International Journal on Advanced Science Engineering Information Technology

Vol.15 (2025) No. 3
ISSN: 2088-5334

Honey Badger Freeman Chain Code (HB-FCC) Feature Extraction Algorithm for Handwritten Character Recognition

Muhammad Arif Mohamad [a,1], Zuriani Mustaffa [a], Muhammad Faizan [a], Shahdatunnaim Azmi [b,2], Nor Amalina Mohd Sabri [b]

[a] *Faculty of Computing, Universiti Malaysia Pahang Al-Sultan Abdullah, Pekan, Pahang, Malaysia*
[b] *Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia, Parit Raja, Johor, Malaysia*
Corresponding author: [1]arifmohamad@umpsa.edu.my; [2]shahdatun@uthm.edu.my

***Abstract*—This study proposed the Honey Badger Freeman Chain Code (HB-FCC) feature extraction algorithm, an approach for feature extraction for Handwritten Character Recognition (HCR). HCR is a critical area of pattern recognition; to make it effective with minimal error and time consumption, the method of feature extraction must be robust enough. Traditional methods using the FCC have been proven efficient, but several drawbacks, including sensitivity to noise, dependence on initial conditions, and computational complexity, also limit them. To address the challenges mentioned above, this study proposes an improved FCC method utilizing the Honey Badger Algorithm (HBA), a recently developed metaheuristic optimization approach. The proposed HB-FCC algorithm focuses on optimizing the process of chain code generation using converted character images to graphs, where HBA is applied to minimize the length of FCCs and enhance recognition performance. The proposed method, in this regard, involves transforming a binary image of a handwritten character into a graph by representing the key points in a character's structure as nodes and the connections between these points as edges. These feature points are thus ranked according to the best paths identified by the HBA, thereby reducing the computational load and increasing the resilience of the feature extraction process against variations in handwriting and noise. The proposed HB-FCC feature extraction algorithm was further tested using a CEDAR dataset of handwritten characters. It demonstrated significant improvements in both computational time and route length compared to conventional FCC methods. This fact indicates the potential of HB-FCC in enhancing accuracy and efficiency in the HCR system.**

***Keywords*—Metaheuristic optimization; honey badger algorithm; freeman chain code; handwritten character recognition.**

Manuscript received 22 Oct. 2024; revised 14 Jan. 2025; accepted 12 Apr. 2025. Date of publication 30 Jun. 2025.
IJASEIT is licensed under a Creative Commons Attribution-Share Alike 4.0 International License.

I. INTRODUCTION

Handwritten character recognition (HCR) is a crucial area in pattern recognition, with applications ranging from automated postal sorting to the digitization of historical manuscripts. In the field of pattern recognition, with a variety of related applications, Character recognition is essential and has become increasingly challenging. It is also a natural form of interaction between humans and computers, and it has been a topic of research in computer science since its early days [1]. The process of detecting and recognizing characters from an input image into ASCII or another machine-editable format is known as character recognition [2], [3].

The effectiveness of HCR systems is primarily based on robust processes of feature extraction and selection. These are the processes that significantly enhance the system's importance, as they impact accuracy and efficiency. The feature extraction stage in HCR represents the process of transforming raw data into a set of measurable properties or features of the text that can be useful in classification. This stage is also crucial because the quality of the extracted features significantly impacts the performance of the subsequent classification phase. The traditional methods for extracting features include statistical, structural, and syntactic approaches. The statistical methods focus on the distribution of pixel values, including zoning and projection histograms. Structural methods, which include contour and skeleton extraction, emphasized the geometric and topological properties of characters. Syntactic methods describe the structure of characters with the use of grammatical rules. One of the good reviews related to the feature extraction methods in HCR was discovered in [4].

Chain code is a traditional feature extraction technique under the structural approach category and has been used widely in HCR. The first chain coding technique, called

Freeman Chain Code (FCC) is one boundary extraction-based representation technique that is helpful for image processing, shape analysis, and pattern recognition [5]. Meanwhile, the chain code provides boundaries of the character image. The codes indicate the direction and location of the subsequent pixel [6].

This can represent the boundary of a character by a sequence of directional codes, which captures the structural information of the character. The primary benefit of using the chain code method is its ability to provide a compact and effective representation of character shapes, which eases the process of matching and recognition. Furthermore, chain codes are invariant and hence can be easily used for the recognition of characters that have been written in a different orientation or position. Recent studies have demonstrated that chain code-based feature extraction is effective in enhancing the accuracy of HCR systems. For example, a research study by [7] found that the use of chain code, along with correlation coefficients, led to improved recognition performance. Another study highlighted the use of metaheuristic optimization methodologies with chain code representation to enhance feature extraction processes further [8]. Such advancements underscore the continued relevance and potential of the chain code method in modern HCR applications.

Unfortunately, the chain coding approach, while effective in HCR feature extraction, also has several drawbacks. The most crucial thing is that it is sensitive to noise and handwriting variations. Slight distortions, which cause significant changes in the chain code, result in loss of invariance of the method concerning the character, leading to a fall in recognition accuracy [9]. Additionally, the procedure can be computationally expensive because it must process every pixel on the boundary for complex and highly detailed characters [8]. Other problems include reliance on the start point of the chain code: different starting points would result in different codes for the same shape, making it very difficult to match them further [10]. In addition, chain codes are much less effective for characters with many branches or intersections, since it is at this stage that the method finds it difficult to maintain continuity without restarting at junctions [10]. These limitations, in turn, underscore the need for effective and robust extraction techniques within contemporary HCR systems.

