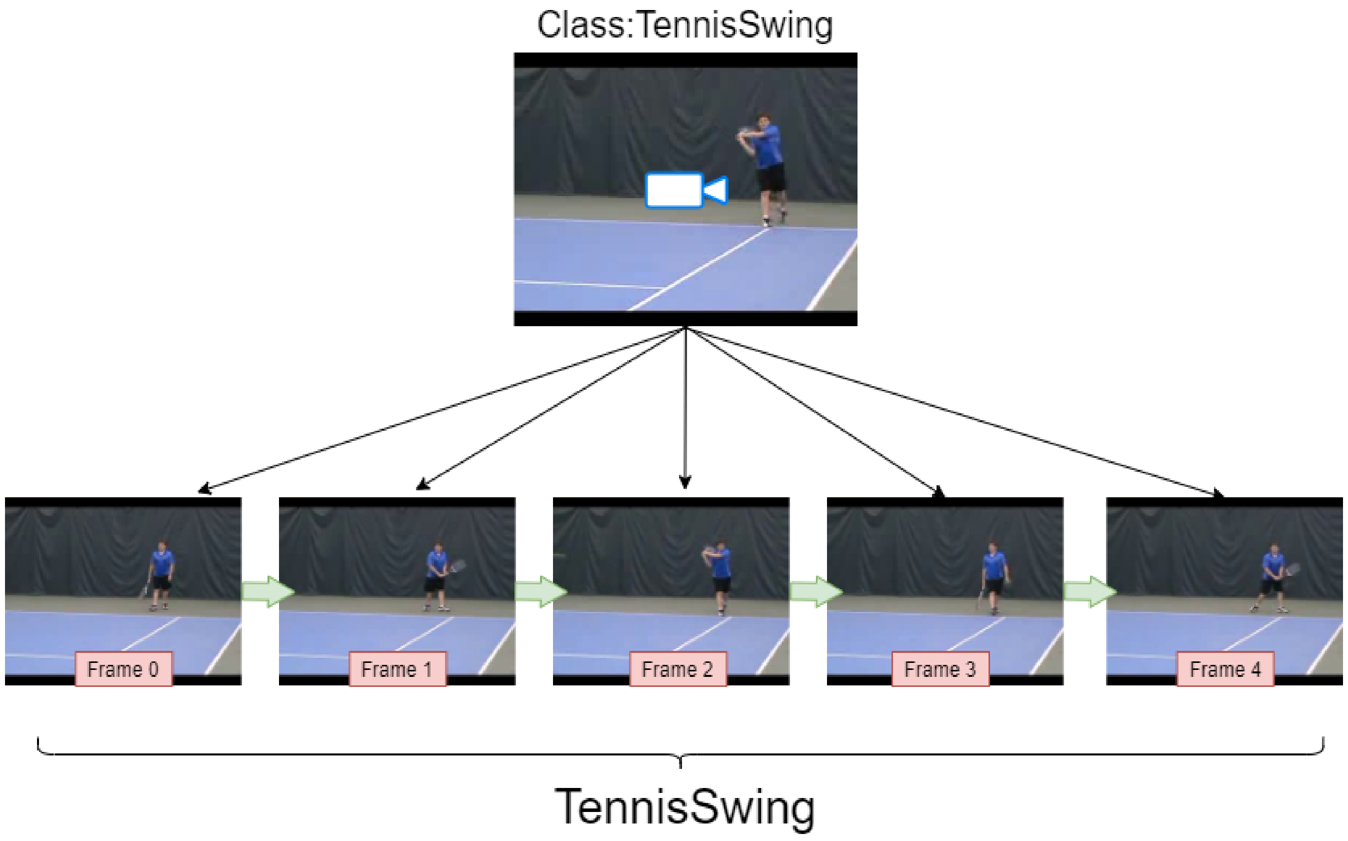

Fig. 3. Illustration of the Preprocessing step on a single video.

The accuracy score obtained by the classification model is used as the fitness function to evaluate the refined feature set as a figure of merit. The hyperparameters are selected after performing hyperparameter testing. Hardware resource availability is taken into consideration while running the experiments. 200 particles and 10 generations are considered for the experiments. An alpha value of 0.88 is finalised as input. The feature count for every iteration is measured and documented to check for consistency.

D. Hyper-parameter Optimization

Hyperparameter Optimization has become an essential part of the current machine learning pipeline. Several optimization techniques are used to select the hyperparameters, which can train the best architecture for a given data set. Hyperparameter Optimization using Search Optimization Techniques suffer when the dimensionality is vast, and failure of one job will lead to failure of subsequent jobs.

