Advances in Intelligent Systems and Computing 1392

Aruna Tiwari · Kapil Ahuja ·
Anupam Yadav ·
Jagdish Chand Bansal · Kusum Deep ·
Atulya K. Nagar *Editors*

Soft Computing for Problem Solving

Proceedings of SocProS 2020, Volume 1

Springer

Advances in Intelligent Systems and Computing

Volume 1392

The series “Advances in Intelligent Systems and Computing” contains publications on theory, applications, and design methods of Intelligent Systems and Intelligent Computing. Virtually all disciplines such as engineering, natural sciences, computer and information science, ICT, economics, business, e-commerce, environment, healthcare, life science are covered. The list of topics spans all the areas of modern intelligent systems and computing such as: computational intelligence, soft computing including neural networks, fuzzy systems, evolutionary computing and the fusion of these paradigms, social intelligence, ambient intelligence, computational neuroscience, artificial life, virtual worlds and society, cognitive science and systems, Perception and Vision, DNA and immune based systems, self-organizing and adaptive systems, e-Learning and teaching, human-centered and human-centric computing, recommender systems, intelligent control, robotics and mechatronics including human-machine teaming, knowledge-based paradigms, learning paradigms, machine ethics, intelligent data analysis, knowledge management, intelligent agents, intelligent decision making and support, intelligent network security, trust management, interactive entertainment, Web intelligence and multimedia.

The publications within “Advances in Intelligent Systems and Computing“ are primarily proceedings of important conferences, symposia and congresses. They cover significant recent developments in the field, both of a foundational and applicable character. An important characteristic feature of the series is the short publication time and world-wide distribution. This permits a rapid and broad dissemination of research results.

Indexed by DBLP, INSPEC, WTI Frankfurt eG, zbMATH, Japanese Science and Technology Agency (JST).

All books published in the series are submitted for consideration in Web of Science.

More information about this series at http://www.springer.com/series/11156

Aruna Tiwari · Kapil Ahuja · Anupam Yadav ·
Jagdish Chand Bansal · Kusum Deep ·
Atulya K. Nagar
Editors

Soft Computing for Problem Solving

Proceedings of SocProS 2020, Volume 1

Springer

Editors
Aruna Tiwari
Computer Science and Engineering
Indian Institute of Technology Indore
Indore, Madhya Pradesh, India

Kapil Ahuja
Computer Science and Engineering
Indian Institute of Technology Indore
Indore, Madhya Pradesh, India

Anupam Yadav
Dr. B. R. Ambedkar National Institute of Technology
Jalandhar, India

Jagdish Chand Bansal
Department of Mathematics
South Asian University
New Delhi, India

Kusum Deep
Department of Mathematics
Indian Institute of Technology Roorkee
Roorkee, India

Atulya K. Nagar
School of Mathematics
Computer Science and Engineering
Liverpool Hope University
Liverpool, UK

ISSN 2194-5357 ISSN 2194-5365 (electronic)
Advances in Intelligent Systems and Computing
ISBN 978-981-16-2708-8 ISBN 978-981-16-2709-5 (eBook)
https://doi.org/10.1007/978-981-16-2709-5

This Springer imprint is published by the registered company Springer Nature Singapore Pte Ltd.
The registered company address is: 152 Beach Road, #21-01/04 Gateway East, Singapore 189721, Singapore

Preface

IIT Indore and the Soft Computing Research Society (SCRS), New Delhi, co-hosted the "10th International Conference on Soft Computing for Problem Solving (SocPros 2020)" from 18 December to 20 December 2020 in a Virtual Mode. The seeds for this conference were laid more than a year ago at the 9th conference in this series at the Liverpool Hope University, UK (in September 2019).

The conference opening was done by Prof. Deepak B. Phatak (BoG Chairman, IIT Indore), Prof. Neelesh K. Jain (Director, IIT Indore), and Prof. Ajit K. Chaturvedi (Director, IIT Roorkee and IIT Mandi). These esteemed guests appreciated the efforts and highlighted the need of taking the technology to the common people.

This mega event, which happens to be the first international conference of Computer Science and Engineering at IIT Indore, covered recent developments in the interdisciplinary areas of Artificial Intelligence, Machine Learning, Optimization, and Soft Computing. The conference received 334 papers from participants belonging to 13 different countries, which went through a very stringent blind review process. This was done by the international expert committee and had a very good acceptance rate of 37%. These papers would be published as two books by Springer.

This year the conference had many innovative features. Prof. Chandra Mohan Gold Medal for excellence in Soft Computing was instituted, which was given to Prof. Sankar Pal of ISI Kolkatta. Twelve eminent academicians gave keynote talks. There was a big industry participation with four keynote talks by distinguished industrialists. Fourteen outstanding paper awards and 5 best paper awards (sponsored by Springer) were given. The conference had two special sessions on "Cognitive Science" and "Remote Sensing" as well as a panel discussion on the future of soft computing with ten renowned panelists from academia and industry.