Indeed, by avoiding the inherent limitations of the chain code method, metaheuristic optimization approaches might optimize the efficiency in feature extraction for HCR. Among them are algorithms such as the Genetic Algorithm (GA), Particle Swarm Optimization (PSO), and Ant Colony Optimization (ACO), which are designed for complex search spaces and aim to find near-optimal solutions without requiring gradient information. By embedding metaheuristic optimization within the chain code technique, it should become feasible to obtain optimal starting point locations, thereby decreasing the sensitivity to noise and variations in handwriting. This will yield more robust and accurate feature extraction as a consequence of calibration via optimization of the chain code parameters, which better represent features that play an essential role in character structure. Besides, metaheuristic algorithms can also aid in enhancing computational efficiency near character boundaries at a detailed level, thus easing recognition that is faster and more reliable. This synergy between the conventional feature extraction methodology and advanced optimization underlies the potential for developing more effective and resilient HCR solutions [11].

In this study, the synergy of traditional methods of feature extraction and the advanced approaches used in the metaheuristic method for optimization represents the potential in devising better, more effective, and more robust feature extraction solutions. In the feature extraction process of chain codes used for extracting characters, a metaheuristic-based approach to chain code features is employed [12]. The study proposed feature extraction based on chain code using metaheuristic techniques with the Honey Badger Algorithm (HBA). HBA mimics the foraging behavior of the honey badger. HBA captures dynamic search tactics because it is crucial for effective search to maintain a balance between exploration and exploitation. This feature enables HBA to address challenging optimization issues with numerous local areas, as it maintains sufficient population diversity during the search phase to explore a wide area of the provided environment [13].

HBA is applied to construct FCC; it could form the chain code that adequately describes the character image to solve these issues. Furthermore, a solution representation based on graph theory is presented for the proposed metaheuristic feature extraction algorithm. The essential concept from the above-mentioned metaheuristic algorithm is that the starting point of the chain code does not need to be specified.

II. MATERIALS AND METHOD

A. Related Works

The domain of handwritten character recognition has witnessed substantial advancements, particularly through the application of metaheuristic algorithms for feature selection. A review by [14] details multiclass feature selection using metaheuristic optimization algorithms. Therefore, it is essential to note that this type of algorithm can effectively adapt to highly complex feature selection problems. These studies identify various metaheuristic techniques, including genetic algorithms (GA), particle swarm optimization (PSO), and ant colony optimization (ACO), and their applications in diverse scenarios. According to [15], it outlines several metaheuristic techniques of feature selection, while underscoring its robustness and applicability to almost all application domains. Their work presented a detailed comparison of [16] state-of-the-art algorithms that include GA, PSO, and ACO, exhibiting their potential to improve the performance of handwriting character recognition systems significantly. While [17] does a comparative analysis of a variety of feature extraction techniques for handwritten character recognition, including those augmented by metaheuristic algorithms. Metaheuristic-based techniques for feature selection are superior to traditional ones in terms of prediction accuracy and computational efficiency [18].

Also, [10] conduct their research in the field of feature selection techniques and are deemed to aid in the enhancement of accuracy in the recognition of handwritten characters. They have emphasized the capability of metaheuristic algorithms in optimizing feature subsets, which is expected to increase recognition performance and reduce computational costs. The researchers have tested various metaheuristic techniques, which combine GA and PSO, and evaluated their efficiency in different scenarios. On the contrary, [19] worked on the extraction and selection of features for the recognition of handwritten Gujarati characters using metaheuristic algorithms to optimize the process of feature selection. Their results show that they effectively enhanced the accuracy of recognition by selecting the most pertinent features. A study by [20] introduced a new dimension of feature extraction for handwritten Arabic character recognition, involving metaheuristic algorithms for optimum feature selection. The study has shown that this approach can significantly enhance both the accuracy and efficiency of recognition. The unique challenges of Arabic OCR involve handling diacritics and variations in character shapes [21]. This is demonstrated by the implementation of metaheuristic algorithms to handle such challenges and enhance the performance of recognition systems. Finally, [22] addresses feature extraction for deep learning in the context of handwritten Chinese character recognition and employs metaheuristic algorithms for optimizing feature selection. Their results have shown that this combination can enhance recognition performance by selecting the most relevant features. The complexities of Chinese character recognition, including the vast number of characters and intricate stroke patterns, are discussed [23]. How metaheuristic algorithms can be applied to select features effectively, thereby reducing these complexities in representation and increasing recognition accuracy have been illustrated.

Therefore, drawing inspiration from previous works, this study has attempted to implement a metaheuristic optimization technique as a chain code feature extraction algorithm to derive solutions for optimized chain code generation. HBA is one of the most recent metaheuristic approaches by [13]. The HBA is a novel, nature-inspired metaheuristic optimization algorithm that imitates the foraging behavior of honey badgers. It initializes a population of agents in the search space, and each agent represents a solution to the problem at hand. An objective function evaluates the fitness value of the agents [24]. HBA consists of two main phases of operation: the digging phase, which is the local search and exploitation phase, and the honey phase, which emphasizes global search and exploration. Such stages have been inspired by the digging behavior of honey badgers and their cooperative interactions with honeyguide birds. The algorithm updates parameters and agent positions dynamically based on the current iteration, ensuring an optimal balance between exploration and exploitation is achieved until a stopping condition is met [25].

