SSRG International Journal of Electrical and Electronics Engineering
ISSN: 2348-8379/ https://doi.org/10.14445/23488379/IJEEE-V10I5P107

Volume 10 Issue 5, 69-78, May 2023

Original Article

Automatic Detection of Sign Language Fingerspelling on Combined Features and Feature Selection using Improvised Battle Royale Optimisation Algorithm

Rajesh George Rajan[1], P Selvi Rajendran[2], Jaison Mulerikkal[3], R. Surendiran[4]

[1, 2]*Department of CSE, Hindustan Institute of Technology and Science, Padur, Chennai, Tamil Nadu, India*
[3]*Department of CSE, Rajagiri School of Engineering & Technology, Kochi, Kerala, India*
[4]*School of Information Science, Annai College of Arts and Science, Kumbakonam, India*

[1]*Corresponding Author : rajeshgeorge1017@gmail.com*

Received: 06 March 2023 Revised: 14 April 2023 Accepted: 08 May 2023 Published: 29 May 2023

Abstract *- Sign language is a need for deaf pupils to communicate with one another. People who are not deaf often do not learn sign language to interact with deaf people. It is also necessary to have an interpreter to explain the sign's meaning to others who are unfamiliar with it. Several unresolved issues, such as uncontrolled signing situations, various types of light, and varying degrees of partial occlusion, have adversely impacted hand gesture recognition efficacy. The suggested technique is unusual because it employs integrated features created by combining features obtained using conventional handcrafted feature extraction methods with deep learning models. Understandably the combined characteristics will include some repetitive and unnecessary characteristics, increasing computation time and wasting resources. We prevent this by using feature selection (FS) before providing the classifier with the merged features. We present the improved version of the newly developed Battle Royale Optimisation, IBROA, for feature selection. The characteristics are fed into a classifier for classification. Experiments were carried out, and the findings show that the proposed IBROA, which utilises integrated features and feature selection, outperforms classifiers and shows novel and efficient techniques for feature selection in Sign language classification.*

Keywords *- Deep learning model, Royal battle optimisation, Sign language.*

1. Introduction

Fingerspelling and body motions are the two types of sign language. Fingerspelling uses just one or both hands to represent alphabets, numerals, or signs of a writing system. On the other hand, body gestures use the movement of the body and hands to communicate particular meanings. Sign language recognition is critical for human-computer interaction, virtual reality, and sports training, among other uses [1]. In today's fast-paced world, there is an urgent need to help the most marginalised communities. People with speech impairments communicate via sign language, making it difficult for them to interact with others. So, there is a communication gap that is impossible to close in mainstream culture, even though it exists. According to the Indian National Association of the Deaf, about a million individuals have some functional hearing loss.

A variety of hand movements represents sign language's alphabet. For the hand gesture, there are four parameters to consider: the shape or arrangement of the hand (finger placements), its orientation, where it is located, and how it moves. Depending on the study, hand gesture recognition relies on direct measurement or vision-based methods. To use the first method, individuals must wear special gloves or markers. To use the second method, users can use image processing techniques and keep their hands free at all times. As a result, it is more affordable and does not necessitate expensive equipment or calibration. Feature extraction (FE), a mix of handcrafted (HC) and deep learning (DL) features, is the next stage in automated detection systems. FE is an efficient method of decreasing the depth of input information so that it is represented in a more understandable and readily analysed manner [2-4]. These methods are often used to extract HC characteristics, which are manually chosen picture-specific traits that establish a focus on the target space's parameters. Due to the simplicity of extracting HC characteristics, especially from small data sets, researchers have used them extensively. With the aid of experts in the related subject, these qualities can be identified.

HC characteristics do not need any training to be derived and may be viewed more readily than other features. Simultaneously, we may utilise HC features to give weights to each feature depending on their importance in completing

the desired result. However, in the case of complex pictures, these characteristics become challenging to choose, and therefore another feature extraction method, called Deep Learning models (DLMs) as feature extractors, is used. Due to recent advances in computing power, such as introducing faster and more compact processors, DLMs have obtained tremendous popularity in the past years, allowing experts to train deeper networks with more accessibility and less time [5-7]. For feature extraction, we often use CNNs as the DLM. These models can automatically recognise features in an image, but they need a large training sample with many differences for good attribute derivation [8]. They can acquire a more significant amount of low-level characteristics, which may better characterise the picture and support subsequent analysis. Since DLM can do the same task automatically, it does not need significant preprocessing. The high data-intensive requirements of such models, however, remain a limitation. As shown, HC and DLM approaches' features have benefits and drawbacks. We suggest a model which uses combined features resulting from the integration of HC and DL characteristics for categorisation to incorporate both techniques' advantages.

Feature extraction from an image is a crucial phase in the image processing technique since the efficiency of a classifier mostly depends on the characteristic features. Nevertheless, not all retrieved characteristics from the above methods (HC and DLM) are equally significant in helping to distinguish the classifications. The inclusion of unnecessary and duplicate features may negatively impact classifier performance. Since feature selection is necessary, characteristics that may inversely influence the decision-making abilities of the model are included. To reduce the loss on the training dataset, selecting the optimum selection of features may be turned into an optimisation issue. The formalisation is as follows: Finding the best combination of features should improve testing accuracy while decreasing the amount of data lost from the training set. Testing every possible feature combination takes much time, which is one of the most significant drawbacks of feature selection methodologies [9].