Indore, India	Aruna Tiwari
Indore, India	Kapil Ahuja
Jalandhar, India	Anupam Yadav
New Delhi, India	Jagdish Chand Bansal
Roorkee, India	Kusum Deep
Liverpool, UK	Atulya K. Nagar

Contents

About the Editors

Dr. Aruna Tiwari is Associate Professor in Computer Science & Engineering at IIT Indore, India. Her all degrees B.Tech., M.Tech., Ph.D. are in Computer Science & Engineering from SGSITS Indore. She has more than 20 years of research experience in the areas such as artificial intelligence, machine learning, and soft-computing. Her work is around artificial neural networks, fuzzy clustering, evolutionary computation and their applications to bioinformatics, and medical diagnosis. Her research directions are toward the design of intelligent computing algorithms for classification, clustering, and feature selection which can handle big data. She has more than 30 journal publications of international repute including more than 5 IEEE transactions and Elsevier journals of high impact factor and has published more than fifty research articles in the ranked conferences. She has established a big data handling lab in IIT Indore which is funded by Council of Science & Industrial Research (CSIR) Government of India in 2017. She has enabled two MOUs, collaborating with Indian Institute of Soybean Research, Indore, and CSIR-Central Electronics Engineering Research Institute, Pilani. She has recently initiated a consortium project on Artificial Intelligence which is approved from Ministry of Electronics & Information Technology (MeitY) Government of India. There are seven PhDs and more than 25 masters awarded under her guidance.

Dr. Kapil Ahuja holds master's and Ph.D. degrees in Mathematics and Computer Science (from Virginia Tech, USA) and has a strong interdisciplinary focus. After graduating earlier this decade from VT, he received his postdoctoral training from the Max Planck Institute in Magdeburg (Germany). Since then, he has established his independent research program in Mathematics of Data Science and Computational Science at IIT Indore, where he is currently working as Associate Professor. Dr. Ahuja is solving challenging problems that are at both the ends of the research spectrum, i.e., theoretical as well as applicable. His core research interests are in artificial intelligence, machine learning, numerical methods, and optimization. He believes that it is necessary to collaborate globally to solve challenging research problems. Hence, he has multiple active international research collaborations (USA, Germany, India, France, and UK). In the recent past, he has also held visiting professor positions at TU Braunschweig (Germany), TU Dresden (Germany), and Sandia National Labs

(USA). Dr. Ahuja's overall research output includes thirty publications (including eighteen in reputed journals) as well as external funding worth more than half-a-million USD from twelve projects. While achieving this, he has graduated two Ph.D. students with three more to graduate soon. Since teaching and service are essential for a fruitful and satisfying research career, he is very committed to these aspects as well. Dr. Ahuja has received the Best Teacher Award four times at IIT Indore. In the past, he has held many administrative positions. Since the past three years, he is heading International Affairs at IIT Indore as Founding Dean for the same.

Dr. Anupam Yadav is Assistant Professor in the Department of Mathematics at Dr. BR Ambedkar National Institute of Technology Jalandhar, India. His research area includes numerical optimization, soft computing, and artificial intelligence, he has more than ten years of research experience in the areas of soft computing and optimization. Dr. Yadav has done Ph.D. in soft computing from Indian Institute of Technology Roorkee, and he worked as Research Professor at Korea University. He has published more than twenty-five research articles in journals of international repute and has published more than fifteen research articles in conference proceedings. Dr. Yadav has authored a text book entitled "An introduction to neural network methods for differential equations". He has edited three books which are published by AISC, Springer Series. Dr. Yadav was General Chair, Convener, and Member of steering committee of several international conferences. He is Associate Editor in the journal of the experimental and theoretical artificial intelligence. Dr. Yadav is Member of various research societies.

Dr. Jagdish Chand Bansal is Associate Professor at South Asian University, New Delhi, and Visiting Faculty at Maths and Computer Science, Liverpool Hope University, UK. Dr. Bansal has obtained his Ph.D. in Mathematics from IIT Roorkee. Before joining SAU, New Delhi, he has worked as Assistant Professor at ABV Indian Institute of Information Technology and Management Gwalior and BITS Pilani. His primary area of interest is swarm intelligence and nature-inspired optimization techniques. Recently, he proposed a fission–fusion social structure-based optimization algorithm, spider monkey optimization (SMO), which is being applied to various problems from engineering domain. He has published more than 60 research papers in various international journals/conferences. He is Series Editor of the book series such as Algorithms for Intelligent Systems (AIS) and Studies in Autonomic, Data-driven and Industrial Computing (SADIC) published by Springer. He is Editor-in-Chief of International Journal of Swarm Intelligence (IJSI) published by Inderscience. He is also Associate Editor of IEEE ACESSS published by IEEE and ARRAY published by Elsevier. He is Steering Committee Member and General Chair of the annual conference series SocProS. He is General Secretary of Soft Computing Research Society (SCRS). He has also received Gold Medal at UG and PG level.

