

Recognizing Ancient South Indian Language Using Opposition Based Grey Wolf Optimization

A. Naresh Kumar^{1,*} and G. Geetha²

¹Department of CSE, Anna University, Chennai, 600025, Tamilnadu, India

²Department of CSE, Agni College of Technology, Chennai, 600130, Tamilnadu, India

*Corresponding Author: A. Naresh Kumar. Email: nareshkumar123phd@gmail.com

Received: 08 February 2022; Accepted: 15 March 2022

Abstract: Recognizing signs and fonts of prehistoric language is a fairly difficult job that requires special tools. This stipulation make the dispensation period over-riding, difficult and tiresome to calculate. This paper present a technique for recognizing ancient south Indian languages by applying Artificial Neural Network (ANN) associated with Opposition based Grey Wolf Optimization Algorithm (OGWA). It identifies the prehistoric language, signs and fonts. It is an apparent from the ANN system that arbitrarily produced weights or neurons linking various layers play a significant role in its performance. For adaptively determining these weights, this paper applies various optimization algorithms such as Opposition based Grey Wolf Optimization, Particle Swarm Optimization and Grey Wolf Optimization to the ANN system. Performance results are illustrated that the proposed ANN-OGWO technique achieves superior accuracy over the other techniques. In test case 1, the accuracy value of OGWO is 94.89% and in test case 2, the accuracy value of OGWO is 92.34%, on average, the accuracy of OGWO achieves 5.8% greater accuracy than ANN-GWO, 10.1% greater accuracy than ANN-PSO and 22.1% greater accuracy over conventional ANN technique.

Keywords: Ancient language; symbols; characters; artificial neural network; opposition based grey wolf optimization

1 Introduction

Several of the rock engravings that are found at various areas of the sphere expose the particulars of indulgence, regime, financial circumstance, principles, and managerial protocols trailed by numerous leaders and empires precise to those areas [1]. Recently, there has been a fast developing concern in the study on more varied languages and writings [2]. Tamil is one of the noteworthy languages in Dravidian Language Family. It has an opulent script basis start from 300BC [3]. Hence, it is essential to develop a new scheme which can identify the prehistoric Tamil fonts accurately and can alter them into contemporary Tamil fonts [4]. The prehistoric Tamil fonts i.e., the initial and primitive Tamil can be normally found in rock engravings and palm leaves. Only epigraphists can examine those rock engravings. To spread the readability and to defend the old realistic abilities, we require a decent appreciation scheme that can translate the prehistoric writing to contemporary writing [5].



This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

There are several schemes available for printed script identification. When the plan formation is over, the test image is input into the scheme. The system splits the image into separate letters and each slice of the image is related with the preloaded patterns. A related technique is revised for every fragmented letter in the image [6]. Handwritten letter identification is a computer facilitated process that enables inferring and reading readable handwritten inputs provided on ancient scripts like palm leaves, rock engravings, etc. [7]. The development of imaging technology has swayed a few regions in the computer apparition community [8]. The process of prehistoric Tamil letter identification comprises removal of some categorized qualities named sorts to categorize an unidentified letter into one of the well-known classes. Pre-processing is fundamentally made use of to reduce ranges of handwritten letters. A feature extractor is rudimentary for creative data depiction and removing significant features for future dispensation. A classifier allocates the letters to one of the few classes [9].

A prehistoric Tamil letter identification procedure centred on fake resistant was suggested to advance the rate of letter identification [10]. ANN is an example of such an agenda that alters the masses of its networks from the training forms. These masses verifiably work as structures for cataloguing [11]. Sánchez et al. (Sánchez, Melin, & Castillo, 2017) have suggested GWO for training segmental neural network with a granulated method for human acknowledgment [12]. Despite the fact that the investigator changes the prehistoric letters up to 90% precision, the translation verses cannot be agreed with the appropriate connotation in the sentence [13].

1.1 Problem Definition

Traditionally, recognizing the ancient language, symbols, and characters indeed required domain experts in that particular context. However, the availability of experts at the right place is not possible most of the time because of inadequate experts in this research field. In some complicated situations, the research experts themselves face complications and consume a long time for recognition. The research desperately required an efficient prediction model to resolve this issue. Hence, this research focuses on utilizing the ANN technique because of its ability to learn and model non-linear and complex relationships. The traditional ANN does not achieve remarkable performance since the randomly generated weight across the ANN layers does not yield appropriate results. Identifying appropriate weights through trial-and-error methods or manually varying all possibilities is monotonous and wastage of time to process.

1.2 Research Objectives

The significant research objective is to develop a machine learning model to recognize ancient language, symbols, and characters. The research intends to reconfigure the randomly generated weights to enhance the performance of the conventional ANN model.

1.3 Novelty of the Work

The main novelty of the work is to recognize the ancient Language, symbols and characters through ANN. OGWO is applied with ANN to identify the appropriate weights. OGWO is chosen for optimization since it includes opposition strategy to increase the chance of predicting appropriate weights over conventional GWO and other optimization techniques.

2 Related Works

Kavitha et al. [14] have exploited the ultra-modern CNN for identifying handwritten Tamil letters in disconnected manner. CNNs differ from conservative procedure of Handwritten Tamil Character Recognition (HTCR) in removing the characteristics accordingly. That work was an effort to fix a yardstick for disconnected HTCR exploiting unfathomable learning methods. That work has generated a training accurateness of 95.16% which was very well compared with the orthodox methods.