The formula for the number of potential solutions with n characteristics is: $(2^n - 1)$. As a result, the number of solutions grows exponentially as the power of 2 increases. So, to get an optimum solution, brute-force feature selection techniques will need an enormous amount of time and computing resources. Metaheuristic algorithms can explore the solution space more quickly than brute force techniques and offer a nearly optimal solution in less time. Metaheuristic algorithms have recently gained popularity among academics, who have shown that they may provide optimum solutions to real-world issues that are harder to solve or take a significant amount of processing time when traditional methods are used. Because of the algorithm's unpredictability, they may develop locally optimum solutions. This gives the motivation for feature selection using a metaheuristic method. In this case, the feature was selected by a newly developed metaheuristic method, Battle Royale Optimization Algorithm (BRO). Here we develop an Improvised Battle Royale Optimization Algorithm (IBROA) for feature selection.

Optimisation is selecting the most practical solution to a problem from a pool of viable options. Optimisation algorithms are essential in various real-world situations, including academic and professional contexts. In the last twenty-five years, metaheuristic algorithms have been increasingly extensively accustomed to address optimisation issues [10]. The stochastic metaheuristics optimisation techniques include swarm intelligence (SI), physical phenomena algorithms and evolutionary algorithms (EA) [11-18].

Exploration and exploitation are two of the principles behind all optimisation algorithms. In order to provide effective results, exploration and exploitation must hang in the balance using optimisation algorithms. The practice of seeking new potential solutions throughout a broad range of the available search area is known as exploration. In other words, unexplored regions are investigated to see whether they might be used as escape routes from bad local optima. Alternatively, exploitation is searching for a new solution close to the best [19]. For instance, selection mechanisms facilitate exploitation, and crossover and mutation processes facilitate exploration. With considerable velocity changes, the particles devote more effort to exploration when the velocity is altered at the start of iterations. The particles will make more effort to get the resources as the velocities move closer to zero. The coefficient parameters should be set appropriately for the ideal equilibrium between exploration and exploitation.

Elitism is a critical feature of metamorphic algorithms; it preserves the optimal solutions obtained during optimisation. During the last decade, new methods have emerged that have been applied to various complex real-world optimisation issues. The balance of exploration and exploitation in metaheuristic methods to solve optimisation issues is critical to their success [20]. Conversely, each metaheuristic algorithm has its mechanism for using the exploration and exploitation features via various techniques and procedures. The NFL theorem [21] shows that we cannot perform well across all optimisation problems using one optimisation method. According to this idea, even though a specific optimisation method outperforms others in solving many optimisation issues, NFL may do poorly in a specific set of issues. Currently, no heuristic method handles all optimisation issues with better performance. This concept shows novel approaches to tackling problems and unsolved situations. Consequently, new ones must be suggested in addition to enhancing existing techniques.

1.1. Battle Royale Game Optimizer

Battle Royale Optimization (BRO) [22] is a population-based approach comparable to other well-known swarm-based systems; it leverages a solution space of a candidate to get the ideal response. Players are assigned to each solution and are responsible for defeating their immediate vicinity (neighbours). Thus, a player in a superior position may beat his competitors and gain victory. To increase their chances of survival, all people strive to advance or move to a more favourable place within the dimension. Afterwards, the best team will be declared the winner after the iteration.

The Battle Royale video games are based on the Japanese Film Battle Royale, and they include intense last-man-standing combat in which players compete against one another. The game area is often reduced, requiring players to explore and use their surroundings to live. Many players may participate in battle royale games on a digital platform, which is one of the most attractive features of these games. Individuals versus individuals and groups of up to five players are permitted to participate in these games. The most challenging part when playing these games is defeating the other players while beating the game since the play area (also known as the "safe region") decreases as the game continues. Any players discovered outside of the safe zone will be subjected to damage or booted out of the game. These players are now confined to a small geographic region and must struggle with various ailments. Players in most battle royale games start with roughly the same amount of power and resources; they are typically positioned randomly inside the game area and are not given any resources or equipment, to begin with. Instead, exploring is a crucial part of the game since players are tasked with looking for items to assist them in their fight for life (all while avoiding being killed by opponents). The ability to respawn after being eliminated from a battle royale game is another facet of these games that is accounted for by the algorithm. Some battle royale games enable players to respawn themselves, but this is not a feature found in every game in the genre.

BRO begins with a randomly generated population evenly dispersed across the problem space, as shown in the diagram. Each person (soldier/player) then attempts to inflict harm on the closest warrior by firing a firearm. Warriors in preferred positions, as a result, inflict harm on their immediate surrounding communities. When another injures one soldier, the damage level of the injured soldier rises by 1. These communications are computed analytically by each x_i. *injury*=x_i. *injury+1*, where x_i. injury is the i^{th} soldier's injury level among the population.

Furthermore, soldiers want to shift their positions as soon as they are damaged, and, as a result, they assault their opponents from another direction. The wounded soldier moves toward a spot that is midway between their previous position and the best position that has been located so far so that he may focus on exploitation (elite player). These communications are computed theoretically by:

$$X_{inj,d} = X_{inj,d} + r(X_{best,d} - X_{inj,d}) \quad (1)$$

Whereas d is the location of the injured warrior in the dimension d, r is a random number equally distributed between [0, 1] and x *in*. Furthermore, xi. the injury will be adjusted to 0 if injured soldiers can deal damage to their adversary in the next round. Additionally, to keep the focus on exploration, when a warrior's injury level reaches a certain threshold, they die and reappear arbitrarily in the viable dilemma space and their xi. injury is altered to zero. Following their death, the soldier has the following traits when they return to the issue area:

$$X_{dam,d} = r(ub_d - lb_d) + lb_d \quad (2)$$

This interaction is beneficial for both the exploration and exploitation stages of the problem-solving process when the lower and upper boundaries of dimension d in the issue space are denoted by lb_d and ub_d. As a necessary result, the lower and upper boundaries are going to be modified in the following way:

$$lb_d = x_{best,d} - SD(\bar{x}_d) \quad (3)$$

$$ub_d = x_{best,d} + SD(\bar{x}_d) \quad (4)$$

The standard deviation is represented as SD, and x_{best} is the position of the best solution.