Prof. Kusum Deep is Full Professor in the Department of Mathematics, Indian Institute of Technology Roorkee, India, and Visiting Professor, Liverpool Hope University, UK, and University of Technology Sydney, Australia. She has completed her B.Sc. (Hons) and M.Sc. (Hons) from Centre for Advanced Studies, Panjab University, Chandigarh. She is an M.Phil. Gold Medalist. She earned her Ph.D. from UoR (now IIT Roorkee) in 1988. She has been a national scholarship holder and a postdoctoral from Loughborough University, UK assisted by International Bursary funded by Commission of European Communities, Brussels. She has won numerous awards like Khosla Research Award, UGC Career Award Starred Performer of IITR Faculty, many best paper awards by Railway Bulletin of Indian Railways, special facilitation in memory of late Professor M. C. Puri, and AIAP Excellence Award. She has authored two books, supervised 20 Ph.Ds, and published 125 research papers. She is Senior Member of IEEE, ORSI, CSI, IMS, and ISIM. She is Executive Editor of International Journal of Swarm Intelligence, Inderscience. She is Associate Editor of Swarm and Evolutionary Algorithms, Elsevier. She is on the editorial board of many journals. She is Founder President of Soft Computing Research Society, India. She is General Chair of series of International Conference on Soft Computing for Problems Solving (SocProS). Her research interests are evolutionary algorithms, swarm Iitelligence, and nature-inspired optimization techniques and their applications.

Prof. Atulya K. Nagar holds Foundation Chair as Professor of Mathematical Sciences and is Pro-Vice-Chancellor for Research at Liverpool Hope University, United Kingdom. He has been Dean of the Faculty of Science and Head of the School of Mathematics, Computer Science and Engineering which he established at the University. He received a prestigious Commonwealth Fellowship for pursuing his doctorate (D.Phil.) in Applied Nonlinear Mathematics, which he earned from the University of York (UK) in 1996. He holds B.Sc. (Hons), M.Sc. (Mathematics), and M.Phil. (with distinction) in Mathematical Physics from the MDS University of Ajmer, India. Prior to joining Liverpool Hope, he was at the Department of Mathematical Sciences and later at the Department of Systems Engineering, at Brunel University, London. He is internationally respected Scholar working at the cutting edge of theoretical computer science, applied mathematical analysis, and systems engineering with his research expertise spanning both applied mathematics and computational methods for nonlinear, complex, and intractable problems arising in science, engineering, and industry. He has edited volumes on Intelligent System, and Applied Mathematics. He is Editor-in-Chief of the International Journal of Artificial Intelligence and Soft Computing (IJAISC) and serves on editorial boards for a number of prestigious journals. He is well published with over 450 publications in prestigious publishing outlets. Professor Nagar sits on a number of strategic UK wide research bodies including the JISC Research Strategy group, and he is Fellow of the Institute of Mathematics and its Applications (FIMA) and Fellow of the Higher Education Academy (FHEA).

Swarm Programming Using Multi-verse Optimizer

Tapas Si

Abstract Swarm programming is the swarm-based automatic programming in which swarm intelligence algorithms are used to evolve computer programs automatically. Automatic programming is a division of machine learning where machines learn how to write the program for themselves. Grammar-based swarm programming is an interesting research topic in recent times. In grammar-based swarm programming, context-free grammar (CFG) is utilized in the generation of computer programs automatically in any arbitrary target computer language. In this paper, grammatical multi-verse optimizer (GMVO) is proposed to generate computer programs automatically. The proposed method is applied to three benchmark problems such as Santa Fe Ant Trail (SFAT), 3-multiplexer, and symbolic regression. The experimental results of the proposed GMVO are compared with that of grammatical fireworks algorithm (GFWA), grammatical bee colony (GBC), and grammatical swarm (GS). The empirical results with analysis demonstrate that the GMVO can be used in automatic generation of computer programs in any arbitrary target computer language.

1 Introduction

Automatic programming (AP) [1, 2] is a notable research area of machine learning (ML) to generate the computer programs automatically in any arbitrary target language. AP helps the computer to learn how to write programs by itself. Genetic programming (GP), developed by Koza in 1992, is a very popular automatic programming algorithm [2]. In GP, each individual is linear or nonlinear genomes, i.e. parse tree, and each individual represents a computer program. But GP suffers from the following two problems: (i) code bloat and (ii) invalid parse tree. In code bloat, the parse tree may grow indefinitely due to iterative use of crossover and mutation operation whether they lead to improve the solutions or not, it causes to reduce the

T. Si (✉)
Department of Computer Science and Engineering, Bankura Unnayani Institute of Engineering, Bankura, West Bengal, India
e-mail: c2.tapas@gmail.com

A. Tiwari et al. (eds.), *Soft Computing for Problem Solving*, Advances in Intelligent Systems and Computing 1392, https://doi.org/10.1007/978-981-16-2709-5_1