Gao et al. [15] have premeditated an enhanced grey wolf optimization procedure with varied weights (VW-GWO). Outcomes validate that the suggested VW-GWO procedure performs higher than the normal GWO, Ant Lion optimization (ALO), PSO and Bat algorithm (BA). The VW-GWO procedure was furthermore tested in high-dimensional problems.

In Zhang et al. [16] work, the engraving images collected from Tamil Nadu Archaeological Department are pre-processed and fragmented. Letters are categorized and established reliant on Vectors mined; exploiting Support Vector Machine (SVM) classifier and the designs of the letter are synchronized with recognized letters and anticipated by means of Trigram method. Hence the suggested agenda can pay attention to the severe problems in reading the engraved images.

Zhang et al. [17] have proposed that identification of unrestrained (Handwritten) Devanagari letter was increasingly difficult because of the form of essential whacks. In that article, technique like PSO and SVM are applied and associated. An android phone was exploited for taking input letter for indicating the documented Devnagari letter. For linking the android device and MATLAB, they have used PHP language. The PSO technique provides exactness up to 90%.

Sridevi et al. [18] have provided a method to achieve grouping of handwritten prehistoric writings in Tamil. The approach utilises Extreme Learning Machine for grouping of handwritten prehistoric Tamil writings. The performance of Extreme Learning Machine is compared with Probabilistic Neural Networks. From the experiment outcomes it is collected that Extreme Learning Machine provides maximum precision rate of 95%.

3 Proposed Methodology

The significant intention of this research is to develop a system for recognizing the ancient Language, symbols and characters through Artificial Intelligence (AI) techniques. This research includes ANN to recognize ancient Language, symbols, and characters. This research intends to identify appropriate weights to enhance the performance. The techniques involved in this process are PSO, GWO, and OGWO. Fig. 1 demonstrates the overall proposed process of ANN based recognition.

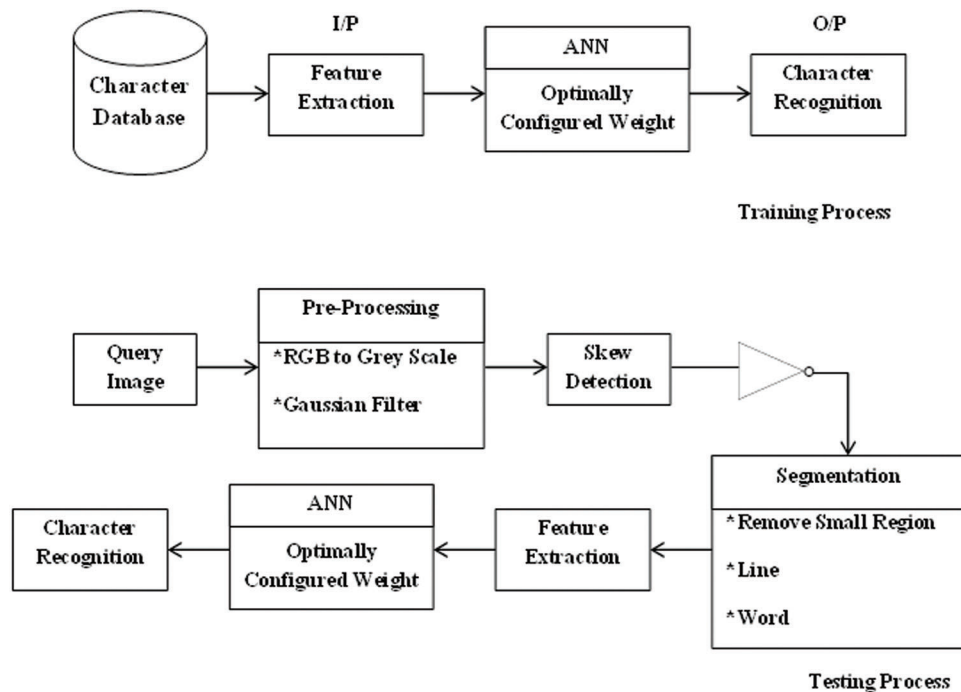


Figure 1: Block diagram of proposed framework

3.1 Training Phase

In training phase characteristics such as area, centroid (X-coordinates and Y-coordinates), bounding-box (corners and X-Y width), major axis distance, minor axis distance, eccentricity, orientation, perimeter, equiv diameter are initially removed from the manually fragmented 77-letters (listed in Tab. 3). These 13-characteristics are considered as an input factor and their equivalent letter images are considered as output class. Generally, the training procedure performs to conclude applicable masses for effective identification.

3.2 Testing Phase

Pre-processing: The trained system receives the given query image (RGB) and it pre-processes the image by converting into a gray-scale, followed by applying a Gaussian filter to remove the noises.

Skew detection: A skew detection technique is used to straighten the image; which results in the image to binary form.

Segmentation based on Region growing: While reading the whole row of a pixel, if a white pixel is noticed and it is greater than 100, it is considered out of line and continues reading the succeeding pixel rows. If the white pixel not gets noticed up, it is considered as in line. A related procedure is made centred on column-wise for separating lyrics, and the above said 13-characteristics are removed for the fragmented word and those values are provided as input for the ANN prototype with organized masses. The presentation of every active method evaluates with exactness amount for two-query investigation case images.

Feature Extraction: Above mentioned 13-features are extracted for recognizing the given query character.