In contrast to other algorithms of a similar nature, there are several benefits for the HBA. It explores the search space in an essentially balanced manner, simultaneously reducing the risk of being trapped in local optima. The dynamic parameter updating and efficient search strategies described provide the algorithm with properties such as faster convergence, i.e., increased time efficiency. HBA has been observed to exhibit high accuracy in solving complex optimization problems, which enables its application to a broad range of problems, from engineering design to image segmentation [26]. Furthermore, HBA required a few parameters to be set up in the initialization step [27]. Therefore, this makes HBA easy to implement and few are liable to overfit. Moreover, the findings of these techniques revealed the fact that HBA possesses versatility and expandability, which enables the HBA to adjust to various optimization difficulties.

B. Proposed HBA-FCC Feature Extraction

This section details the procedure of the development of the honey badger algorithm freeman chain coding (HBA-FCC) feature extraction algorithm. The procedure of developing the proposed HBA-FCC is illustrated in Fig. 1.

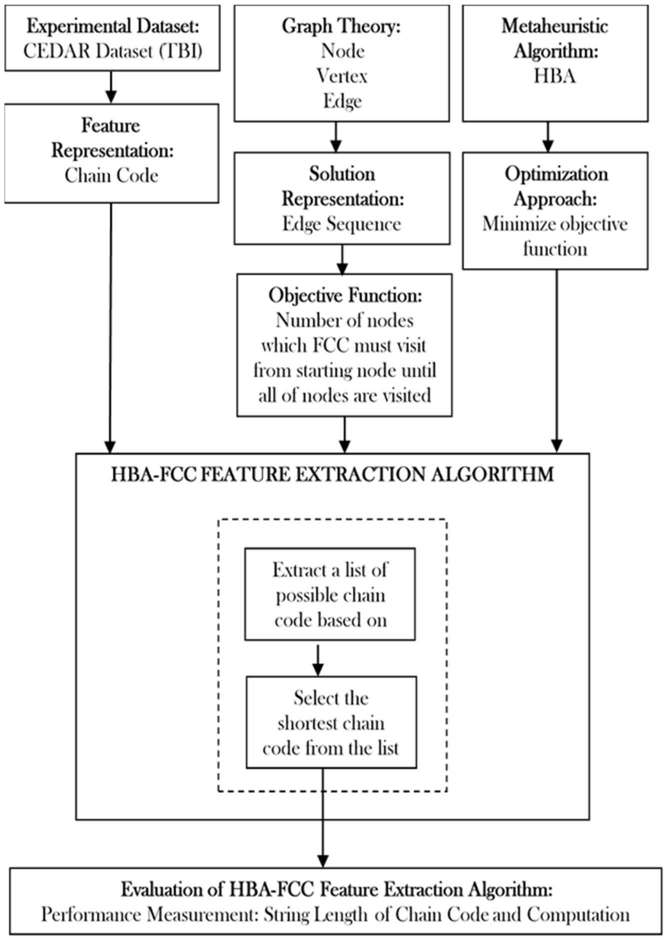


Fig. 1 Procedure of Development of the proposed HBA-FCC Feature Extraction Algorithm

As shown in Fig. 1, the proposed HBA-FCC extracts the chain code from the character image in the form of a thinned binary image (TBI), as shown in Fig. 2. The proposed HBA-FCC feature extraction technique has the following phases to construct the chain code, as shown below:

i. Transform characters into graphs.
ii. Use graphs to represent solutions.
iii. HBA metaheuristics approach to optimize FCC.

1) Character Transformation into Graph

A technique of generating chain codes from a binary image of a character can be modeled as a graph routing issue. As a result, the binary image is converted into a set of vertices with connecting edges. The graph's vertices are chosen from one of two options: nodes with only one neighbor (endpoint) or nodes with more than two neighbors. A red circle shows the first form of vertex identification for a character picture, whereas the latter is represented by a blue circle, as shown in Fig 3(a). Furthermore, the edges of the graph begin at nodes with two neighbors connecting the preceding vertices (edge directional duality). The vertices identification for “B1” as a complete graph is shown in Fig 3(b).