efficiency of the genetic operators and exhaustion of memory. A variant of GP, grammatical evolution (GE) [3, 4] was proposed by M. O'Neill and C. Ryan in 2001. In GE, elements of linear genomes are the integer values termed as codons in the range [0, 255]. The evolved computer programs are the *phenotype* generated from the *genotype* using CFG in Backus–Naur form (BNF). The mapping process in GE has an advantage over GP for reducing the bloat and increasing the possibilities to generate valid program. Swarm-based automatic programming [5] termed as swarm programming (SP) has now become an emerging research topic. Swarm programming is the swarm-based automatic programming in which SI algorithms are used in the automatic evolution of computer programs. The ant programming is developed by O. Roux and C. Fonlupt [6] in the year 2000, and its learning algorithm was ant colony optimizer (ACO) developed by M. Dorigo [7] which was applied to evolve programs. Ant programming uses a tree structure to represent the computer program like GP. Artificial bee colony programming (ABCP) is developed by D. Karaboga et al. [8] to solve symbolic regression problems. In ABCP, tree-structured genomes represent computer programs like GP. Artificial bee colony (ABC) [9] is SI algorithm developed by modelling the foraging behaviour of honey bees in the nature. Apart from the ant programming and ABCP algorithms, M. O'Neill et al. [10, 11] first introduced grammar-based swarm programming known as grammatical swarm (GS). In GS, particle swarm optimizer (PSO) [13] algorithm which mimics the bird-flockings in nature is used as a learning algorithm. M. O'Neill et al. [12] also made further development of GS by using variable-length PSO as search engine. Grammatical bee colony (GBC) was developed by T. Si et al. [14], and GBC used artificial bee colony (ABC) as a learning algorithm. Grammatical fireworks algorithm (GFWA) was introduced by T. Si [15], and its learning algorithm was fireworks algorithm (FWA) [16] modelling the process of fireworks explosion at night in the sky. In study [17], T. Si developed grammatical moth-flame optimizer (GMFO) to generate computer programs automatically. In GMFO, moth-flame optimizer (MFO) [18] is used as a learning algorithm, and it is inspired by the navigation called as the transverse orientation of moths in nature. In the same study, grammatical whale optimizer (GWO) is also developed, and its learning algorithm is whale optimization algorithm (WOA) [19] which mimics the hunting behaviour of humpback whales in nature. Like GS, GBC, and GFWA, both GMFO and GWO use linear array that consists of integer codons from which programs are evolved. It is also found from the same study that GMFO performs better than GWO in automatic program generation. In GS, GBC, GFWA, GMFO, and GWO, the individual search agent is a linear array of integer codons, and CFG is used in the *genotype-to-phenotype* mapping process, whereas ant programming and ABCP use tree data structure to represent the computer programs. A. Mahanipour and H. Nezamabadi-pour [20] developed gravitational search programming (GSP) which used the gravitational search algorithm (GSA) [21] as the search engine. GSP was applied to solve the symbolic regression and feature construction problems. It performed better than GP and ABCP in the aforementioned problems. Headleand and Teahan proposed grammatical herding which modeled the herd behaviour of horses and used linear array of binary strings from which GE type mapping process was used to generate programs [26]. Its performance was analysed only on SFAT problem. Liu et al. [24] applied fish swarm algorithm (FSA) to generate programs for solving symbolic regression. In this algorithm, parse tree that represents a program encoded into a linear string having fixed size using gene expression scheme in gene expression programming (GEP). Husselmann and Hawick developed geometric firefly algorithm (GFA) where the firefly algorithm (FFA) was used to construct the gene expression in GEP [25]. J. Togelius et al. [22] developed particle swarm programming (PSP) in which geometric PSO [23] and GP were combined to evolve the expression trees as computer programs. Apart from the development of different SP algorithms, they are also applied in different areas such as protein classification [27], medical data classification [28], brain MRI segmentation [29], optimal control [5], pattern recognition [30], and artificial neural network (ANN) training [31]. In this current work, grammatical multi-verse optimizer (GMVO) is presented for the automatic computer program generation. To validate the efficiency and effectiveness of GMVO, it is applied to three widely used benchmark problems such as SFAT, 3-input multiplexer, and symbolic regression. All these benchmark problems are collected from GP literature. A comparative study is conducted with GFWA and GBC. The empirical results demonstrate that GMVO outperforms other algorithms.

The remaining of this paper is organized as follows: grammatical multi-verse optimizer is discussed in Sect. 2. The experimental set-up is given in Sect. 4. Results and discussion are given in Sect. 5. Finally, a conclusion is given in Sect. 6.

2 Grammatical Multi-verse Optimizer

The GMVO is a swarm-based automatic programming to generate computer programs automatically. In GMVO, recently developed multi-verse optimizer (MVO) [32] is applied as a learning algorithm (i.e. search engine) in evolving computer programs through *genotype-to-phenotype* mapping. The MVO is an SI algorithm based on multi-verse theory. Each candidate solution in MVO represents a universe in the space. There are three main components of the universe such as white holes, black holes, and wormholes. The movements of the objects are allowed from one universe to another through white/black hole tunnels. Generally, the universe having the higher inflation rate is considered to have white holes, whereas the universe having the lower inflation rate is considered to have the black holes. The exploration of the search process is maintained by the movement of the objects through the white/black hole tunnels, whereas the exploitation is maintained by the object movements from the universes towards the best universe through wormholes. Let assume $\mathscr{X}_i = (x_{i1}, x_{i2}, \ldots, x_{i\mathscr{D}})$ is the ith universe in the space, and each component x_{ij} represents the jth object of the ith universe where $\mathscr{D}$ is the dimension of the problem to be solved. The inflation rate (I) of the ith universe is represented by the fitness, i.e. the objective function value $f(\mathscr{X}_i)$. For object movement, first the kth universe $\mathscr{X}_k$ has been selected using roulette wheel selection strategy, and then, to move the j objects of the kth universe to the ith universe, the following equation is used:

$$x_{ij} = \begin{cases} x_{kj} & \text{if} \quad r_1 < NI(\mathscr{X}_i) \\ x_{ij} & \text{if} \quad r_1 \geq NI(\mathscr{X}_i) \end{cases} \tag{1}$$