3.3 Artificial Neural Network (ANN)

ANN is a mathematical learning technique utilised in cognitive psychology and artificial intelligence. The ANN is an encoded computational prototype that aims to replicate the neural construction and working of the human brain. Fig. 2 shows the ANN architecture.

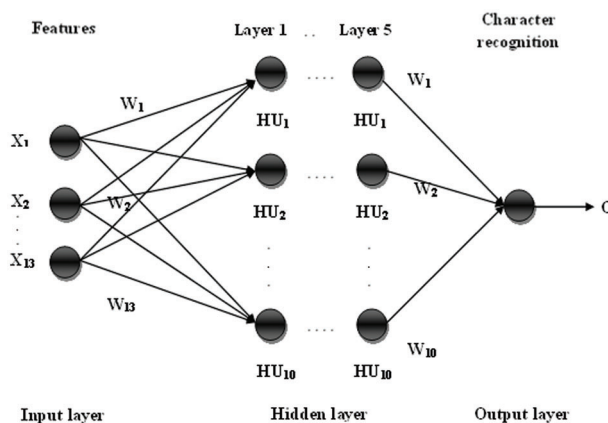


Figure 2: ANN structure

In Fig. 2, ANN contains three types of layers: input layer, hidden layer and output layer. Each layer of the ANN consists of several neurons. In this, every single neuron in the input layer is associated with the concealed layer neuron, and every concealed layer neuron associated with the output layer with an

arbitrary weight. In our ANN model, 5 hidden layers are considered with 10 neurons. These arbitrary weights are allocated to every interrelated layer.

3.3.1 Structure Initialization

In the initialization process, 13 inputs based on the input layer weight w_j and the hidden layer weights w_{ij} are initialized.

3.3.2 Input Layer

The input layer considers 13-input attributes as mentioned in the previous section. The input layer is the first layer of the neural network in which each input is connected with hidden layer neurons. The inputs I_1, I_2, \dots, I_n are applied to the input neurons u_1, u_2, \dots, u_n . Each neuron in the input layer is connected with the hidden layer neurons with random weight $w_{11}, w_{12}, \dots, w_{ij}$. The basic function of hidden neurons is evaluated by

$$X_f = \sum_{j=1}^N I_i \times w_{ij} \quad (1)$$

where X_f is a basics function, w_{ij} is an input layer weight and i is a number of input with this basic function the active function is combined.

The activation function of hidden neurons is evaluated in this sigmoidal function is exercised based on the equation shown below,

$$\tan \text{sig} (X_f) = \frac{2}{(1 + \exp(-2 * X_f)) - 1} \quad (2)$$

Multilayer perception is a sigmoidal activation function in the form of a hyperbolic tangent. The next layer of the ANN is hidden layer.

3.3.3 Hidden Layer

In this section, five numbers of hidden layers are revealed and in each hidden layer 10 neurons are performed. The number of neurons in the hidden layer is defined as Hu_1, Hu_2, \dots, Hu_n , which are connected with the output layer neuron. Each neurons in the hidden layer is connected with the output layer neurons with random weight $w_{11}, w_{12}, \dots, w_{ij}$

3.3.4 Output Layer

The hidden layer neurons are connected with the output layer and each connection has a weighted value such as w_1, w_2, \dots, w_n . The recognised character is the output of ANN. For optimizing the weight, OGWO is employed.

3.4 Opposition Based Grey Wolf Optimization Algorithm (OGWO)

When compared to traditional GWO, in OGWO, the opposition strategy utilizes to increase the probability of finding optimal weights to enhance the ANN predicting performance. The grey wolves sufficiently mount a Canidae's piece family and are respected as the top marauders presenting their place at the sustenance's food chain. They regularly display a disposition to make due as a set. The results produced by the alpha are distributed on to the set. The Beta says to the second rank in the striking sequence of the grey wolves. They are fundamentally and supplementary wolves that sufficiently provide certain aid to the alpha in the choice making or equivalent set purposes. The omega is the least bands of the grey wolf pack and it works as an ancillary proposing into the additional principle wolves very closely on every event and is allowed to have just the slight leftovers resembling a great doddle by the leader wolves. In our technique, the alpha (α) esteemed as the most suitable arrangement with a

perspective to replicating the social pecking order of wolves logically while conceiving the OGWO. Thus, the second and the third-best arrangements are named as beta (β) and delta (δ) separately. The remaining applicant arrangements are regarded to be the omega (ω). In the OGWO method, the hunting (optimization) is guided by the α , β , δ , and ω .

Fig. 3 shows the flowchart for OGWO and the following sections explain the working procedures to identify of the appropriate weights for ANN