2. Improvised Battle Royale Optimisation Algorithm

Input: The dataset contains the feature set of each picture acquired during the feature extraction step.

Output: The optimal subset of features that may be utilised for picture categorisation.

Here for the discrete optimisation, we used the fitness of the solider and the sigmoid function is represented below in equations 5 and 6.

$$\text{fitness}_{\text{sold}} = \text{feature}_{imp} + weight\ factor * \left(1 - \frac{fs}{fp}\right) \quad (5)$$

$$\text{sig}(X) = \frac{1}{1+e^{10*(X-0.5)}} \quad (6)$$

Pseudocode for BRO [22]
Begin
Randomly initialise a population $\bar{x}$
Initialise all parameters;
shrink= ceil ($\log_{10}$(MaxXicle)))
Δ = round (MaxXicle / shrink)

Iter = 0;
while the termination criterion is not met, do
iter = iter + 1
for i =1: population_size
//compare i^{th} soldier with nearest one (j^{th})
dam = i
vic = j
if $f(x_i) < f(x_j)$
dam = j
vic = i
end if
if x_{dam}.damage < Threshold
for d = 1: Dimension
Change the position of damaged soldier based on:
$x_{dam,d} = r\ (\max(x_{dam,d}, x_{best,d}) - \min(x_{dam,d}, x_{best,d})) + \max(x_{dam,d}, x_{best,d})$
end for d
x_{dam}. damage = x_i. damage + 1
x_{vic}. damage = 0
else
for d = 1: Dimension
$x_{dam,d} = r\ (ub_d - lb_d) + lb_d$
end for d
update f (x_{dam})
x_{dam}. damage = 0
end for i
if iter >= Δ
update (ub - lb) based on equation (3)
Δ = Δ + round(Δ/2);
end if
if the lb_d or ub_d exceeds the original lower/upper bound, then it set to the original lb_d or ub_d
end while

Pick the best warrior to be the solution.

2.1. Handcrafted Features

2.1.1. GLCM

The GLCM functions determined the recurrence of occurrence sets of pixels with specific measures in a specified spatial relationship, creating a GLCM and deriving quantitative measurements from this matrix to characterise texture in an image. Two stages are required to extract co-occurrence texture characteristics from a picture.

Initially, a GLCM is used to tabulate the pairwise spatial co-occurrences of pixels separated by a certain distance and angle. Then, a collection of scalar values describing various underlying texture features are computed using the GLCM. In the GLCM, the number of times various combinations of gray levels appear together in a picture is tabulated. The dimension of the square matrix that makes up the GLCM depends on N, the number of discrete grey levels in a picture.

2.1.2. LBP

The most basic LBP operator was developed to describe texture characteristics [23]. Each pixel is described using the method of comparing values around it. If the values of the pixels surrounding it are the same, then the pixel value is fixed to one; otherwise, it is fixed to 0. The result is a string of binary patterns from the surrounding area, which is then turned into a decimal number that describes each pixel uniquely. The new system uses the concept of uniform patterns, modifying the original local binary pattern (LBP).

The dimension of the FV is significantly brought down when there are predictable or standardised patterns. This add-on was created based on research discovering that specific binary patterns appear in texture pictures more often than others. If there are just two bitwise varies from 0 to 1 or vice versa in any direction, a binary pattern is said to be uniform. When we use uniform patterns in the feature representation, the length of the feature vector becomes reduced. If we utilise 8 pixels from the same neighbourhood, there are 256 patterns, 58 of which are uniform, and hence 59 distinct labels. In the current world, the LBP operator's most significant feature is its ability to handle changes in monotonic grayscale values that are induced, for example, by changing lighting conditions. Computational simplicity is also essential since pictures can be quickly and easily analysed in demanding real-time scenarios.

2.2. Deep Learning Features

2.2.1. Xception

The idea behind the Inception model [24] was the motivation behind developing the Xception architecture [25]. According to this approach, it is possible to properly separate the cross-channel and spatial correlation of N-dimensional data so they do not map together. By entirely decoupling cross-channel and spatial correlations, Xception can improve its performance significantly. A separate convolution layer is used in order to achieve this goal successfully.