Two parallel experiments are conducted in the proposed work to perform hyperparameter optimization for the architecture. The first one uses PSO as the optimizer, and the second augments the power of ES(Evolution Strategies). The mentioned two techniques optimize these hyperparameters: the number of hidden layers in each layer, the learning rate, and the choice activation and

Each of these hyper-parameters is encoded as a single gene for each individual [24]. The work defines the fitness function, which will be used to evaluate individuals at each generation. A range of values is provided for each hyperparameter to be optimized to restrict the search space. Each gene for the initial population is sampled from a uniform random distribution. Parallel computations for PSO based and Evolution Strategies are implemented. Evolution Strategy has been illustrated in **Figure 2**.

The initial population is evolved, and a new generation is formed using selection, mutation, and crossover using individuals from the previous generation with the best fitness scores. The hyperparameters obtained at the last generation are fed into the model to give the best possible accuracy score with the given architecture and value range.

PSO optimization performs hyperparameter optimization by initializing a number of positions simultaneously at the same time. These positions are moved closer to the best position iteratively using an evolutionary process to give the best solutions.

In Hyperparameter Optimization using Evolution Strategies, mutation and combination is performed on the best individuals of a population in each generation without transforming their values into an array of bits and maintaining the real values of the positions.

Algorithm 3 demonstrates pseudo-code for Hyperparameter optimization using ES [31]. The initial population of individuals is first generated based on the input range selected for the experiments. The best individuals are refined in each iteration based on the objective function defined after mutations and crossovers. Upon termination, the best individuals will be returned as the ideal hyperparameters for the designed neural network architecture to work with. The process is repeated for every experiments conducted for multiple dataset to ensure consistency and robustness.

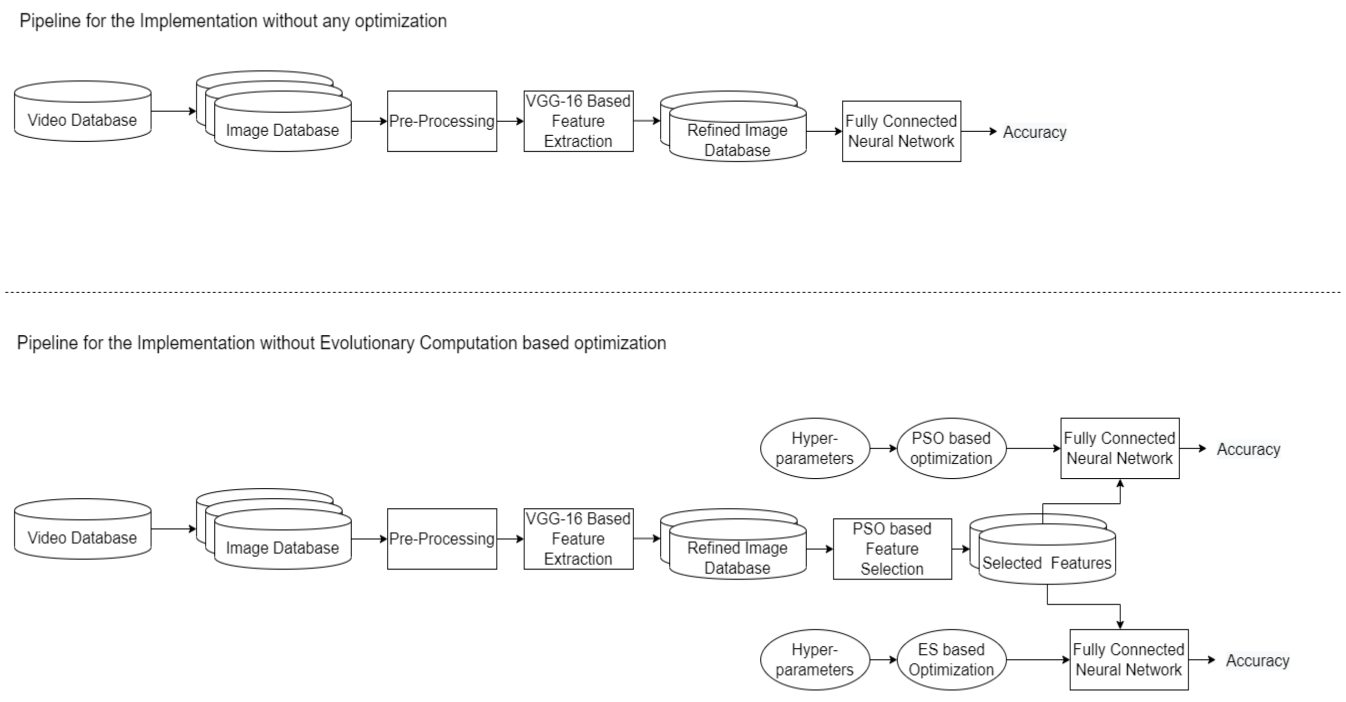
Pipeline for the Implementation without any optimization
Video Database
Image Database
Pre-Processing
VGG-16 Based Feature Extraction
Refined Image Database
Fully Connected Neural Network
Accuracy
Pipeline for the Implementation without Evolutionary Computation based optimization
Video Database
Image Database
Pre-Processing
VGG-16 Based Feature Extraction
Refined Image Database
PSO based Feature Selection
Selected Features
Hyper-parameters
PSO based optimization
Fully Connected Neural Network
Accuracy
Hyper-parameters
ES based Optimization
Fully Connected Neural Network
Accuracy