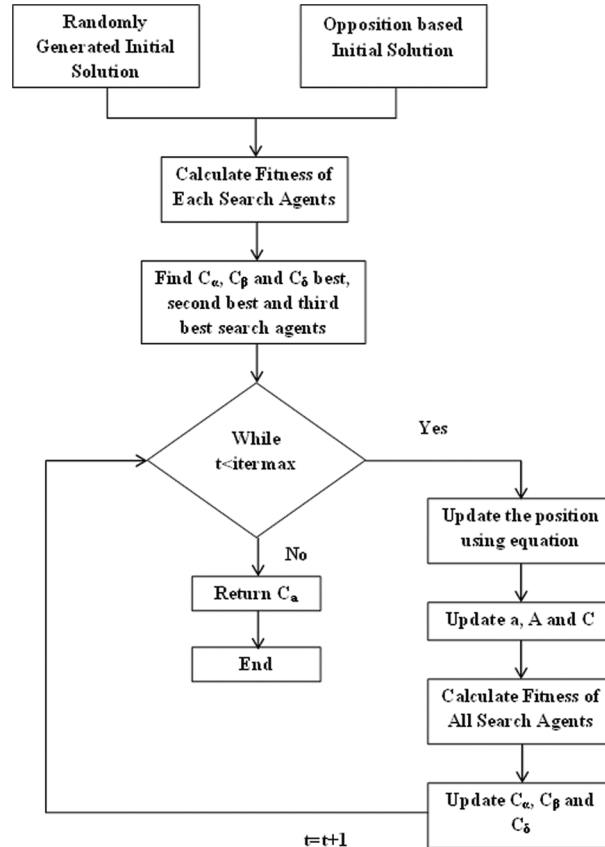


Figure 3: Flowchart of OGWO

3.4.1 Initial and Opposition Based Solution Generation

The preliminary step in the optimization process; here, the solutions is generated randomly based on constraints. The randomly generated solutions range from $[-100$ to $100]$ and length of a solution is 280. The length of the solution is mathematically expressed as follows,

$$\text{Solution length} = [NI * NH] + NH \quad (3)$$

where NI is the number of inputs (features: 13), NH is the number of hidden neurons; considered 20 (user-defined). Along with the preliminary random solution generation, this research includes opposition based solution generation, which would increase the probability of identifying an optimal weight as output solution. The mathematical expression of opposition-based solution generation is as follows.

$$Op_i = x + y - X_i \quad (4)$$

$X_i \in X_1, X_2, X_3 \dots NP$, are initial solution randomly generated suggest to generate opposition based solution where x and y are minimum and maximum values.

3.4.2 Fitness Computation

This process is utilized to evaluate the fitness of randomly and opposition based solution generation with the aid of following mathematical function.

$$Output = \sum_{j=1}^{NH} \alpha_j * \frac{1}{1 + \exp \left(\sum_{i=1}^{NI} X_i \beta_{ij} \right)} \quad (5)$$

$$Fitness (error) = Target - Output \quad (6)$$

where, X_i is known as input and α & β are called weights.

3.4.3 Updating the Position

We assume that the alpha (best candidate solution), beta and delta have improved knowledge about the potential location of the prey to reproduce the hunting behavior of the grey wolves mathematically. As a result, we hoard the first three best solutions attained so far and require the other search agents (including the omegas) to revise their positions according to the location of the best search agent. For repetition, the new solution $c(t+1)$ estimated by using the formulae mentioned below.

$$D^\alpha = |C_1 \cdot c_\alpha - c|, \quad D^\beta = |C_2 \cdot c_\beta - c|, \quad D^\delta = |C_3 \cdot c_\delta - c| \quad (7)$$

$$c_1 = c_\alpha - A_1 \cdot (D_\alpha), \quad c_2 = c_\beta - A_2 \cdot (D_\beta), \quad c_3 = c_\delta - A_3 \cdot (D_\delta) \quad (8)$$

To have hyper-spheres with different random radii, the arbitrary parameters A and C help the candidate solutions. Examination and usage are ensured by the adaptive estimations of A and a . With diminishing A , half of the iterations are committed to the investigation ($A < 1$) and the other half are dedicated to the usage. Enclosing the conduct, the subsequent equations are utilized keeping in mind the end goal to give numerical model.

$$D = |C \cdot c_p(t) - c(t)| \quad (9)$$

$$A = 2a \cdot r_1 - a, \quad C = 2 \cdot r_2 \quad (10)$$

where t indicates the current iteration, A and C are coefficient vectors, C_p is the position vector of the prey c and indicates the position vector of a grey wolf. The components of a are linearly decreased from 2 to 0 over the course of iterations and r_1, r_2 are random vectors in $[0, 1]$.

The OGWO has only two main parameters (A and C) to be adjusted. However, we have kept the OGWO algorithm as simple as possible, with the fewest operators to be adjusted. The process will be continued until the termination is obtained.

3.5 Overall Algorithm

The entire steps involved in the process of ANN-OWGO based character recognition is given in the following algorithm.

ANN-OWGO Algorithm

1. Get character database for ANN-training process
2. Extract 13-Features from the characters
3. 13-features are utilized as an input for ANN training
4. OGWO associate with ANN for identifying appropriate weights
5. Here, opposition strategy associates with conventional GWO as OGWO to increase the chance of finding appropriate solutions (weights).
6. Finally, ANN will recognize the character for a given features value
7. The above mentioned steps are the fundamental of following testing process
8. The trained ANN model receives aw query image
9. For pre-processing, the trained system received a given query image (RGB) and pre-processed the image by converting it to a gray-scale, followed by applying a Gaussian filter to remove noises
10. A skew detection technique is used to straighten the image; results in the image to binary form.
11. Segmentation: line segmentation, while reading the entire row of a pixel, if white pixel detected and it is higher than 100, which is considered for line and keeps reading the subsequent pixel rows if the white pixel not gets detected up to that is regarded as the line. A similar process is done based on column-wise for segregating words, and the above said 13-features are extracted for the segmented word and those values given as input for the ANN model with configured weights. The performance of each employed technique evaluates with accuracy measure for two-query test case images.