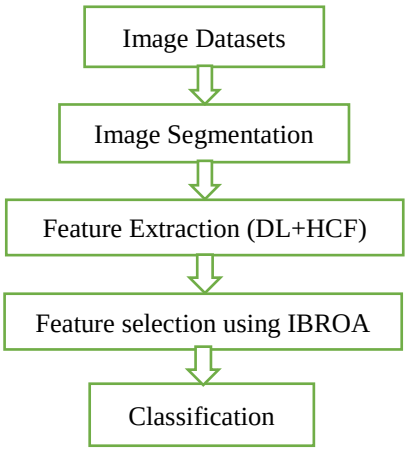

Fig. 1 Proposed methodology

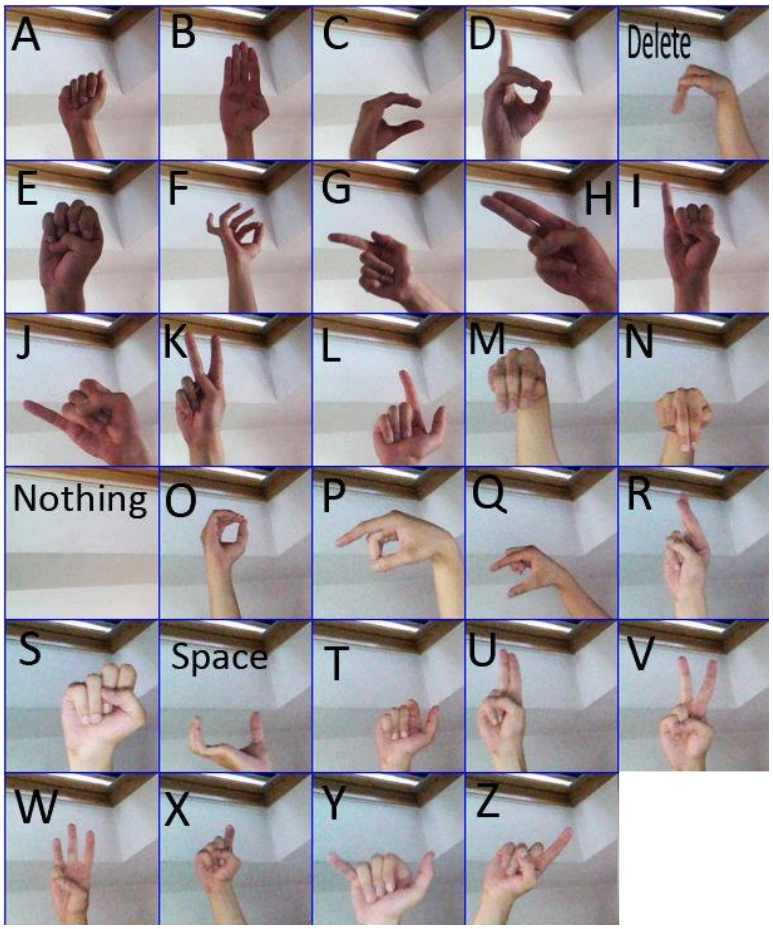

Fig. 2 American sign language (ASL alphabet)

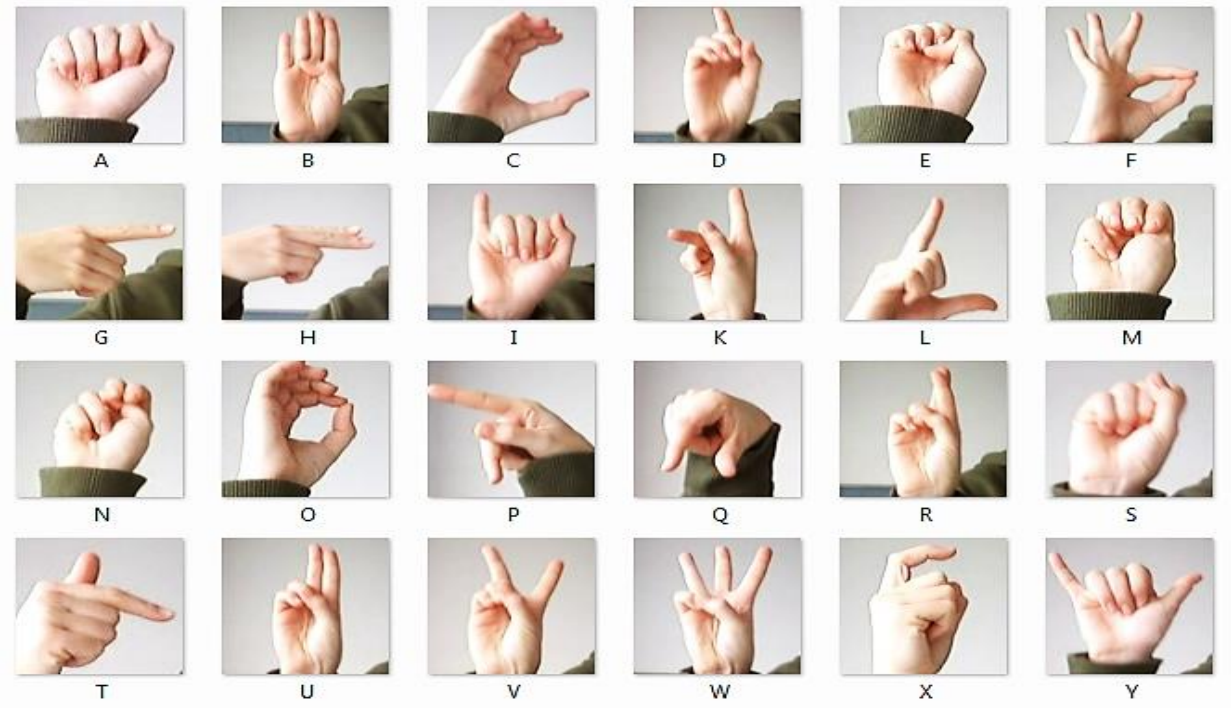

Fig. 3 American sign language (ASL MNIST alphabets)