Fig. 4. Illustration of the Preprocessing step on a single video.

Algorithm 3 Evolution Strategy Algorithm for Hyperparameter Optimization

```
Require: input: size of the parent population $\mu$ , size of the offspring $\lambda$
 1: calculate objective function (OF)
 2: no of generations n← 0
 3: evaluate P(t) using OF
 4: randomly generate $\mu$ inidividuals in initial population of the hyperparameter based on the input range
 5: while not terminated do
 6:     n = n + 1
 7:     create population T(t) by reproducting $\lambda$ individuals from population P(t-1)
 8:     using crossover and mutation, create population M(t)
 9:     evaluate individuals in population M(t)
10:     select $\mu$ the best individuals to Population P(t) from M(t) and P(t-1) Populations
11: end while
12: return best individual population of Hyperparameter values P(t)
```

E. Implementation

The images and the corresponding labels obtained are processed to make them fit for training. The work augments the power of the VGG-16(Visual Geometry Group-16) [7] pre-trained model to extract [32] implicit and explicit features from the image dataset. VGG16 is CNN based architecture for image recognition, which the Visual Geometry Group proposed at the University of Oxford.

The architecture has 16 weight layers and an output layer for softmax prediction on 1000 classes. Thus the architecture constituting the input layer and the max-pooling layer is used as the feature extractor of the model, with the final layers being for the final prediction.

Upon extraction from the VGG-16 architecture, the features are passed into the fully connected neural network classifier, which consists of 5 hidden layers and an output softmax function without performing any hyperparameter optimization or feature selection. **Figure 4** shows the pipeline for implementation.

After implementing classification without using any optimization, the work employs the power of evolutionary algorithms to improve the efficiency of the implementation. BinaryPSO is introduced after performing feature extraction using the VGG-16 model to perform feature selection and only retain the relevant features and eradicate the irrelevant features which could hinder the classifier's performance. Two separate experiments are conducted to perform hyper-parameter optimization, one using PSO and another using the Evolution Strategy. The best set of hyper-parameters fetched after these optimizations is used to design the Fully Connected Neural Network classifier. The results obtained are showcased in the next section.

All the steps are repeated for UCF-11 and Human Activity Recognition with Smartphones datasets to comprehensively compare the proposed work against the vanilla classification pipeline.

IV. Results and Discussions

A. Feature Selection

BinaryPSO (BPSO) based feature elimination is used on the features extracted from the pre-trained VGG-16 architecture. BinaryPSO is run on the feature set 10 times to get a general idea of the size of the refined feature set. Results obtained for UCF-11 dataset are illustrated in **Table I** and **Figure 5**. Results obtained for Human Activity Recognition WithSmart Phones Dataset are illustrated **Table II** and **Figure 6**.

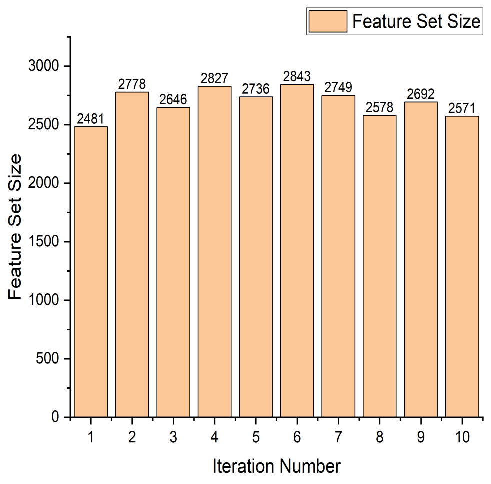

Fig. 5. Feature Size over 10 iterations running BinaryPSO - UCF-11 Dataset.