4 Results and Discussion

This section discusses the different results are compared with existing techniques. Apart from OGWO, the other optimization techniques used in ANN weight determination are PSO and GWO. Accuracy and the convergence graphs are explained in the below sections. In [Tab. 1](#), sample test case 1, [Tab. 2](#) sample test case 2 and [Tab. 3](#) input features extracted from concerned characters are presented. [Tabs. 4](#) and [5](#) presents the performance results of the approaches.

Table 1: Sample test case 1

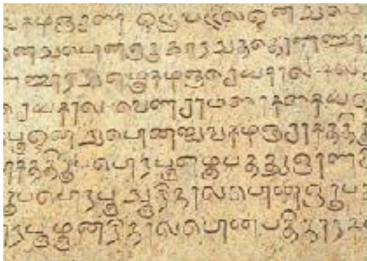
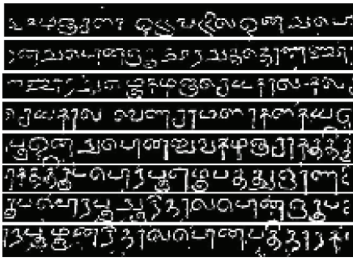

Actual image	Line image	Word image
		

Table 2: Sample test case 2

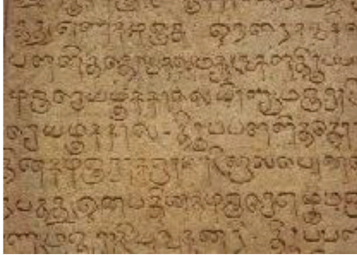
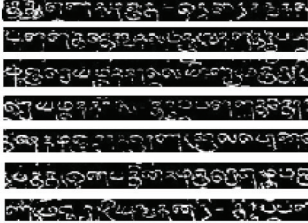









Actual image	Line image	Word image
		

Table 3: Features extracted from concern characters

Image	Area	Centroid		Bounding box				Major axis length	Minor axis length	Eccentricity	Orientation	Perimeter	Equiv diameter
	656	25.15	45.35	1.5	22.5	48	43	56.14	49.08	0.48	31.80	374.3	28.90
	763	36.33	44.78	1.5	22.5	67	48	75.28	47.53	0.77	5.163	274.24	31.16
	639	31.31	27.20	1.5	6.5	59	44	71.71	37.00	0.85	-26.57	429.78	28.52
	482	27.14	53.29	1.5	35.5	49	39	60.67	41.71	0.72	4.43	222.19	24.77
	480	25.31	67.16	1.5	48.5	43	46	59.65	45.90	0.63	-54.21	250.71	24.72
	461	20.42	56.59	1.5	36.5	40	41	54.76	29.69	0.84	53.11	137.98	24.22
	606	19.31	23.34	1.5	0.5	35	41	45.16	35.45	0.61	62.45	187.36	27.77
	255	23.03	24.65	1.5	11.5	43	27	45.90	22.18	0.87	13.51	211.74	18.01

(Continued)

Table 3 (continued)

Image	Area	Centroid		Bounding box				Major axis length	Minor axis length	Eccentricity	Orientation	Perimeter	Equiv diameter
	221	15.61	55.38	1.5	32.5	36	36	40.25	30.63	0.64	-84.67	143.57	16.77
	194	16.19	34.67	1.5	24.5	30	23	38.32	24.66	0.76	12.17	131.64	15.71
	569	22.14	68.69	1.5	43.5	39	61	64.55	41.84	0.76	-64.91	241	26.91
	428	19.75	25.14	1.5	3.5	40	44	57.34	33.33	0.81	48.44	217.74	23.34
	1021	33.23	54.83	1.5	23.5	62	57	70.94	58.89	0.55	28.76	407.95	36.05
	382	15.21	28.95	1.5	16.5	31	25	29.73	24.81	0.55	38.09	116.32	22.05
	529	23.51	51.52	1.5	34.5	43	39	51.95	42.26	0.58	-15.35	209.93	25.95
	591	24.63	54.71	1.5	33.5	48	40	56.06	41.55	0.67	-2.95	255.99	27.43
	554	28.42	42.58	1.5	26.5	53	34	64.54	37.91	0.80	0.15	289.63	26.55
	446	18.30	57.71	1.5	38.5	35	34	42.94	36.32	0.53	32.80	159.22	23.82
	717	31.01	61.85	1.5	43.5	56	37	63.92	37.91	0.80	-1.95	252.35	30.21












(Continued)

Table 3 (continued)

Image	Area	Centroid		Bounding box				Major axis length	Minor axis length	Eccentricity	Orientation	Perimeter	Equiv diameter
	716	24.57	66.76	1.5	44.5	53	41	58.45	40.46	0.72	-7.15	276.47	30.19
	388	21.86	19.10	1.5	7.5	44	24	53.94	24.32	0.89	-9.70	250.09	22.22
	518	26.66	30.77	1.5	17.5	56	27	61.69	28.56	0.88	-0.05	214.65	25.68
	653	20.83	54.63	1.5	31.5	42	45	47.13	45.74	0.24	62.80	154.29	28.83
	295	14.98	68.77	1.5	50.5	27	33	39.04	32.02	0.57	35.80	152.71	19.38
	415	16.02	51.41	1.5	32.5	32	41	51.84	30.14	0.81	64.25	200.36	22.98
	363	16.88	38.66	1.5	24.5	37	41	42.87	36.81	0.51	27.91	151.76	21.49
	146	14.17	55.88	1.5	37.5	36	27	46.21	20.31	0.89	-22.26	112.92	13.63
	595	17.47	64.96	1.5	41.5	34	56	59.11	34.12	0.81	80.08	252.18	27.52
	170	16.1	40.96	1.5	25.5	40	23	49.32	20.67	0.90	-8.61	123.16	14.71
	169	11.56	22.017	1.5	13.5	26	18	25.48	13.89	0.83	-25.31	91.12	14.66