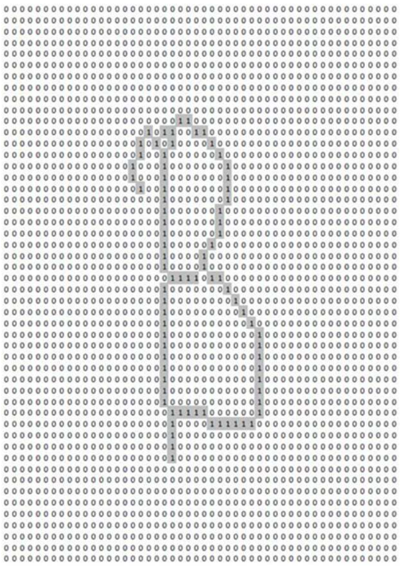
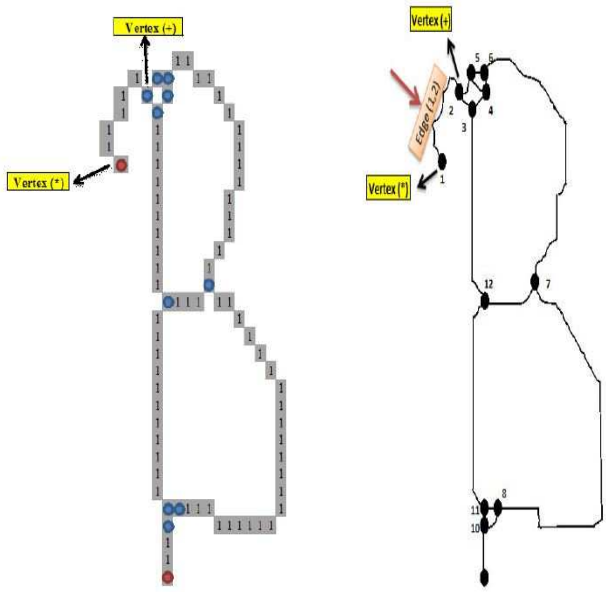


Fig. 2 Sample of TBI



(a) (b)

Fig. 3 Character Transformation of Fig. 2 Into Graph

The representation solution is offered as a directed graph (digraph) because the starting FCC vertex must be specified. A digraph is a graph with all of its edges directed in a certain way. As a result, Table 1 lists all of the directed graph's edges along with their respective lengths. A solution for connecting all edges in continuous flow can be built using the sample digraph in Fig 3(b) and its related edge pairing list in Table 1. According to Figure 3(a), there are 81-pixel nodes. Starting with Vertex 1 (*), the traverse continues until Vertex 2 (+) is located. The total amount of pixels between them is referred to as "length". This set of pixels referred to as "Edge" will be labeled as one (refer row 1 of Table 1).

Next, it resumes its journey at Vertex 2, which is marked with (+). Three vertices are related to Vertex 2. The initial vertex linked from Vertex 2 had previously visited Vertex 1, however the edge is now labelled as 2 (as in row 2, Table 1) to reflect the different direction. The second vertex linked from Vertex 2 to Vertex 3 has a length of 0 and an edge labelled as 3 (as illustrated in row 3 of Table 1). The third vertex linked from Vertex 2 to Vertex 5 (as shown in Fig 3(b)) has a length of 0 and an edge labelled 5 (refer row 5 of Table 1). It should be noticed that an edge length of 0 indicates that there is no node connecting those vertices. After all pixels related to existing vertex are traversed, the next step is to look for any untraversed vertex in the graph. The adventure continues until all pixels have been traversed.

TABLE I
EDGES AND THEIR LENGTH

Start Vertex	End Vertex	Length	Edge
1(*)	2(+)	5	1
2(+)	1(*)	5	2
2	3	0	3
3	2	0	4
2	5	0	5
5	2	0	6
3	4	0	7
4	3	0	8
3	12	10	9
12	3	10	10
4	5	0	11
5	4	0	12
4	6	0	13
6	4	0	14
5	6	0	15
6	5	0	16
6	7	0	17
7	6	15	18
7	8	15	19
8	7	23	20
8	10	23	21
10	8	0	22
8	11	0	23
11	8	0	24
10	9	2	25
9	10	2	26
11	10	0	27
10	11	0	28
11	12	11	29
12	11	11	30
12	7	3	31
7	12	3	32

2) Graph Solution and Representation

The proposed feature extraction algorithm uses a sequence of edges to represent the FCC solutions. To avoid misidentification under the following scenarios, the edge (rather than the vertex) is employed as the solution representation.

i. An edge is derived and terminate at the same node.
ii. Two edges can originate from the same node and connect to another node.

In addition, the proposed feature extraction algorithm employs a more relaxed solution representation. They assume that one edge can be visited twice to ensure that the solution representation can complete a full tour, as returning to previously visited nodes is frequently required. Fig 4 depicts an example of the solution representation.

26	22	20	18	14	11	6	2	1	3
9	30	24	20	32	10	4	2	1	5
12	13	17	32	30	27	25	26	22	20
18	14	11	6	2	1	3	9	30	

Fig. 3 Sample for Solutions Representation for Character shown in Fig 2

Based on Fig 4, each solution represents the FCC using an edge sequence. The sequence is tested to see if it is connected continuously between the end vertex of the current edge and the start vertex of the following edge until all nodes have been visited. As an illustration, at the beginning of the sequence, edge 26 has a start vertex of 9 and an end vertex of 10, and it progresses to edge 22, where the start vertex is 10 and the end vertex is 8, and so on. Here, vertex 10 functions as a bind between two edges, allowing a smooth travel between those edges, hence the edge pair is regarded continuously connected.

3) HBA Approach to Minimize FCC

The purpose of HBA is to minimize the objective function of the solution representation. When calculating the objective function of a solution representation, the solution is repaired to meet the feasibility condition, which is whether the edge sequence is continuously connected until all nodes have been visited. To accomplish continuous connection, the edge that is not connected to the prior edge is swapped with the next closest viable edge in the sequence.