TABLE I
FEATURE SIZE AFTER RUNNING BINARYPSO FOR FEATURE SELECTION - UCF-11 DATASET.

Initial	Mean	Maximum	Minimum
4608	2690	2843	2481

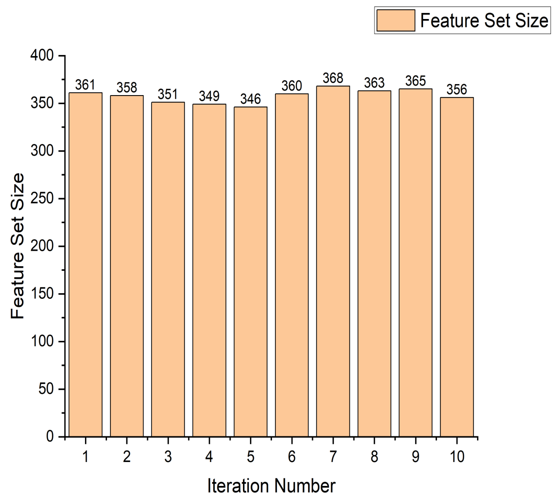

Fig. 6. Feature Size over 10 iterations running BinaryPSO - Human Activity Recognition WithSmart Phones Dataset.

TABLE II
FEATURE SIZE AFTER RUNNING BINARYPSO FOR FEATURE SELECTION - HUMAN ACTIVITY RECOGNITION WITHSMART PHONES DATASET.

Initial	Mean	Maximum	Minimum
562	358	368	356

B. Hyperparameter Optimization

The proposed work uses PSO and ES to find the best values of the hyperparameters Activation, the number of hidden layers, model optimizer, and CNN Block architecture. **Table III** illustrates the best hyperparameter values obtained for the UCF-11 Dataset.**Table IV** illustrates the best hyperparameter values obtained for the Human Activity Recognition with Smartphones Dataset.

TABLE III
HYPERPARAMETER OPTIMZATION RESULTS - UCF-11 DATASET

Hyper-paramater	Input Values	PSO Output	ES Output
Activation	['relu',' tanh']	'relu'	'relu'
Hidden Layers	[16,32, 64, 128,256, 512, 1024, 2048]	256	128
Model Optimizer	['SGD', 'Adam']	'Adam'	'Adam'
CNN Block	['With Dropout', 'Without Dropout']	'With Dropout'	'With Dropout'

TABLE IV
HYPERPARAMETER OPTIMZATION RESULTS - HUMAN ACTIVITY RECOGNITION WITH SMARTPHONES DATASET

Hyper-paramater	Input Values	PSO Output	ES Output
Activation	['relu',' tanh']	'tanh'	'relu'
Hidden Layers	[16,32, 64, 128, 256, 512]	64	128
Model Optimizer	['SGD', 'Adam']	'Adam'	'Adam'
CNN Block	['With Dropout', 'Without Dropout']	'With Dropout'	'Without Dropout'

C. Metrics

Eqs. (2), (3), (4) and (5) illustrate the equations of the metrics used to measure the performance of the implementation.

$$Accuracy = \frac{Number\ of\ Correct\ Prediction}{Total\ number\ of\ prediction} \quad (2)$$

$$Precision = \frac{True\ Positives}{True\ Positives\ +\ False\ Positives} \quad (3)$$

$$Recall = \frac{True\ Positive}{True\ Positive\ +\ False\ Negatives} \quad (4)$$

$$F1Score = 2 \times \frac{Precision\ *\ Recall}{Precision\ +\ Recall} \quad (5)$$

D. Classification Results

1) UCF-11 Dataset: The implementation, devoid of any EC(Evolutionary Computation) based optimization, gives a respectable accuracy score of 93.50% (0.9350) and an F1 score of 0.9392. However, the implementation, which involves EC-based computation, significantly outperforms the performance, which does not include any optimization. PSO found hyper-parameter optimization on the model being trained on data that undergoes BinaryPSO based feature selection gave an accuracy score of 98.33% (0.9833) and an F1-Score of 0.9834. At the same time, Evolution Strategy based hyper-parameter optimization on the model being trained on data that undergoes BinaryPSO based feature selection gave an accuracy score of 97.76% and an F1-Score of 0.9771.