(Continued)

Table 3 (continued)

Image	Area	Centroid		Bounding box				Major axis length	Minor axis length	Eccentricity	Orientation	Perimeter	Equiv diameter
	289	25.51	31.41	1.5	8.5	40	38	50.11	37.66	0.65	27.23	228.43	19.18
	513	17.94	52.47	1.5	34.5	38	36	42.08	38.75	0.38	9.22	138.94	25.55
	749	22.74	58.73	1.5	38.5	47	43	52.85	46.06	0.49	-11.49	184.17	30.88
	735	21.87	51.75	1.5	11.5	41	66	71.38	40.66	0.82	75.30	238.45	30.59
	981	19.75	47.24	1.5	11.5	47	65	64.50	47.49	0.67	77.35	295.80	35.34
	636	33.01	36.19	1.5	18.5	89	30	86.37	28.36	0.94	-10.49	414.34	28.45
	539	25.56	48.99	1.5	30.5	45	37	51.60	39.85	0.63	-0.55	249.71	26.19
	647	30.98	39.12	1.5	23.5	60	36	69.93	40.04	0.81	6.46	353.85	28.70
	474	25.02	29.27	3.5	2.5	38	50	51.83	42.55	0.57	83.86	366.18	24.56
	318	17.26	46.98	1.5	30.5	35	34	35.79	31.84	0.45	-54.89	125.55	20.12
	462	29.63	22.17	1.5	5.5	47	35	56.72	32.12	0.82	-3.68	305.83	24.25












(Continued)

Table 3 (continued)

Image	Area	Centroid		Bounding box				Major axis length	Minor axis length	Eccentricity	Orientation	Perimeter	Equiv diameter
	550	23.08	54.41	1.5	33.5	45	38	52.34	43.75	0.54	9.78	206.47	26.46
	814	26.86	49.02	1.5	19.5	54	59	63.08	59.69	0.32	-73.95	430.15	32.19
	319	29.56	35.90	14.5	20.5	36	27	39.41	26.11	0.74	20.10	135.51	20.15
	812	34.93	57.55	1.5	38.5	70	37	77.49	37.24	0.87	11.78	256.76	32.15
	929	39.98	51.58	1.5	31.5	77	39	90.65	36.44	0.91	7.08	380.80	34.39
	816	40.10	49.46	1.5	30.5	77	38	87.48	42.70	0.87	4.14	344.92	32.23
	698	27.81	30.89	1.5	1.5	51	66	72.51	43.10	0.80	45.81	463.65	29.81
	377	23.41	57.09	1.5	45.5	47	24	52.46	26.10	0.86	6.75	181.60	21.90
	448	16.37	34.01	1.5	15.5	34	53	51.51	36.27	0.71	75.22	280.28	23.88
	499	24.45	48.40	1.5	30.5	52	38	50.85	34.23	0.73	-2.89	320.21	25.20
	652	46.16	26.06	1.5	9.5	90	33	96.36	27.32	0.95	-3.12	337.82	28.81












(Continued)

Table 3 (continued)

Image	Area	Centroid		Bounding box				Major axis length	Minor axis length	Eccentricity	Orientation	Perimeter	Equiv diameter
	649	29.19	63.27	1.5	40.5	54	50	68.71	47.71	0.71	-27.82	418.28	28.74
	305	20.92	64.80	1.5	45.5	37	34	46.77	32.75	0.71	28.63	171.55	19.70
	253	29.24	23.78	11.5	8.5	30	38	43.98	27.74	0.77	-49.68	187.17	17.94
	539	28.09	67.82	1.5	42.5	44	56	54.69	51.45	0.33	21.31	247.70	26.19
	473	18.34	61.88	1.5	39.5	38	47	56.98	41.85	0.67	60.44	247.65	24.54
	223	23.42	34.46	8.5	24.5	38	22	47.74	21.29	0.89	8.90	171.90	16.85
	100	10.14	18.5	1.5	11.5	21	17	25.28	16.26	0.76	31.07	88.87	11.28
	627	21.78	30.75	1.5	0.5	45	69	66.43	47.02	0.70	-78.76	379.11	28.25
	565	17.68	53.26	1.5	25.5	33	52	57.47	37.20	0.76	-89.06	318.46	26.82
	778	27.56	69.63	1.5	45.5	57	50	68.24	48.79	0.69	24.17	299.13	31.47
	774	23.15	59.12	1.5	31.5	48	68	73.89	47.75	0.76	77.95	300.60	31.39

(Continued)

Table 3 (continued)