The objective function describes the quality of an FCC solution. In this study, the objective function is specified as the number of nodes that the FCC must visit from the starting node until all nodes have been reached. The objective function is calculated by counting the number of nodes visited (including revisits) from the vertex of the first edge to the end vertex of the final edge, where all nodes have been visited. It should be mentioned that while walking from one edge to the next, a repair process is attempted in order to meet the feasibility requirements. If the repair procedure fails, it will be assigned an objective function value equal to (2 * the number of nodes) which acts as a penalty to the solution.

$$ObjectiveFunction = 2 * NumberofNodes \tag{1}$$

For the solution representation in Fig 4, the process walk only requires the following sequence: (26 22 20 18 14 11 6 2 1 3 9 30 24) because all of the nodes have previously been visited using that sequence. Equation 2 shows the equation for the objective function

$$ObjectiveFunction = TotalNodeCount + EdgeSequenceLength \tag{2}$$

After all, this sequence's objective function is (2 + 0 + 23 + 15 + 0 + 0 + 0 + 5 + 5 + 0 + 10 + 11 + 0) + 13 = 84.

In summary, the proposed HB-FCC extraction algorithm represents a chain code solution in the collection of edges. Edge is a representation of the answer. An edge starts and ends at similar nodes. Two distinct edges may originate from the same node or one edge may end in another of the same nodes. Due to the frequent need to revisit previously visited nodes, an edge may be visited twice, allowing the solutions representations to have the entire tours. The entire number of nodes the chain code needs to visit (including revisits), beginning from the starting nodes, was defined as an objective function. Fig. 5 shows the proposed pseudocode of HB-FCC feature extraction.

The Pseudocode of HBA-FCC Extraction Algorithm
Input data and settings parameter values
Clean image
Enumerate junction/end as nodes and interconnecting routes
Generate random node sequences.
Start HBA search with objective function()=path length
Initialize the population of honey badger with random position
Define intensity (*I*)
Evaluate the objective function of each honey badger position
Save the best position
Calculate the intensity (*I*)
Update position
Evaluate new position
Until stopping criterion is archive

Fig. 5 Proposed HBA-FCC Feature Extraction Algorithm

III. Results and Discussion

This section presents the results and discussion. HBA is the key method employed to compose the continuous FCC that covers all the nodes of the image, a feature of the handwritten character image. The proposed HB-FCC feature extraction involved ten replications where each replication consisted of 100 FCC solutions for every TBI. In evaluation of the proposed HB-FCC, two performance measurements were used using: (a) String length of chain code (directional code count in chain code) and (b) Computation time (time taken to solve the extraction of whole handwritten character image). Measurement of string length of extracted chain code from proposed HB-FCC was measured by recording the chain code length produced by HB-FCC in table form for plot on graph. Analysis has been done manually by observing the output string length trend from the graph. Computation time taken to extract the chain code was plotted into a single graph.

Initially, the process of producing the output was by computing the chain code of every character and then sorting the length of the string with the best, average, and worst with the total computation time. Subsequently, Table 2 and Table 3 indicate the output produced for route length and computation time, respectively, for each character by HB-FCC. Meanwhile, the result of all characters (A-Z) best, average, and worst in terms of route length and computation time are presented in Table 4.

Table 2 shows the detailed result of route lengths for different characters measured over ten runs, with corresponding averages. The most obvious observation from the table is that the average route lengths between the characters seem to vary a great deal. For instance, the average route length for Character A comes out with a longest length at 136.04 units and the shortest is 42.36 units for Character L. The high range in this data reveals that the underlying conditions or factors affecting these pathways are significantly different for each character.