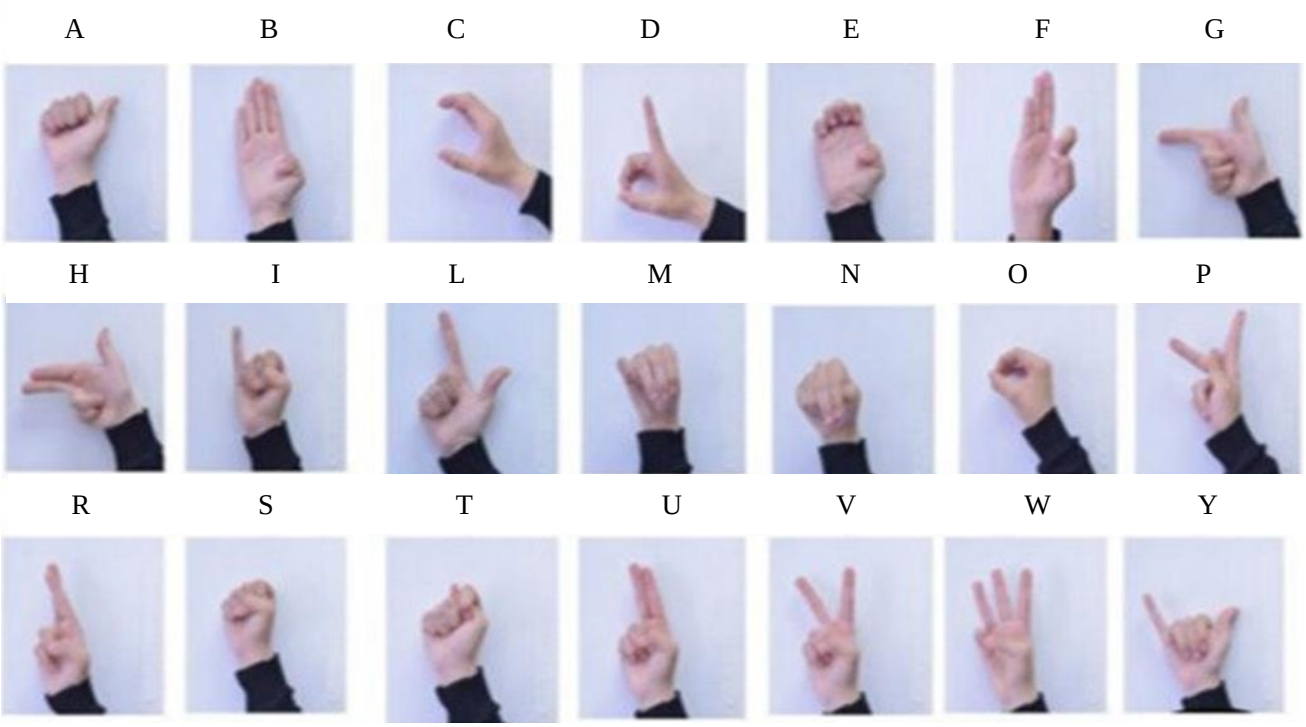

Fig. 4 Mexican sign language (MSL alphabet)

Additionally, it utilises shortcut connections between convolution layers to do identity mapping, as suggested by ResNet [26]. Each block of the segmented model comprises at least two independently moveable convolution layers with a 3x3 kernel size. The image size is reduced by combining the MaxPooling of kernel 3x3 in the early blocks with the convolution of kernel 3x3 with stride 2 in shortcut connections. This reduces overall size of the picture. The resulting model was particularly good at learning from complex datasets, which was one of its primary strengths.

2.3. Datasets Used

2.3.1. American Sign Language (ASL Alphabet)

There are 87,000 colour images with a dimension of 200 by 200 pixels that make up the American Sign Language alphabet dataset. It contains 29 classes, is labelled from zero to twenty-eight, and has a one-to-one relationship with every symbol of the American alphabet, from A - Z (0 - 25 for the alphabet, meaning 0 = A, 25 = Z), as well as a one-to-one relationship with the number zero (0). The last three classes (numbered twenty-six to twenty-eight, i.e., 26 = space, 27 = delete, and 28 = null) relate to space, delete, and null symbols, respectively. The ASL alphabet database sample of the pictures is shown in Figure 2.

2.3.2. American Sign Language (ASL MNIST)

This is a dataset group that consists of 34,627 pictures that are grayscale and have a resolution of 28 by 28 pixels. It contains twenty-four named classes with numbers ranging from zero to twenty-five, all associated with an alphabet letter from A to Z (nine = J and twenty-five = Z were omitted because of the movement of gestures). The above Figure 3 shows an example of a sign image.

2.3.3. Mexican Sign Language (MSL Alphabet)

This was constructed by 18 individuals, including sign language interpreters and deaf children. Students from a Mexican high school formed an inclusive organisation. As seen in Figure 4, this database has 21 classes that use the MSL alphabet without movement. Each letter was photographed ten times, yielding 3780 grayscale pictures of 32 by 32 pixels.

3. Result Analysis

This division talks about the effectiveness of the proposed IBRO approach. Five benchmark datasets were taken from the University of California, Irvine (UCI) machine learning repository [27] sign language datasets are utilised to test the recommended approaches. Table 1 depicts the specifics of the datasets, like how many people, traits, and classes they have. The characteristics were first normalised for every dataset to a value between 0 and 1.

When selecting wrapper features, the k-nearest neighbour (KNN) classifier can be used to determine the classification error rate (CER) in the fitness function. Euclidean distance is used as the distance metric, and k is set to 5. Previously published work [28] has shown that the KNN has promising performance and a fast calculation speed.

Table 1. The list of five benchmarked databases

No.	Database Name	No. of Instances	No. of Features	No. of Classes
1	Breast Cancer Wisconsin	699	9	2
2	Ionosphere	351	34	2
3	Musk 1	476	167	2
4	Breast Cancer Coimbra	116	9	2
5	Seeds	210	7	3