Image	Area	Centroid		Bounding box				Major axis length	Minor axis length	Eccentricity	Orientation	Perimeter	Equiv diameter
	379	18.51	43.68	3.5	21.5	32	52	45.30	29.84	0.75	77.64	233.81	21.96
	1092	25.49	66.03	1.5	36.5	59	64	73.34	63.22	0.50	51.57	398.65	37.28
	631	30.75	50.55	1.5	27.5	67	41	65.90	46.84	0.70	-24.29	391.63	28.34
	706	29.17	64.00	1.5	44.5	57	38	65.05	43.83	0.73	-4.58	397.34	29.98
	203	21.95	37.65	1.5	23.5	38	27	48.55	22.96	0.88	-33.95	169.55	16.07
	226	14.49	17.46	1.5	4.5	27	25	33.35	22.19	0.74	15.11	161.42	16.96
	429	28.09	38.73	1.5	14.5	53	52	63.46	36.17	0.82	-15.14	259.96	23.37
	555	24.03	56.21	1.5	40.5	48	33	50.88	35.11	0.72	0.64	248.51	26.58
	511	28.81	46	1.5	17.5	49	51	62.55	45.58	0.68	62.86	382.73	25.50
	172	22.31	45.09	1.5	32.5	35	21	38.91	22.30	0.819	17.33	159.72	14.79
	313	22.74	33.68	1.5	18.5	35	36	46.64	29.38	0.77	-17.89	184.93	19.96

(Continued)

Table 3 (continued)





Image	Area	Centroid		Bounding box				Major axis length	Minor axis length	Eccentricity	Orientation	Perimeter	Equiv diameter
	236	19.09	31.63	1.5	16.5	32	33	35.57	25.71	0.69	-41.02	171.20	17.33
	239	20.12	50.74	1.5	33.5	33	39	40.14	30.86	0.63	-54.34	204.71	17.44
	260	18.58	41.25	1.5	24.5	30	36	42.38	26.16	0.78	-40.49	159.39	18.19
	575	22.38	39.30	1.5	11.5	42	51	52.82	42.51	0.59	-60.12	220.23	27.05

Table 4: Result table

Sample data	ANN-OGWO	ANN-GWO	ANN-PSO	ANN
1	0.95	0.89	0.82	0.73
2	0.91	0.88	0.82	0.7

Table 5: Result table

Iteration	ANN-OGWO	ANN-GWO	ANN-PSO
50	0.44	0.44	0.44
100	0.38	0.44	0.44
150	0.38	0.23	0.44
200	0.22	0.23	0.3
250	0.15	0.23	0.3
300	0.15	0.23	0.25
350	0.15	0.2	0.25
400	0.04	0.2	0.25
450	0.04	0.2	0.25
500	0.04	0.2	0.25

From the Fig. 4 illustrated the proposed OGWO is having superior accuracy over other techniques. In case 1, the accuracy value for OGWO is 94.89%, for GWO having 89%, for PSO having 85%, and for default

having 73%. In case 2, the accuracy value for OGWO in the validation above is 92.34%, for GWO having 86.7%, for PSO having 82% and for default having 70%. On average, the accuracy of OGWO achieves 5.8% greater accuracy than GWO, 10.1% greater accuracy than PSO, and 22.1% greater accuracy over conventional ANN technique (Randomly configured weights). In general, the incorporation of optimization techniques to configure ANN weights consequences the performance improvement. Employing optimization techniques to identify appropriate weights and integration of opposition strategy in GWO ensure the proficient performance of OGWO associate ANN.

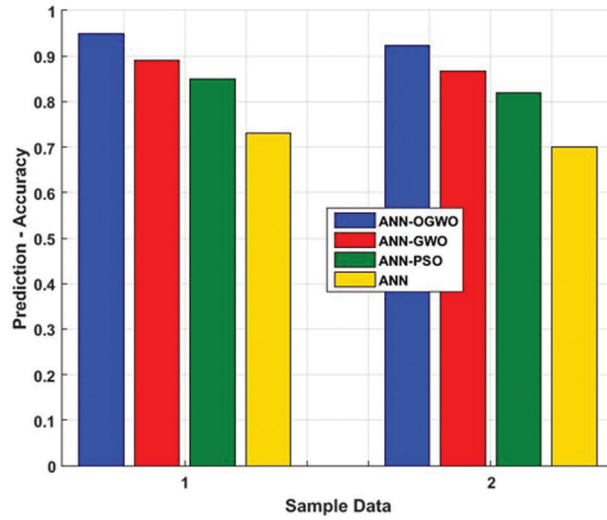


Figure 4: Comparison accuracy graph

This section Fig. 5, carryout different optimization technique involves in designing the ANN structure namely OGWO, GWO, and PSO amid OGWO converge faster than other at the same time fitness error reveal by the proposed OGWO also good compare with contest technique. The straight competition between PSO and GWO up to 100th iteration, then GWO varies low than PSO, and both saturate error value at 350th to 500th iteration. OGWO slightly saturate up to 400th iteration to 500th iteration, and the error value is lower than other two algorithms due to the optimization of ANN weights using OGWO. Employing optimization techniques to identify appropriate weights and integration of opposition strategy in GWO ensure the proficient performance of OGWO associate ANN