TABLE II

GENERATED ROUTE LENGTH BY HB-FCC

CHAR	ROUTE LENGTH										AVG
	1	2	3	4	5	6	7	8	9	10	
A	136.40	150.60	134.80	137.60	137.60	133.80	130.40	141.80	130.80	126.60	136.04
B	103.20	105.00	97.80	100.20	102.20	99.40	98.60	100.20	93.80	98.20	99.86
C	63.00	59.80	61.20	62.00	60.20	62.80	63.40	62.60	66.40	60.00	62.14
D	67.20	67.20	67.20	67.20	67.20	167.2	67.20	67.20	67.20	67.20	67.20
E	81.20	84.40	85.40	84.80	86.80	78.20	84.80	83.80	91.80	86.20	84.74
F	64.60	63.20	61.80	66.20	61.80	63.20	64.80	67.20	63.00	64.00	63.98
G	74.60	73.80	83.60	72.40	76.40	79.80	73.40	75.00	79.00	68.40	75.64
H	123.40	128.00	106.40	126.80	118.20	118.00	115.40	129.40	118.00	116.80	120.04
I	55.75	53.50	56.50	58.75	55.25	59.25	56.00	53.00	59.75	57.25	56.50
J	60.80	63.00	61.60	63.60	61.40	63.00	61.60	61.20	61.20	62.60	62.00
K	90.00	86.60	83.40	91.00	100.4	92.80	95.00	92.20	83.20	94.00	90.86
L	45.00	42.80	43.60	46.00	40.80	40.80	41.40	41.60	41.20	40.40	42.36
M	84.00	79.40	74.00	80.20	80.00	80.60	81.40	78.00	86.20	76.80	80.06
N	49.60	49.20	49.00	50.00	48.80	49.20	48.40	49.20	48.80	49.40	49.16
O	64.20	64.00	64.20	64.00	64.00	64.00	64.00	64.00	64.00	64.00	64.04
P	112.50	111.50	112.25	117.00	112.50	112.75	101.50	113.00	106.75	124.75	112.45
Q	113.33	114.33	107.33	99.67	115.67	114.00	108.00	116.00	109.33	110.00	110.77
R	74.60	78.60	75.00	77.80	75.00	74.80	75.20	78.00	77.00	76.20	76.22
S	51.20	51.20	51.20	51.20	51.20	51.20	51.20	51.20	51.20	51.20	51.20
T	51.60	51.60	51.60	51.60	51.60	51.60	51.60	51.60	51.60	51.60	51.60
U	59.80	59.80	59.80	59.80	59.80	59.80	59.80	59.80	59.80	59.80	59.80
V	48.60	48.60	48.60	48.60	48.60	48.60	48.60	48.60	48.60	48.60	48.60
W	74.20	74.20	77.60	73.20	71.20	81.60	69.40	74.40	79.80	70.20	74.58
X	82.60	71.40	67.80	83.20	71.20	83.60	79.80	69.20	71.80	70.40	75.10
Y	46.40	44.60	44.40	44.80	45.20	44.80	45.60	45.80	45.20	45.80	45.26
Z	54.00	57.80	54.20	53.40	55.60	57.80	56.80	55.40	56.60	55.20	55.68

However, characters D, S, T, U, and V generally return highly consistent values for route lengths across all measurements. For instance, Characters S, T, and U take on the same value in each run, which implies a constant environment or a very deterministic process. However, the sixth measurement of Character D provides for an outlier value in route length of 167.20, whereas all other values are 67.20. Such a marked discrepancy throws suspicion on data entry error or on some unusual condition having occurred at that time. Characters like A, G, P, Q and K, in contrast, show greater consistency over the ten measurements but more variation in their route lengths. That means the high variability could be caused by the change of operational conditions, changes in environmental factors, or inherent randomness in the processes governing these routes. The causes of such variation might need to be elicited through further testing, and such routes might be stabilized if needed.

TABLE III

COMPUTATIONAL TIME BY HB-FCC

CHAR	TOTAL TIME(S)										AVG
	1	2	3	4	5	6	7	8	9	10	
A	0.09	0.07	0.07	0.07	0.07	0.08	0.07	0.07	0.07	0.08	0.08
B	0.05	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.05	0.05	0.04
C	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02
D	0.03	0.02	0.03	0.03	0.03	0.03	0.03	0.02	0.02	0.03	0.03
E	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05
F	0.05	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04
G	0.04	0.04	0.03	0.04	0.03	0.04	0.04	0.03	0.04	0.04	0.04
H	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05
I	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04
J	0.03	0.03	0.04	0.04	0.03	0.03	0.03	0.04	0.04	0.04	0.04
K	0.06	0.05	0.06	0.06	0.06	0.06	0.05	0.05	0.05	0.06	0.06
L	0.03	0.03	0.03	0.02	0.03	0.03	0.03	0.03	0.03	0.03	0.03
M	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05
N	0.03	0.03	0.04	0.03	0.03	0.03	0.03	0.04	0.03	0.05	0.03
O	0.02	0.02	0.03	0.02	0.02	0.02	0.02	0.02	0.03	0.02	0.03
P	0.08	0.08	0.08	0.08	0.08	0.08	0.08	0.07	0.08	0.07	0.08
Q	0.11	0.08	0.08	0.08	0.08	0.08	0.07	0.08	0.08	0.08	0.08
R	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04
S	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02
T	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.05	0.04	0.04
U	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02
V	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.02	0.01
W	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05
X	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.05	0.06
Y	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04
Z	0.06	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.03	0.04

In terms of computational time, as shown in Table 3, many characters, such as C, E, F, H, M, R, and W, have quite uniformly distributed computational times, with deviations of not more than 0.01 second. This invariability indicates that the processes of feature extraction of these characters by the HB-FCC are highly effective. On the other hand, some characters vary slightly in regard to their computational time. For instance, characters A, K, and Q present small deviations, and character Q has a big deviation time in operations to present an average of 0.08 seconds and it peaks sometimes at 0.11. This means that a character like Q is likely to have higher complexity in its operations, or it may be hit with rare computational overhead issues that increase processing.

TABLE IV
RESULT OF ALL CHARACTER (A-Z)

	Route length	Computation time (s)
Best	1880.28	1.07
Average	1915.88	1.10
Worst	1934.13	1.16

Summarily, according to the summarized result of all characters in Table 4, the result has pointed out to the fact that the relationship between the length of a route and computational time is clear. The shortest route length for the best-case scenario is expectedly 1880.28 units, while the corresponding smallest computational time is 1.07 s, on the other hand. The longest route length in the worst-case scenario is 1934.13 units, while its related highest computational time is 1.16 seconds. This correlation suggests that longer routes require more computational resources and lead to more processing time.