Table 2. Results of five feature selection approaches on UCI depository

Database	FSM	BF	WF	MF	STD	Accuracy	Feature Size
1	BPSO	0.0153	0.0234	0.0155	0.0009	98.96	4.72
	GA	0.0146	0.0179	0.0154	0.0005	99.02	4.62
	BGSA	0.0115	0.0180	0.0138	0.0027	99.19	4.25
	Proposed IBROA	0.0152	0.0422	0.0139	0.0006	99.20	4.65
2	BPSO	0.1246	0.1382	0.1329	0.0036	88.01	14.30
	GA	0.1165	0.1263	0.1290	0.0107	89.48	14.05
	BGSA	0.1026	0.1371	0.1230	0.0112	90.12	12.50
	Proposed IBROA	0.0622	0.1285	0.0922	0.0110	91.58	12.55
3	BPSO	0.0852	0.1225	0.0912	0.0095	91.88	76.05
	GA	0.0935	0.1122	0.0944	0.0033	91.02	81.15
	BGSA	0.0809	0.1170	0.1006	0.0116	92.32	80.10
	Proposed IBROA	0.0738	0.1415	0.0684	0.0079	92.16	82.21
4	BPSO	0.1403	0.1493	0.1535	0.0033	86.09	4.09
	GA	0.1276	0.1449	0.1281	0.0019	87.56	4.66
	BGSA	0.0989	0.1526	0.1295	0.0224	90.39	4.32
	Proposed IBROA	0.0982	0.1552	0.1102	0.0118	90.87	4.03
5	BPSO	0.0516	0.0521	0.0517	0.0002	95.32	3.15
	GA	0.0515	0.0515	0.0520	0.0000	95.23	2.91
	BGSA	0.0503	0.0511	0.0504	0.0005	95.22	2.04
	Proposed IBROA	0.0505	0.0510	0.0506	0.0006	95.21	2.53

Table 3. Results of five feature selection approaches on ASL datasets

Database	FSM	BF	WF	MF	STD	Accuracy	Feature Size
ASL Alphabet (8)	BPSO	0.1566	0.1968	0.1995	0.0251	81.58	41.16
	GA	0.1756	0.3525	0.1778	0.0121	82.66	42.44
	BGSA	0.1375	0.3365	0.2534	0.0887	86.58	40.65
	Proposed IBROA	0.1001	0.2420	0.1244	0.0256	88.27	37.65
ASL MNIST Alphabet (5)	BPSO	0.2113	0.2751	0.2248	0.0129	76.64	43.62
	GA	0.2247	0.2765	0.2847	0.0030	78.78	40.84
	BGSA	0.2426	0.2645	0.2172	0.0128	78.05	45.40
	Proposed IBROA	0.1825	0.2652	0.1946	0.0261	80.08	40.01
MSL Alphabet (3)	BPSO	0.2521	0.2884	0.2475	0.0026	93.85	42.20
	GA	0.2350	0.3511	0.2351	0.0027	96.63	43.64
	BGSA	0.2291	0.1834	0.0412	0.0083	97.56	41.63
	Proposed IBROA	0.1197	0.2961	0.0281	0.0129	97.97	41.05

Table 4. T-test with p-values

Dataset	BPSO	GA	BGSA	IBROA
1	0.36329	0.21567	0.22821	0.20394
2	1.10×10^{-5} *	0.00000 *	0.00251 *	0.69354
3	0.00592 *	1.00×10^{-5} *	0.02259 *	0.00182 **
4	0.00013 *	0.00001 *	0.38690	1.00×10^{-5} *
5	1.00100	1.00060	1.00040	1.00030
ASL	0.00015 *	0.00000 *	0.00001 *	0.00189 *
ASL-MNIST	0.00013 *	0.00000 *	0.00001 *	0.00129 *
ASL-MSL	0.00014 *	0.00040 *	0.00011 *	0.00078 *

Table 5. Average computational cost

Dataset	BPSO	GA	BGSA	IBROA
1	5.600	8.872	5.602	6.862
2	2.465	3.803	2.429	3.888
3	4.043	6.108	4.395	4.922
4	1.132	1.849	1.428	2.256
5	1.519	2.477	1.621	2.947
ASL	6.340	4.820	4.288	4.967
ASL-MNIST	2.556	2.370	2.332	2.688
ASL-MSL	2.668	2.012	2.358	2.803

Specifically, the hold-out approach divides each dataset into two portions: 80% is used for training, and 20% is used for testing. Wrapper features are chosen using the fitness function, which considers classification performance and feature size. The fitness function could be written as follows.

$$Fitness = \propto \frac{|L|}{|F|} + (1-\propto)Error \quad (7)$$

In this equation, "Error" represents the classification error rate, |L| indicates the length of the feature subset, and |F| represents the complete count of characteristics in each dataset. Finally, "0, 1" denotes the parameter that regulates the impact of classification performance and feature size. Following [29, 30], we decided to make the value 0.01 throughout this research. The error is calculated by combining the KNN algorithm with Euclidean distance and setting k equal to 5. This is done in combination with the Euclidean distance [30-32].

The KNN was chosen as the algorithm because it has a cheap computational cost and is easy to implement [30-32]. The last stage of the feature selection process culminates in producing the best solution, i.e., the feature subset comprising the most valuable features. The performance of four cutting-edge feature selection methods (FSM) was compared, including binary particle swarm optimisation (BPSO) [33], genetic algorithm (GA) [28], and binary gravitational search algorithm (BGSA) [34]. The best fitness (BA), worst fitness (WF), mean fitness(MF), the standard deviation of fitness (STD), feature size (number of chosen characteristics), and accuracy are among the six assessment measures collected in the experiment.

$$\text{Best Fitness} = \min_{i=1}^{I\max} FV_i \quad (8)$$

$$\text{Worst Fitness} = \max_{i=1}^{I\max} FV_i \quad (9)$$

$$\text{Mean Fitness} = \frac{1}{I_{\max}} \sum_{i=1}^{I\max} FV_i \quad (10)$$

$$\text{STD} = \sqrt{\frac{\sum_{i=1}^{I\max} (FV_i - \mu)^2}{I_{\max}}} \quad (11)$$

$$\text{Accuracy} = \frac{\text{No.of precisely predicted occurrences}}{\text{Total no. of occurrences}} \times 100\% \quad (12)$$

Table 3 outlines the five distinct feature selection approaches for ten different datasets regarding the abovementioned measures. Note that the bold parameter value yields the best results for each approach. In Table 2 and Table 3, the performance is considered superior if the values for best, worst, and mean fitness are lower.STD is a representation of the algorithm's robustness and consistency. Therefore, a feature selection technique is highly robust if it achieves the lowest STD value possible. This approach has the potential to yield very consistent findings.