where r_1 is the uniformly distributed random number in (0, 1), and $NI(\mathscr{X}_i)$ is the normalize inflation rate of the ith universe. In order to make local changes for each universe through wormholes for exploitation or local search, the following equation is used:

$$x_{ij} = \begin{cases} \begin{cases} x_{ij} + TDR \times (\mathscr{X}_{\min} + (\mathscr{X}_{\max} - \mathscr{X}_{\min}) \times r_4) & r_3 < 0.5 \\ x_{ij} - TDR \times (\mathscr{X}_{\min} + (\mathscr{X}_{\max} - \mathscr{X}_{\min}) \times r_4) & r_3 \geq 0.5 \end{cases} & r_2 < WEP \\ x_{ij} & r_2 \geq WEP \end{cases} \tag{2}$$

where r_2, r_3, r_4 are uniformly distributed random number in (0, 1). TDR is the travelling distance rate, and WEP is the wormhole existence probability. TDR is defined as

$$TDR = 1 - \frac{t^{\frac{1}{p}}}{T^{\frac{1}{p}}} \tag{3}$$

where t is the current iteration, T is the maximum iterations, and p is the exploitation accuracy over iterations. The higher p makes the MVO to have faster and more accurate exploitation during the search. WEP is defined as

$$WEP = WEP_{\min} + (WEP_{\max} - WEP_{\min}) \times \frac{t}{T} \tag{4}$$

The flow chart of MVO is given in Fig. 1.

3 Genotype-to-Phenotype Mapping

In GMVO, each universe represents the genome and consists of integer codons in [0, 255]. A genotype representation is given in Fig. 2.

The BNF of CFG is utilized in the mapping process of phenotype from genotype. An example of BNF of CFG is given in Fig. 3:
A mapping process is to derive the choice number of a non-terminal in the string by the following formula:
rule=(codon value) MOD (number of choices for the current non-terminal).

A derivation from the codons in Fig. 2 is depicted in Fig. 4.

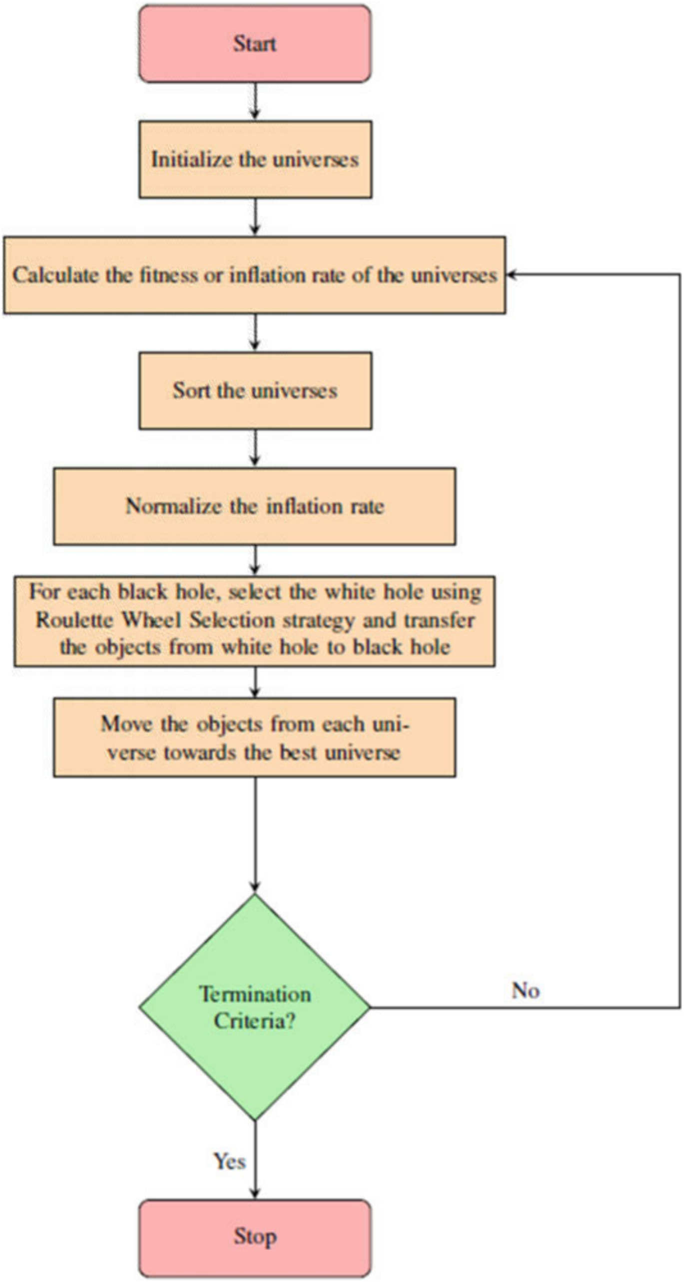

Fig. 1 Flow chart of MVO algorithm

156	63	171	101	223	204	...