IV. CONCLUSION

This study introduces an alternative approach for the improvement of the feature extraction technique in HCR by proposing the Honey Badger Freeman Chain Code (HB-FCC) feature extraction algorithm. Conventional feature-extraction algorithms like the Freeman Chain Code are inherently noise-sensitive, starting-point dependent, and require substantial computation for complex character processing. In most cases, these become very challenging and yield poor accuracy and efficiency in HCR systems that handle real-world data, normally characterized by variability and distortions in handwritten text.

The recognition problems are eliminated in the algorithm proposed here. HB-FCC eradicates such challenges using the HBA as the optimization metaheuristic technique. The proposed method optimizes the selection of starting points and reduces noise-induced variations using HBA in the feature extraction of the chain code, thus leading to a more robust and reliable set of features for character recognition. Next, handwritten character images from binary images are taken, each transformed into a graph with nodes representing important key structural features, as follows: This reduces the length of the FCC with the most efficient sequence of edges identified by the HBA, giving general recognition accuracy.

A good improvement in both the computational efficiency and the recognition accuracy can be observed with the application of the HB-FCC algorithm, in comparison to the traditional methods of FCC. It results not only in the computational time needed for feature extraction but also in the representative features of the real character structures, hence achieving better recognition across different character sets. The relative decrease in route length and computational time over different characters gives evidence that this algorithm might have wider applicability in HCR systems.

In conclusion, the HB-FCC algorithm represents a significant advancement in the field of HCR. By addressing the shortcomings of traditional FCC methods and introducing a metaheuristic approach to feature extraction, this study lays the groundwork for the development of more efficient and accurate HCR systems. The proposed method's ability to handle noise and variability in handwritten text makes it particularly suitable for real-world applications. Future research could explore the hybridization of HB-FCC with other metaheuristic optimization algorithms as suggested in studies [28]-[30], potentially leading to further enhancements in HCR performance across diverse languages and scripts.

ACKNOWLEDGMENT

This research work is supported by Universiti Malaysia Pahang Grant: RDU230359. The authors honorably appreciate the support of the Soft Computing and Optimization Research Group (SCORE). The full article expresses the following acknowledgements: “Communication of this research is made possible through monetary assistance by Universiti Tun Hussein Onn Malaysia and the UTHM Publisher’s Office via Publication Fund E15216”.

REFERENCES

[1] H. A. Alhamad et al., "Handwritten recognition techniques: A comprehensive review," *Symmetry*, vol. 16, no. 6, 2024, doi:10.3390/sym16060681.

[2] K. Ding, Z. Liu, L. Jin, and X. Zhu, "A comparative study of Gabor feature and gradient feature for handwritten Chinese character recognition," in *Proc. Int. Conf. Wavelet Anal. Pattern Recognit.*, vol. 3, pp. 1182-1186, Nov. 2007, doi: 10.1109/icwapr.2007.4421612.

[3] S. Inunganbi, "A systematic review on handwritten document analysis and recognition," *Multimedia Tools Appl.*, vol. 83, pp. 5387-5413, 2024, doi: 10.1007/s11042-023-15326-9.

[4] R. Kaur, M. Uppal, and D. Gupta, "A comprehensive and comparative study of handwriting recognition system," in *Proc. IEEE Renewable Energy Sustainable E-Mobility Conf. (RESEM)*, 2023, pp. 1-6, doi:10.1109/resem57584.2023.10236301.

[5] F. Baji and M. Mocanu, "Chain code approach for shape based image retrieval," *Indian J. Sci. Technol.*, vol. 11, no. 3, pp. 1-17, 2018, doi:10.17485/ijst/2018/v11i3/119998.

[6] M. A. Mohamad, J. Sallim, and K. Moorthy, "Whale optimisation Freeman chain code (WO-FCC) extraction algorithm for handwritten character recognition," in *Proc. Int. Conf. Softw. Eng. Comput. Syst. 4th Int. Conf. Comput. Sci. Inf. Manage. (ICSECS-ICOCSIM)*, Aug. 2021, pp. 598-602, doi: 10.1109/ICSECS52883.2021.00115.

[7] R. Sheth, N. C. Chauhan, M. M. Goyani, and K. A. Mehta, "Handwritten character recognition system using chain code and correlation coefficient," *Int. J. Comput. Appl.*, 2011, pp. 31-36.

[8] M. A. Mohamad, H. Haron, and H. Hasan, "Metaheuristic optimization on conventional Freeman chain code extraction algorithm for handwritten character recognition," in *Intell. Inf. Database Syst.*, Springer, 2017, pp. 518-527, doi: 10.1007/978-3-319-54472-4_49.

[9] S. Singh, N. K. Garg, and M. Kumar, "Feature extraction and classification techniques for handwritten Devanagari text recognition: A survey," *Multimedia Tools Appl.*, vol. 82, pp. 747-775, 2023, doi:10.1007/s11042-022-13318-9.

[10] M. A. Mohamad et al., "A review on feature extraction and feature selection for handwritten character recognition," *Int. J. Adv. Comput.*

Sci. Appl., vol. 6, no. 2, pp. 230-237, 2015, doi:10.14569/ijacsa.2015.060230.

[11] K. Rajwar, K. Deep, and S. Das, "An exhaustive review of the metaheuristic algorithms for search and optimization: Taxonomy, applications, and open challenges," *Artif. Intell. Rev.*, vol. 56, no. 9, pp. 13187-13257, 2023, doi: 10.1007/s10462-023-10470-y.