Figure 7 and **Figure 8** illustrates bar graphs for the results obtained. **Table V** showcases the same results in a tabular format.

TABLE V
MODEL PERFORMANCE COMPARISON - UCF11 DATASET.

Implementation	Accuracy	F1 Score	Precision	Recall
Vanilla Implementation	0.9350	0.9316	0.9579	0.9075
PSO Based Optimization	0.9833	0.9834	0.9850	0.9818
ES Based Optimization	0.9776	0.9771	0.9774	0.9769

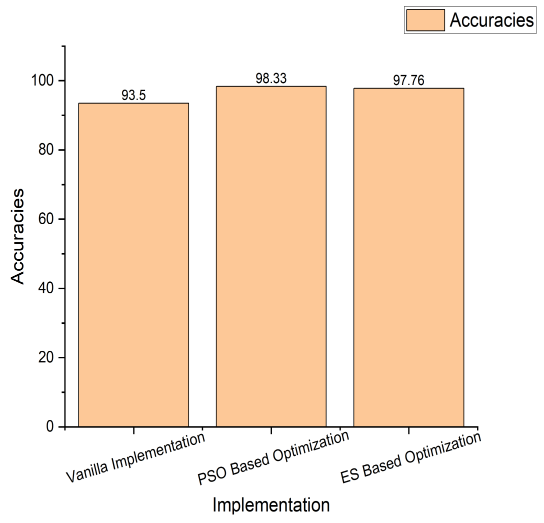

Fig. 7. Model Accuracy Comparison - UCF-11 Dataset.

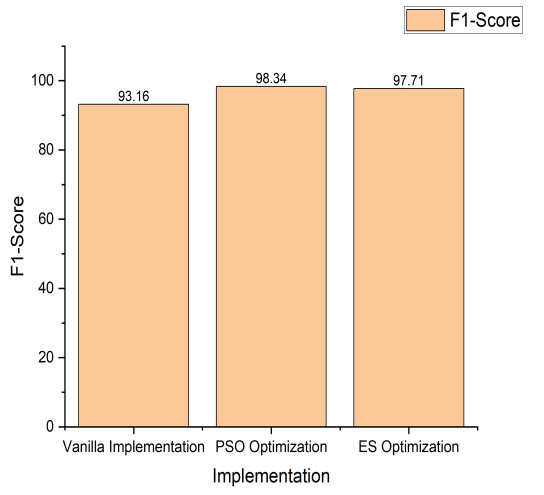

Fig. 8. Model F1-Score Comparison - UCF-11 Dataset.

2) Human Activity Recognition With Smart Phones Dataset: The implementation, devoid of any EC(Evolutionary Computation) based optimization, gives an accuracy score of 89.58% (0.8958) and an F1 score of 0.8904. However, the implementation, which involves EC-based computation, significantly outperforms the performance, including optimization. PSO found hyper-parameter optimization on the model trained on data that undergoes BinaryPSO based feature selection gave an accuracy score of 95.28% (0.9528) and an F1-Score of 0.9544. Similarly, Evolution Strategy based hyper-parameter optimization on the model trained on data that undergoes BinaryPSO based feature selection gave an accuracy score of 94.40% and an F1-Score of 0.9450.

Figure 9 and **Figure 10** illustrates bar graphs for the results obtained. **Table VI** showcases the same results in a tabular format.

TABLE VI
MODEL PERFORMANCE COMPARISON - HUMAN ACTIVITY RECOGNITION WITH SMART PHONES DATASET.

Implementation	Accuracy	F1 Score	Precision	Recall
Vanilla Implementation	0.8958	0.8904	0.8889	0.8919
PSO Based Optimization	0.9528	0.9544	0.9549	0.9538
ES Based Optimization	0.944	0.9450	0.9448	0.9451

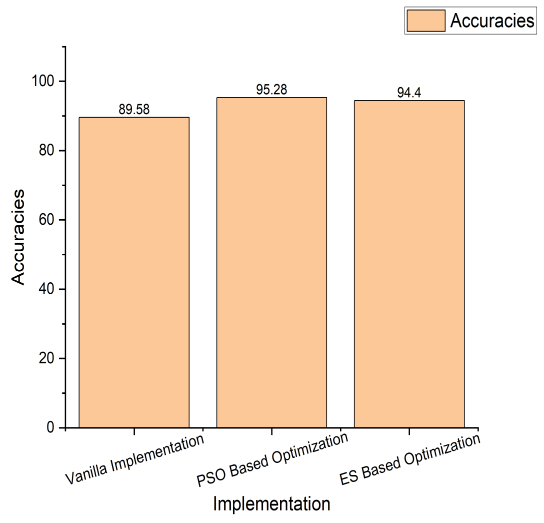

Fig. 9. Model Accuracy Comparison - Human Activity Recognition With Smart Phones Dataset.