Fig. 2 Genotypic representation

```
1. <expr> := (<expr><op><expr>) (0)
                | <var> (1)
2. <op> := + (0)
          | - (1)
          | * (2)
          | / (3)
3. <var> := x1 (0)
          | x2 (1)
```

Fig. 3 BNF of CFG

```
<expr> :=(<expr><op><expr>)  (156 mod 2)=0
       :=(<var><op><expr>)   ( 63 mod 2)=1
       :=(x2<op><expr>)      (171 mod 2)=1
       :=(x2-<expr>)         (101 mod 4)=1
       :=(x2-<var>)          (223 mod 2)=1
       :=(x2-x1)             (204 mod 2)=0
```

Fig. 4 Genotype-to-phenotype mapping

4 Experimental Set-up

4.1 Benchmark Problems and Grammar

The proposed GMVO is applied to SFAT, symbolic regression, and 3-input multiplexer problems [10]. In the case of SFAT problem, the evolved programming code in inducted in a loop with the following termination criteria: the ant takes total 600 time steps, or all 89 food pieces in the trail are eaten by the ant. The problem-specific grammars for BNF of CFG for SFAT, regression, and multiplexer are given in Figs. 5, 6 and 7.

```
1. <code> := (<code><line>) | <line>
2. <line> := <condition> | <op>
3. <condition> := if(foodahead()){<line>}
                  else{<line>}
4. <op> := left(); | right(); | move();
```

Fig. 5 Grammar for SFAT

```
1. <expr> := (<expr><op><expr>) | <var>
2. <op> := + | - | * | /
3. <var> := x
'/' is protected division.
```

Fig. 6 Grammar for symbolic regression

```
1. <expr> := (<expr><op><expr>) | <var>
2. <op> := or | and | nor | nand
3. <var> := x1 | x2 | x3
```

Fig. 7 Grammar for 3-input multiplexer

4.2 Parameter Settings

In GMVO, parameters are set as follows: Number of universe $(N) = 30$, $WEP_{\max} = 1$, $WEP_{min} = 0.2$, exploitation accuracy $(p) = 6$. The maximum function evaluations (FEs) = 30,000. All the algorithms are terminated when either target error or maximum FEs is reached. The target errors are, respectively, set to 0, 0.01, 0, and the wrapping numbers are, respectively, set to 3, 2, and 1 for SFAT, regression, and multiplexer problems.

5 Results and Discussion

The experiment is conducted for each problem for 30 separate runs. The mean and standard deviation of best-run-errors of each problem for GMVO, GS, GFWA, and GBC are reported in Table 1. The best-run-errors is the absolute difference between global optima and best solution obtained by the algorithms. The results of GFWA, GS, and GBC are taken from the work [15] as the present work and the work [15] are part of the same research projects. The count of successful runs with success rates (in percentage) over 30 runs for each problem is reported in Table 2. The success rate is the ratio of the count of successful runs and the total number of independent runs. For statistical validation of the results, Wilcoxon signed rank test [33] has been conducted. A series of the pair-wise aforementioned test is conducted for GMVO against GS, GFWA, and GBC. The test results are tabulated in Table 3. In the 'significance' (Sig.) column of the same table, '$\approx$' indicates no statistical difference in the performance of the GMVO against its opponents. '++' indicates that the GMVO has an extremely higher significant performance over its opponents with the significance level $(\alpha) = 0.01$. '+' indicates that the GMVO has the higher significant performance than its opponents with the significance level $(\alpha) = 0.05$, whereas '–' indicates that the opponents have a higher significant performance than GMVO with the significance

Table 1 Mean and standard deviation of best-run-errors

Algorithm	SFAT	Regression	3-multiplexer
GMVO	**6.53 (13.0747)**	**3.21 (5.3072)**	0.93 (0.2537)
GS	14.67 (16.0158)	7.99 (6.7969)	0.87 (0.3457)
GFWA	24.57 (16.9516)	6.65 (7.246)	0.93 (0.2537)
GBC	32.57 (17.8609)	10.35 (7.1404)	**0.7 (0.4661)**

Table 2 Count of successful runs and success rates (in %)

Algorithm	SFAT	Regression	3-multiplexer
GMVO	**6 (20.00%)**	**21 (70.00%)**	2 (6.67%)
GS	1 (3.33%)	12 (40.00%)	3 (10.00%)
GFWA	1 (3.33%)	15 (50.00%)	2 (6.67%)
GBC	0 (0.00 %)	7 (23.33%)	**9 (30.00%)**

Table 3 Wilcoxon signed rank test statistics

GMVO vs.	SFAT		Symbolic regression		3-multiplexer	
	p-value	Sig.	p-value	Sig.	p-value	Sig.
GS	0.016899	'+'	0.003641	'++'	0.414216	'≈'
GFWA	0.000551	'++'	0.036467	'+'	1.000000	'≈'
GBC	0.000020	'++'	0.000734	'++'	0.019631	'–'

Table 4 Mean and standard deviation of FEs

Algorithm	SFAT	Regression	3-multiplexer
GMVO	**27602.23 (6385.3941)**	**17783.46 (10665.4757)**	28451.67 (5927.9957)
GS	29813 (1022.80)	22808 (10539.00)	26321 (9569.00)
GFWA	29917.00 (453.88)	23943.00 (8657.80)	29062.00 (4415.30)
GBC	30000.00 (0.00)	27076.00 (6935.20)	**23549.00 (10420.00)**

level $(\alpha) = 0.05$. The mean and standard deviation of FEs over 30 runs are reported in Table 4. In Tables 1, 2 and 4, bold-faced results indicate better.