Additionally, IBROA has the potential to produce a higher fitness value than BPSO in most cases. This suggests that the suggested method is superior to BPSO since it overcomes the constraints of BPSO in premature convergence. One way to look at it is that a higher accuracy value implies that more samples have been correctly predicted. The fact that other samples have been scheduled supports this. For instance, fewer features have been chosen if the size includes fewer features. Then, IBROA performed very well regarding the worst, best and mean fitness values. Experiments show that IBROA did an excellent job of choosing the essential characteristics of the problem, which led to a promising performance.

Using a two-sample t-test with a confidentially level of 95%, a comparison was made between the classification performance of the recommended IBROA approach and that of the other methodologies. This was done as a precautionary measure because the categorisation outcome for each individual is the average accuracy from twenty separate runs.

In the context of a statistical test, the term "null hypothesis" refers to the assumption that the accuracy of categorisation provided by two distinct methods is equivalent. If the p-value is less than 0.05, it is assumed that the null hypothesis, which asserts that there is no discernible difference in classification performance between the two alternative approaches, is wrong. This is so because the null hypothesis suggests that the two groups vary noticeably from one another. The t-test outcomes and accompanying p-values are shown in Table 4, which uses the IBROA as its point of reference. Also included are the corresponding t-test results.

The proposed approach's performance was considerably better in this table with a notation of "*," but the IBROA's performance was noticeably poorer with a notation of "**." On a total of five datasets from the University of California, Irvine (UCI), and three datasets from the field of sign language, it was shown that IBROA was much better at classifying data than BPSO and GA (p-values less than 0.05).

4. Conclusion

This study presents an automated detection approach for sign language alphabets, Integrated Features-Feature Selection Model for Classification. Here, KNN classification is performed using a combination of features acquired using handcrafted approaches and deep learning models. A feature selection phase is performed to eliminate redundant and irrelevant characteristics. The newly proposed IBROA is used for feature selection. The findings suggest that combining features and feature selection using IBROA is advantageous. The proposed approach yielded promising results, demonstrating that the experimental data outperforms current methods. Extensive trials on image data sets demonstrate that our method for selecting the appropriate feature subset achieves greater accuracy.

References

[1] M. Suriya et al., "Survey on Real Time Sign Language Recognition System: An LDA Approach," *International Journal of P2P Network Trends and Technology, vol.* 7, pp. 8-13, 2017.[Google Scholar][Publisher Link]

[2] Hanan Samet, "*Foundations of Multidimensional and Metric Data Structures*," Morgan Kaufmann, 2006.[Google Scholar][Publisher Link]

[3] C. Ding et al., "Adaptive Dimension Reduction for Clustering High Dimensional Data," *Proceedings of International Conference on Data Mining*, Maebashi City, Japan, pp. 147-154, 2002.[CrossRef][Google Scholar][Publisher Link]

[4] S. Senthamizhselvi, and A. Saravanan, "Intelligent Visual Place Recognition using Sparrow Search Algorithm with Deep Transfer Learning Model," *International Journal of Engineering Trends and Technology*, vol. 71, no. 4, pp. 109-118, 2023. [CrossRef][Publisher Link]

[5] Girika Jyoshna, and Md Zia Ur Rahman, "An Adaptive Learning Based Speech Enhancement Technique for Communication Systems," *International Journal of Engineering Trends and Technology*, vol. 69, no. 6, pp. 31-37, 2021. [CrossRef] [Publisher Link]

[6] Prafulla Mohapatra et al., "Sentiment Classification of Movie Review and Twitter Data Using Machine Learning," *International Journal of Computer and Organization Trends, vol.* 9, no. 3, pp. 1-8, 2019. [CrossRef][Publisher Link]

[7] V. V. Narendra Kumar, and T. Satish Kumar, "Smarter Artificial Intelligence with Deep Learning," *SSRG International Journal of Computer Science and Engineering* , vol. 5, no. 6, pp. 10-16, 2018.[CrossRef][Google Scholar][Publisher Link]

[8] Lakshman Karthik Ramkumar, Sudharsana Premchand, and Gokul Karthi Vijayakumar, "Sign Language Recognition using Depth Data and CNN," *SSRG International Journal of Computer Science and Engineering* , vol. 6, no. 1, pp. 9-14, 2019. [CrossRef][Google Scholar][Publisher Link]

[9] Mrinalini Rana, and Omdev Dahiya, "A Hybrid Grouped-Artificial Bee Colony Optimization (G-ABC) Technique for Feature Selection and Mean-Variance Optimization for Rule Mining," *International Journal of Engineering Trends and Technology*, vol. 71, no. 4, pp. 12-20, 2023. [CrossRef][Publisher Link]

[10] A. Lazar, "Heuristic Knowledge Discovery for Archaeological Data using Genetic Algorithms and Rough Sets," *Heuristic and Optimization for Knowledge Discovery*, pp. 263–278, 2002. [CrossRef][Google Scholar][Publisher Link]

[11] K. M. Passino, "Biomimicry of Bacterial Foraging for Distributed Optimization and Control," *IEEE Control Systems Magazine*, vol. 22, no. 3, pp. 52-67, 2002. [CrossRef][Google Scholar][Publisher Link]