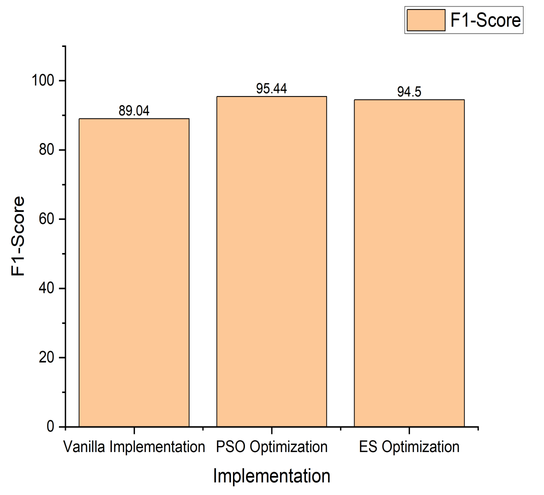

Fig. 10. Model F1-Score Comparison - Human Activity Recognition With Smart Phones Dataset.

E. Statistical Testing

A series of statistical tests are performed on the results obtained to test the hypothesis that the increase in model performances after employing Evolutionary Computation techniques as the optimizer is statistically significant. K-Fold Cross Validation with 10 folds is created to create a population for the tests and ensure that less biased models are created as the technique ensures that each data point in the database is involved in the training process. Thus K-Fold cross-validation is employed to check for model consistency and create the sample space to test the statistical significance of the optimized models.

1) Independent samples t-test: There is a significant difference in the scores of the models trained using PSO Based Optimization technique (Mean = 0.98603, Standard Deviation = 0.00359) compared to the models trained using vanilla CNN(Mean = 0.93575, Standard Deviation = 0.00359). Independent samples t-test gives a very high t-value of 23.0359 and an infinitesimally small p-value eliminating the null hypothesis that the accuracy increase achieved is not statistically significant.

Similarly, there is a significant difference in the model accuracy scores of models trained using ES Based Optimization technique (Mean =0.9849, Standard Deviation = 0.003887) compared to the models trained using Vanilla CNN(Mean = 0.93575, Standard Deviation = 0.00359). Independent samples t-test giving a very high t-value of 22.0282 and an infinitesimally small p-value.

2) Wilcoxon Signed-Rank Test: Wilcoxon Signed-Rank Test between the scores of models trained using PSO Based Optimization technique (Mean = 0.98603, Standard Deviation = 0.00359) and the scores of models trained using vanilla CNN(Mean = 0.93575, Standard Deviation = 0.00359) gives a low p-value of 0.005. Thus eliminating the hypothesis that the result population samples are fetched from the same distribution.

Similarly, the test also returns a low p-value of 0.005 when run using the model scores of ES Based Optimization technique (Mean =0.9849, Standard Deviation = 0.003887) against the scores of models trained using vanilla CNN(Mean = 0.93575, Standard Deviation = 0.00359)—indicating a noticeable significant difference in the model scores.

F. Discussion

Models trained using feature set distilled by Particle Swarm Optimization performed better than the model trained on the original features with the model whose hyperparameters were optimized using PSO performed marginally better than the ES based technique. The models are tested on two datasets and tested using multiple metrics to verify the pipeline's consistency, with all the metrics favouring the models optimized using Evolutionary Computation. The significance of the boost in the model scores post Evolutionary optimizations is also tested, indicating a noticeable significance.

PSO and ES are both population-based search approaches, which rely heavily on the information shared among the population members to enhance the searching process. PSO was successful in eliminating weak features from the feature space. In addition, PSO and ES were successful in selecting the best set of hyperparameters.

V. Future Potential

The proposed work leaves scope for plenty of future work potential. For example, a different Convolutional Neural Network Model can be used instead of VGG-16 used in the proposed work for frame prediction, thus conducting a comparative study.

Different classifier algorithms can be used for calculating the fitness function for PSO in the feature selection algorithm. A comparative study can be conducted for every classification model used, indicating how the classification model selection can influence the optimization. The pipeline can be tested on the much larger UCF-101 dataset to verify the robustness and reliability of the technique.