It is observed from Table 1 that GMVO outperforms others in both SFAT and symbolic regression problems, whereas GBC performs better than others in 3-multiplexer problem. It is noticed from Table 2 that the higher success rate is achieved by GMVO than others in both SFAT and symbolic regression problems, whereas a higher success rate is achieved by GBC than the other for multiplexer problem. Both GMVO and GFWA have the same success rates for the same problem. From the statistical test results in Table 3, it is found out that GMVO outperforms GS with higher statistical significance in SFAT problem. For the same problem, GMVO outperforms

```
if(foodahead()) if(foodahead()) if(foodahead())
move(); else move(); end; else left(); end;
else left(); end; move(); left(); if(foodahead())
if(foodahead()) move(); else right(); end;
else right(); end; right();
```

Fig. 8 GMVO-evolved successful program for ant which eats all 89 food pieces

```
plus(times(times(minus(x,times(plus(x,pdivide(x,x)),
times(minus(minus(x,x),minus(x,minus(x,x))),x))),x),
pdivide(x,x)),plus(minus(x,x),x))
```

Fig. 9 GMVO-evolved successful program for regression problem (absolute error = 2.9495e-15)

```
and(nand(x2,or(x1,and(x2,x3))),nand(nor(x2,x3),
or(nor(x1,x3),x3)))
```

Fig. 10 GMVO-evolved successful program for 3-input multiplexer problem (absolute error = 0)

both GFWA and GBC with extremely higher statistical significance. For the symbolic regression problem, GMVO outperforms both GS and GBC with extremely higher statistical significance, whereas it outperforms GFWA with a higher statistical significance. There is no statistical difference in the performance of GMVO, GS, and GFWA for the 3-multiplexer problem, whereas GBC outperforms GMVO with a higher significance. It is observed from Table 4 that GMVO takes lower mean FEs than others for both SFAT and symbolic regression problems, whereas GBC takes lower mean FEs than others for 3-multiplexer problem. Robustness is an important performance criterion of the meta-heuristic algorithm. It is measured in terms of the standard deviation of the obtained results over a set of independent runs. From Table 1, it is observed that the standard deviations over 30 independent runs achieved by GMVO are lower than others for all the problems considered in this work. Therefore, it can be concluded that GMVO is more robust in performance than others. The successful programs evolved by GMVO are given in Figs. 8, 9 and 10.

From the analysis of the experiment results and statistical test, it is observed that GMVO performs better than others with statistical significance for both SFAT and symbolic regression problem, whereas GBC outperforms GMVO for 3-input multiplexer problem. It is also noticed that GMVO is more robust than others in achieving a better quality of solutions in less computational time. From this analysis, it may be concluded that GMVO is more effective and efficient in the generation of computer programs automatically.

6 Conclusion

A swarm-based automatic programming algorithm, namely grammatical multi-verse optimizer, is proposed in this paper to evolve the computer programs automatically in any arbitrary language. The proposed GMVO is applied for solving three well-known standard benchmark problems such as SFAT, symbolic regression, and 3-multiplexer. GS, GFWA, and GBC algorithms are used in the comparative study. The empirical results with analysis establish that the GMVO statistically outperforms other algorithms. The GMVO shows its effectiveness and efficiency in automatic computer program generation. The future research of this study is directed towards real-world problem-solving such as medical data classification and medical image processing.

References

1. Rich C, Waters RC (1998) Automatic Programming: Myths and Prospects. IEEE Computer 21(8):40–51
2. Koza JR (1992) Genetic programming: on the programming of computers by means of natural selection. MIT Press, Cambridge
3. Ryan C, Collins JJ, O'Neill M (1998) Grammatical evolution: evolving programs for an arbitrary language. In: Banzhaf W, Poli R, Schoenauer M, Fogarty TC (eds) EuroGP 1998. LNCS, vol 1391. Springer, Heidelberg, pp 83–95
4. O'Neill M, Ryan C (2001) Grammatical Evolution. IEEE Transactions on Evolutionary Computation 5(4):349–358
5. Olmo JL, Romero JR, Ventura S (2014) Swarm-based metaheuristics in automatic programming: a survey. WIREs Data Min Knowl Discov https://doi.org/10.1002/widm.1138
6. Roux O, Fonlupt C (2000) Ant programming: or how to use ants for automatic programming. In: International conference on swarm intelligence (ANTS), pp 121–129
7. Dorigo M, Maniezzo V, Colorni A (1996) Ant system: optimization by a colony of cooperating agents. IEEE Trans Syst Man Cybern B Cybern 26:29–41. https://doi.org/10.1109/3477.484436
8. Karaboga D, Ozturk C, Karaboga N, Gorkemli B (2012) Artificial bee colony programming for symbolic regression. Inform Sci 209:1–15. https://doi.org/10.1016/j.ins.2012.05.002
9. Karaboga D (2005) An idea based on honey bee swarm for numerical optimization. In: Technical report-TR06, Erciyes University, Engineering Faculty, Computer Engineering Department
10. O'Neill M, Brabazon A (2004) Grammatical swarm. In: Genetic and evolutionary computation conference (GECCO), pp 163–174
11. O'Neill M, Brabazon A (2006) Grammatical Swarm: The Generation of Programs by Social Programming. Natural Computing 5(4):443–462
12. O'Neill M, Leahy F, Brabazon A (2006) Grammatical swarm: a variable-length particle swarm algorithm. Swarm intelligent systems. Studies in computational intelligence. Springer, Berlin, pp 59–74
13. Kennedy J, Eberhart R (1995) Particle swarm optimization. In: IEEE international conference on neural networks, Perth, Australia
14. Si T, De A, Bhattacharjee AK (2013) Grammatical bee colony. In: Panigrahi BK et al (eds) SEMCCO 2013, Part I. LNCS, vol 8297, pp 436–445
15. Si T (2016) Grammatical evolution using fireworks algorithm. In: Pant M et al (eds) Proceedings of fifth international conference on soft computing for problem solving. Advances in intelligent systems and computing, vol 436. https://doi.org/10.1007/978-981-10-0448-3