[12] Xin-She Yang, "Firefly Algorithms for Multimodal Optimization," *Stochastic Algorithms: Foundations and Applications*, vol. 5792, pp. 169–178, 2009. [CrossRef][Google Scholar][Publisher Link]

[13] Seyedali Mirjalili, Seyed Mohammad Mirjalili, and Andrew Lewis, "Grey Wolf Optimizer," *Advances in Engineering Software*, vol. 69, pp. 46-61, 2014. [CrossRef][Google Scholar][Publisher Link]

[14] Mu Dong Li et al., "A Novel Nature-Inspired Algorithm for Optimization: Virus Colony Search," *Advances in Engineering Software*, vol. 92, pp. 65–88, 2016. [CrossRef][Google Scholar][Publisher Link]

[15] Yousef Sharafi, Mojtaba Ahmadieh hanesar, and Mohammad Teshnehlab, "COOA: Competitive Optimization Algorithm," *Swarm and Evolutionary Computation*, vol. 30, pp. 39-63, 2016. [CrossRef][Google Scholar][Publisher Link]

[16] Poonam Savsani, and Vimal Savsani, "Passing Vehicle Search (PVS): A Novel Metaheuristic Algorithm," *Applied Mathematical Modelling*, vol. 40, no. 5–6, pp. 3951-3978, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[17] Najmeh Sadat Jaddi, Jafar Alvankarian, and Salwani Abdullah, "Kidney-Inspired Algorithm for Optimization Problems," *Communications in Nonlinear Science and Numerical Simulation*, vol. 42, pp. 358-369, 2017. [CrossRef][Google Scholar][Publisher Link]

[18] Amir Seyyedabbasi, and Farzad Kiani, "I-GWO and Ex-GWO: Improved Algorithms of the Grey Wolf Optimizer to Solve Global Optimization Problems," *Engineering with Computers*, vol. 37, pp. 509–532, 2021. [CrossRef][Google Scholar][Publisher Link]

[19] Alireza Askarzadeh, "Bird Mating Optimizer: An Optimization Algorithm Inspired by Bird Mating Strategies," *Communications in Nonlinear Science and Numerical Simulation*, vol. 19, no. 4, pp. 1213-1228, 2014. [CrossRef][Google Scholar][Publisher Link]

[20] Matej Črepinšek, Shih-Hsi Liu, and Marjan Mernik, "Exploration and Exploitation in Evolutionary Algorithms: A Survey," *ACM Computing Surveys*, vol. 45, no. 3, pp. 1–33, 2013. [CrossRef][Google Scholar][Publisher Link]

[21] D. H. Wolpert, and W. G. Macready, "No Free Lunch Theorems for Optimization," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 67-82, 1997. [CrossRef][Google Scholar][Publisher Link]

[22] Taymaz Rahkar Farshi, "Battle Royale Optimization Algorithm," *Neural Computing and Applications*, vol. 33, pp. 1139-1157, 2021. [CrossRef][Google Scholar][Publisher Link]

[23] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution Gray-Scale and Rotation Invariant Texture Classification with Local Binary Patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 971-987, 2002. [CrossRef][Google Scholar][Publisher Link]

[24] Christian Szegedy et al., "Going Deeper with Convolutions," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1-9, 2015 [Google Scholar][Publisher Link]

[25] Francois Chollet, "Xception: Deep Learning with Depth wise Separable Convolutions," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1251-1258, 2017. [Google Scholar][Publisher Link]

[26] Kaiming He et al., "Deep Residual Learning for Image Recognition," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770-778, 2016. [Google Scholar][Publisher Link]

[27] UCI Machine Learning Repository. [Online]. Available: https://archive.ics.uci.edu/ml/index.php

[28] Cheng-Lung Huang, and Chieh-Jen Wang, "A GA-Based Feature Selection and Parameters Optimization for Support Vector Machines," *Expert Systems with Applications*, vol. 31, no. 2, pp. 231-240, 2006. [CrossRef][Google Scholar][Publisher Link]

[29] Nailah Al-Madi, Hossam Faris, and Seyedali Mirjalili, "Binary Multi-Verse Optimization Algorithm for Global Optimization and Discrete Problems," *International Journal of Machine Learning and Cybernetics*, vol. 10, pp. 3445-3465, 2019. [CrossRef][Google Scholar][Publisher Link]

[30] E. Emary, Hossam M. Zawbaa, and Aboul Ella Hassanie, "Binary Ant Lion Approaches for Feature Selection," *Neurocomputing*, vol. 213, pp. 54–65, 2016. [CrossRef][Google Scholar][Publisher Link]

[31] Bing Xue, Mengjie Zhang, and Will N. Browne, "Particle Swarm Optimisation for Feature Selection in Classification: Novel Initialisation and Updating Mechanisms," *Applied Soft Computing*, vol. 18, pp. 261-276, 2014. [CrossRef][Google Scholar][Publisher Link]

[32] E. Emary, Hossam M. Zawbaa, and Aboul Ella Hassanien, "Binary Grey Wolf Optimization Approaches for Feature Selection," *Neurocomputing*, vol. 172, pp. 371-381, 2016. [CrossRef][Google Scholar][Publisher Link]

[33] Li-Yeh Chuang et al., "Improved Binary PSO for Feature Selection using Gene Expression Data," *Computational Biology and Chemistry*, vol. 32, no. 1, pp. 29-38, 2008. [CrossRef][Google Scholar][Publisher Link]

[34] Esmat Rashedi, Hossein Nezamabadi Pour, and Saeid Saryazdi, "BGSA: Binary Gravitational Search Algorithm," *Natural Computing*, vol. 9, pp. 727-745, 2010. [CrossRef][Google Scholar][Publisher Link]