[12] S. Yacoubi et al., "A metaheuristic perspective on extracting numeric association rules: Current works, applications, and recommendations," *Arch. Comput. Methods Eng.*, Mar. 2024, doi:10.1007/s11831-024-10109-3.

[13] F. A. Hashim et al., "Honey Badger Algorithm: New metaheuristic algorithm for solving optimization problems," *Math. Comput. Simul.*, vol. 192, pp. 84-110, 2022, doi: 10.1016/j.matcom.2021.08.013.

[14] O. O. Akinola et al., "Multiclass feature selection with metaheuristic optimization algorithms: A review," *Neural Comput. Appl.*, vol. 34, pp. 19751-19790, 2022, doi: 10.1007/s00521-022-07705-4.

[15] S. Jain, A. Jain, and M. Jangid, "Review of metaheuristic techniques for feature selection," in *Soft Comput.: Theories Appl.*, R. Kumar et al., Eds. Springer, 2023, pp. 397-410, doi: 10.1007/978-981-19-9858-4_33.

[16] A. T. Sahlol et al., "Handwritten Arabic optical character recognition approach based on hybrid whale optimization algorithm with neighborhood rough set," *IEEE Access*, vol. 8, pp. 23011-23021, 2020, doi: 10.1109/access.2020.2970438.

[17] R. Tokas and A. Bhadu, "A comparative analysis of feature extraction techniques for handwritten character recognition," *Int. J. Adv. Technol. Eng. Res.*, vol. 2, no. 4, pp. 215-219, Jul. 2012.

[18] S. S. Kareem et al., "An effective feature selection model using hybrid metaheuristic algorithms for IoT intrusion detection," *Sensors*, vol. 22, no. 4, Feb. 2022, doi: 10.3390/s22041396.

[19] D. S. Joshi and Y. R. Risodkar, "Deep learning based Gujarati handwritten character recognition," in *Proc. 2018 Int. Conf. Adv. Commun. Comput. Technol. (ICACCT)*, Sangamner, India, Feb. 2018, pp. 563–566.

[20] M. Kef, L. Chergui, and S. Chikhi, "A novel fuzzy approach for handwritten Arabic character recognition," *Pattern Anal. Appl.*, vol. 19, no. 4, pp. 1041–1056, Jul. 2015, doi: 10.1007/s10044-015-0500-4.

[21] S. Alghyaline, "Arabic optical character recognition: A review," *Comput. Model. Eng. Sci.*, vol. 135, no. 3, pp. 1825-1861, 2023, doi: 10.32604/cmes.2022.024555.

[22] X. Y. Zhang, Y. C. Wu, F. Yin, and C. L. Liu, "Deep learning based handwritten Chinese character and text recognition," in *Deep Learning: Fundamentals, Theory and Applications*, K. Huang, A. Hussain, Q. F. Wang, and R. Zhang, Eds., vol. 2, Cham, Switzerland: Springer, 2019, pp. 45–72, doi: 10.1007/978-3-030-06073-2_3.

[23] Y. Sun and H. Wang, "Similar Chinese character recognition algorithm based on glyphs and strokes information," in *Proc. 2nd Int. Conf. Algorithm, Image Process. Mach. Vis. (AIPMV)*, Jul. 2024, pp. 58-65, doi: 10.1109/AIPMV62663.2024.10692045.

[24] J. Cui et al., "An improved multi-objective honey badger algorithm based on global searching strategy," *J. Supercomput.*, vol. 81, no. 5, Apr. 2025, doi: 10.1007/s11227-025-07177-y.

[25] Z. Ye et al., "An improved honey badger algorithm through fusing multi-strategies," *Comput. Mater. Contin.*, vol. 76, no. 2, pp. 1479-1495, 2023, doi: 10.32604/cmc.2023.038787.

[26] S.-W. Zhang et al., "Improved honey badger algorithm based on elementary function density factors and mathematical spirals in polar coordinate system," *Artif. Intell. Rev.*, vol. 57, 2024, doi:10.1007/s10462-023-10658-2.

[27] S. N. Qasem, "A novel honey badger algorithm with multilayer perceptron for predicting COVID-19 time series data," *J. Supercomput.*, vol. 80, pp. 3943-3969, 2024, doi: 10.1007/s11227-023-05560-1.

[28] M. BinJubier et al., "Optimizing genetic algorithm by implementation of an enhanced selection operator," *JOIV: Int. J. Informatics Vis.*, vol. 8, no. 3-2, pp. 1643-1650, 2024, doi: 10.62527/joiv.8.3-2.3449.

[29] N. S. Nordin and M. A. Ismail, "A hybridization of butterfly optimization algorithm and harmony search for fuzzy modelling in phishing attack detection," *Neural Comput. Appl.*, vol. 35, no. 7, pp. 5501-5512, 2023, doi: 10.1007/s00521-022-07957-0.

[30] N. F. Idris and M. A. Ismail, "A review of homogenous ensemble methods on the classification of breast cancer data," *Przegląd Elektrotechniczny*, vol. 2024, no. 1, 2024, doi:10.15199/48.2024.01.21.