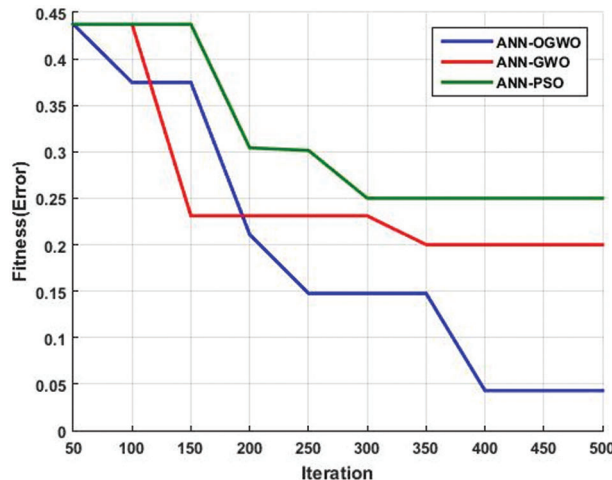


Figure 5: Convergence graph

5 Conclusion

Resolving the research problem through redesigned ANN associated with optimization techniques accomplished effectively. Incorporation of opposition strategy in conventional GWO literally enhances the performance even better over other techniques for recognizing the ancient language, symbols and characters. The results reveal that the proposed method attains accuracy as 93.6%, which is 5.8% greater than GWO, 10.1% greater than PSO, and 22.1% superior to conventional ANN model (Auto-configured weights). In the future, this research intends to reconfigure the structure of ANN to enhance predicting performance further.

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

- [1] S. Aswatha, A. Talla, J. Mukhopadhyay and P. Bhowmick, "A method for extracting text from stone inscriptions using character spotting," *Indian Institute of Technology Kharagpur*, vol. 5, no. 2, pp. 1–14, 2014.
- [2] H. Winskel, "Reading and writing in southeast asian languages," *Procedia-Social and Behavioural Sciences*, vol. 97, pp. 437–442, 2019.
- [3] S. Franklin Thambi Jose and J. Samikkanu, "Criteria for the identification of adverbial phrases for tamil," *Procedia-Social and Behavioral Sciences*, vol. 208, pp. 138–142, 2019.
- [4] E. Vellingiriraj and P. Balasubramanie, "A novel hybrid neural learning based tamil handwritten character recognition system in palm manuscripts for siddha medicine," *Journal of Advance Research in Dynamical & Control Systems*, vol. 9, no. 3, pp. 152–164, 2018.
- [5] G. Bhuvanewari and V. Subbiah Bharathi, "Recognition of ancient stone inscription characters using normalized positional distance metric features," *Asian Journal of Research in Social Sciences and Humanities*, vol. 6, no. 5, pp. 604–615, 2018.
- [6] D. Pugazhenthii and S. Arul Vallarasi, "Offline character recognition of printed tamil text using template matching method of bamini tamil font," *Indian Journal of Science and Technology*, vol. 8, no. 35, pp. 1–4, 2015.
- [7] N. Panyam, T. Vijaya Lakshmi, R. Krishnan and N. Koteswara Rao, "Modelling of palm leaf character recognition system using transform based techniques," *Pattern Recognition Letters*, vol. 84, pp. 29–34, 2018.
- [8] A. Barmpoutis, E. Bozia and S. Wagman, "A novel framework for 3D reconstruction and analysis of ancient inscriptions," *Machine Vision and Applications*, vol. 21, pp. 989–998, 2017.
- [9] S. Rajakumar and V. Subbiah Bharathi, "Eighth century tamil consonants recognition from stone inscriptions," in *Int. Conf. on Recent Trends in Information Technology*, China, pp. 40–43, 2019.
- [10] G. Bhuvanewari and V. Subbiah Bharathi, "An efficient positional algorithm for recognition of ancient stone inscription characters," in *Seventh Int. Conf. on Advanced Computing*, Bangkok, pp. 1–5, 2019.
- [11] N. Shanthi and K. Duraiswamy, "A novel SVM-based handwritten tamil character recognition system," *Pattern Anal Application*, vol. 13, pp. 173–180, 2019.
- [12] S. Manikandan, A. Delphin Carolina Rani, C. Rajeswari, T. Suma and D. Sivabalaselvamani, "Recognition of font and tamil letter in images using deep learning," *Applied Computer Science*, vol. 17, no. 2, pp. 90–99, 2021.
- [13] R. Giridharan, E. Vellingiriraj and P. Balasubramanie, "Identification of tamil ancient characters and information retrieval from temple epigraphy using image zoning," in *Int. Conf. on Recent Trends in Information Technology*, Bangkok, pp. 1–7, 2019.
- [14] B. Kavitha and C. Srimathi, "Benchmarking on offline handwritten tamil character recognition using convolutional neural networks," *Journal of King Saud University-Computer and Information Sciences*, vol. 5, no. 3, pp. 1–8, 2018.
- [15] Z. Gao and J. Zhao, "An improved grey wolf optimization algorithm with variable weights," *Computational Intelligence and Neuroscience*, vol. 2019, pp. 1–13, 2020.

- [16] X. R. Zhang, W. F. Zhang, W. Sun, X. M. Sun and S. K. Jha, "A robust 3-D medical watermarking based on wavelet transform for data protection," *Computer Systems Science & Engineering*, vol. 41, no. 3, pp. 1043–1056, 2022.
- [17] R. Zhang, X. Sun, X. M. Sun, W. Sun and S. K. Jha, "Robust reversible audio watermarking scheme for telemedicine and privacy protection," *Computers, Materials & Continua*, vol. 71, no. 2, pp. 3035–3050, 2022.
- [18] N. Sridevi and P. Subashini, "Combining zernike moments with regional features for classification of handwritten ancient tamil scripts using extreme learning machine," in *IEEE Int. Conf. on Emerging Trends in Computing, Communication and Nanotechnology*, Japan, vol. 14, no. 3, pp. 158–162, 2017.