VI. Conclusion

The results obtained showcase the power of Evolutionary Computation techniques for optimizing Deep Learning Architectures. PSO based feature selection is an efficient technique and does a great job capturing relevant features and eradicating irrelevant features. Furthermore, PSO and ES (Evolutionary Strategy) based model selection and hyperparameter optimization do an excellent job of generating the best set of hyperparameters, which would allow the model to operate at its full efficacy. This proves the power and efficacy of Genetic algorithms and evolutionary computations techniques to improve the accuracy and performance of neural network-based architecture.

These techniques can be used in several problem statements like Object Detection, Image Segmentation, and Edge Detection. Feature Selection techniques involving genetic algorithms can be used for all these applications to get the most of the vast input image data. The optimization used in the proposed work can also be extended to data involving texts and relational databases to solve fraud detection and sentiment analysis problems.

References

[1] M. S. Nadeem, V. N. L. Franqueira, X. Zhai and F. Kurugollu, ”A Survey of Deep Learning Solutions for Multimedia Visual Content Analysis”, IEEE Access, vol. 7, pp84003-84019, 2019.

[2] Sharma, Neha & Jain, Vibhor & Mishra, Anju, ”An Analysis Of Convolutional Neural Networks For Image Classification,” Procedia Computer Science, 132, pp377-384, 2018.

[3] Liam Hiley, Alun Preece, Yulia Hicks, ”Explainable Deep Learning for Video Recognition Tasks: A Framework & Recommendations,” arXiv:1909.05667

[4] Carlos A. Coello Coello, ”An introduction to evolutionary algorithms and their applications,”. In Proceedings of the 5th international conference on Advanced Distributed Systems (ISSADS'05). Springer-Verlag, Berlin, Heidelberg, pp 425–442, 2005.

[5] Alejandro Baldominos et al. “Hybridizing Evolutionary Computation and Deep Neural Networks: An Approach to Handwriting Recognition Using Committees and Transfer Learning.” Hindawi,Complexity,pp 1-16, 2019

[6] Orive, D. & Sorrosal, G. & Borges, Cruz & Martin, Cristina & Alonso-Vicario(2014), ”Evolutionary algorithms for hyperparameter tuning on neural network models,”. In: 26th European Modeling and Simulation Symposium, EMSS (2014). pp402-409, 2014.

[7] Simonyan, Karen & Zisserman, Andrew, ”Very Deep Convolutional Networks for Large-Scale Image Recognition”,arXiv:1409.1556, ICLR 2015.

[8] Liu, Jingen, Jiebo Luo, and Mubarak Shah. ”Recognizing realistic actions from videos “in the wild”.” In 2009 IEEE Conference on Computer Vision and Pattern Recognition, pp1996-2003. IEEE, 2009.

[9] Anguita, Davide, Alessandro Ghio, Luca Oneto, Xavier Parra, and Jorge Luis Reyes-Ortiz. ”A public domain dataset for human activity recognition using smartphones.” In Esann, vol. 3, p. 3. 2013.

[10] Sharma, Neha & Jain, Vibhor & Mishra, Anju. (2018), ”An Analysis Of Convolutional Neural Networks For Image Classification”, Procedia Computer Science, no. 132, pp377-384, 2018

[11] J. Yang and J. Li, ”Application of deep convolutional neural network”, In: 2017 14th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP), Chengdu, (2017), pp229-232, 2017.

[12] J. Kennedy and R. Eberhart, ”Particle swarm optimization,”. In: Proceedings of ICNN’95 - International Conference on Neural Networks, Perth, WA, Australia, 1995, pp1942-1948 vol.4, 1995.

[13] B. Qolomany, M. Maabreh, A. Al-Fuqaha, A. Gupta and D. Benhaddou, ”Parameters optimization of deep learning models using Particle swarm optimization”, In: 2017 13th International Wireless Communications and Mobile Computing Conference (IWCMC), Valencia, pp1285-1290, 2017.

[14] B. Xue, M. Zhang and W. N. Browne, ”Particle Swarm Optimization for Feature Selection in Classification: A Multi-Objective Approach,”. IEEE Transactions on Cybernetics, vol. 43, no. 6, pp1656-1671, 2013.

[15] Beyer, Hans-Georg & Schwefel, Hans-Paul, ”Evolution strategies - A comprehensive introduction”, Natural Computing, vol. 1, no. 1, pp3-52, 2002.

[16] Miao, Jianyu & Niu, Lingfeng, ”Survey on Feature Selection,”. Procedia Computer Science, 91, pp919-926, 2016.

[17] James Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl, ”Algorithms for hyper-parameter optimization”, In Proceedings of the 24th International Conference on Neural Information Processing Systems (NIPS’11), pp2546–2554, 2011.

[18] S. Zha, F. Luisier, W. Andrews, N. Srivastava, and R. Salakhutdinov, “Exploiting Image-trained CNN Architectures for Unconstrained Video Classification,” 26th British Machine Vision Conference (BMVC’15). pp. 60.1-60.13, 2015.

[19] Wu, Zuxuan, Ting Yao, Yanwei Fu, and Yu-Gang Jiang. ”Deep learning for video classification and captioning,”. In Frontiers of multimedia research, pp3-29. 2017.

[20] Dan, Wu and Xie Wei-Hua, ”Short Video Classification Based on Spatio-Temporal Features and SVM,”. In: IEEE/ACIS 18th International Conference on Computer and Information Science (ICIS), pp493-496, 2019.

[21] Sun, Yanan & Xue, Bing & Zhang, Mengjie & Yen, Gary, ”Automatically Designing CNN Architectures Using Genetic Algorithm for Image Classification,”. IEEE Transactions on Cybernetics 50, no. 9 (2020) pp3840-3854, 2020.

[22] Q. Shen, C. Liu, H. Zou, S. Zhou and T. Chen. ”A Method of Image Classification with Optimized BP Neural Network by Genetic Algorithm,”. In: 2015 International Conference on Intelligent Networking and Collaborative Systems, Taipei, pp123-129, 2015.

[23] Vieira, Susana M., Luís F. Mendonça, Goncalo J. Farinha, and João MC Sousa. ”Modified binary PSO for feature selection using SVM applied to mortality prediction of septic patients,”. Applied Soft Computing 13, no. 8 (2013), pp3494-3504, 2013.

[24] Young, Steven & Rose, Derek & Karnowski, Thomas & Lim, Seung-Hwan & Patton, Robert, ”Optimizing deep learning hyper-parameters through an evolutionary algorithm,”. In MLHPC ’15, pp1-5, 2015.

[25] Wei-Zhong Sun, Fu-Jun Guo, Jie-Sheng Wang, Lin Chen, Dong Wei, Xin-Feng DuFeature. ”Extraction, Essential Dimension Estimation and Dimension Reduction Method of Colony Images,”. IAENG International Journal of Computer Science, Volume 48, Issue 2, pp379-391, 2021.

[26] Wei-Zhong Sun, Min Zhang, Jie-Sheng Wang, Sha-Sha Guo, Min Wang, and Wen-Kuo Hao, ”Binary Particle Swarm Optimization Algorithm Based on Z-shaped Probability Transfer Function to Solve 0-1 Knapsack Problem,” IAENG International Journal of Computer Science, vol. 48, no.2, pp294-303, 2021.

[27] Elgendy, Islam T., Moheb R. Girgis, and Adel A. Sewisy. ”A GA-Based Approach to Automatic Test Data Generation for ASP .NET Web Applications,”. IAENG International Journal of Computer Science 47, no. 3 (2020): pp557-564, 2020.

[28] Chethan Sharma, Siddharth Singh, Poornalatha G, and Ajitha Shenoy KB, ”Performance Analysis of Object Detection Algorithms on YouTube Video Object Dataset,” Engineering Letters, vol. 29, no.2, pp813-817, 2021

[29] Ahmad, Iftikhar, ”Feature Selection Using Particle Swarm Optimization in Intrusion Detection,”. International Journal of Distributed Sensor Networks 11 , no. 10, pp1-8, 2015.

[30] Aghdam, Mehdi Hosseinzadeh and Heidari, Setareh. ”Feature Selection Using Particle Swarm Optimization in Text Categorization” Journal of Artificial Intelligence and Soft Computing Research, vol.5, no.4, 2015, pp.231-238. https://doi.org/10.1515/jaiscr-2015-0031.

[31] Maheswaranathan, N., Luke Metz, George Tucker, Dami Choi and J. Sohl-Dickstein, ”Guided evolutionary strategies: augmenting random search with surrogate gradients,”. ICML, pp4264-4273, PMLR, 2019.

[32] Rajaraman S, Antani SK, Poostchi M, Silamut K, Hossain MA, Maude RJ, Jaeger S, Thoma GR, ” Pre-trained convolutional neural networks as feature extractors toward improved malaria parasite detection in thin blood smear images” PeerJ. 2018 Apr 16;6:e4568. doi: 10.7717/peerj.4568. PMID: 29682411; PMCID: PMC5907772.