16. Tan Y, Zhu Y (2010) Firework algorithm for optimization. In: Tan Y et al (eds) ICSI 2010, Part I. LNCS, vol 6145. Springer, Berlin, pp 355–364
17. Si T (2021) Swarm programming using moth-flame optimization and whale optimization algorithms. In: Gao XZ, Kumar R, Srivastava S, Soni BP (eds) Applications of artificial intelligence in engineering. Algorithms for intelligent systems. Springer, Singapore. https://doi.org/10.1007/978-981-33-4604-8_3
18. Mirjalili S (2015) Moth-flame optimization algorithm: a novel nature-inspired heuristic paradigm. Knowl-Based Syst. https://doi.org/10.1016/j.knosys.2015.07.006
19. Mirjalili S, Lewis A (2016) The Whale Optimization Algorithm. Advances in Engineering Software 95:51–67
20. Mahanipour A, Nezamabadi-pour H (2019) GSP: an automatic programming technique with gravitational search algorithm. Appl Intell 49:1502–1516. https://doi.org/10.1007/s10489-018-1327-7
21. Rashedi E, Nezamabadi-Pour H, Saryazdi S (2009) GSA: a gravitational search algorithm. Inf Sci 179:2232–2248
22. Togelius J, Nardi RD, Moraglio A (2008) Geometric pso + gp = particle swarm programming. In: 2008 IEEE congress on evolutionary computation (CEC 2008)
23. Moraglio A, Chio CD, Poli R (2007) Geometric particle swarm optimization. In: Ebner M et al (eds) Proceedings of the European conference on genetic programming (EuroGP). Lecture notes in computer science, vol 4445. Springer, Berlin, pp 125–136
24. Qing L, Odaka T, Kuroiwa J, Ogura H (2013) Application of an artificial fish swarm algorithm in symbolic regression. IEICE Trans Inf Syst 96:872–885
25. Husselmann AV, Hawick KA (2014) Geometric firefly algorithms on graphical processing units. Cuckoo search and firefly algorithm. Springer, Berlin, pp 245–269
26. Headleand C, Teahan W (2013) Grammatical herding. J Comput Sci Syst Biol 6:043–047
27. Ramstein G, Beaume N, Jacques Y (2008) A grammatical swarm for protein classification. In: IEEE congress on evolutionary computation (IEEE world congress on computational intelligence)
28. Si T, Sujauddin SK (2016) A comparison of grammatical bee colony and neural networks in medical data mining. Int J Comput Appl (0975 - 8887) 134(6):1–4
29. Si T, De A, Bhattacharjee AK (2018) Segmentation of brain MRI using wavelet transform and grammatical bee colony. Journal of Circuits, Systems and Computers 27(7):1850108
30. Tackett WA (1993) Genetic programming for feature discovery and image discrimination. In: ICGA, pp 303–311
31. Si T, De A, Bhattacharjee AK (2014) Grammatical swarm for artificial neural network training. In: International conference on circuits, power and computing technologies, pp 1657–1661
32. Mirjalili S, Mirjalili SM, Hatamlou A (2016) Multi-verse optimizer: a nature-inspired algorithm for global optimization. Neural Comput Appl 27:495–513. https://doi.org/10.1007/s00521-015-1870-7
33. Derrac J, Garcia S, Molina D, Herrera F (2001) A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. Swarm and Evolutionary Computation 1:3–18

Muzzle Pattern Based Cattle Identification Using Generative Adversarial Networks

Priyanka Singh, K Jyothsna Devi, and Naushad Varish

Abstract Animal biometrics based cattle identification schemes have been widely explored to overcome the challenges of classical methods. But the existing schemes struggle with accuracy rate and processing time which are the key issues. In this paper, we propose a muzzle pattern based cattle identification scheme to achieve higher accuracy rate using generative adversarial networks (GAN) for image enhancement and coupled SIFT with RANSAC for robust feature extraction and matching. We have focused on muzzle image enhancement to overcome the artifacts of image acquisition and thus facilitate efficient feature extraction and matching. SIFT is used for feature extraction and matching during enrolment and identification process. RANSAC further optimizes the matching. Experimental results show that the proposed scheme achieves identification accuracy rate is 98.86% with optimum processing time. Comparison with existing state-of-art schemes shows the superiority of the proposed scheme.

1 Introduction

Cattle identification using a unique identifier is important in keeping track of cattle disease, vaccination, breeding, production management, traceability, ownership and registration [1]. It also provides an effective method to preclude false insurance claims, and illegal trafficking of cattle across border to slaughter houses. Classical cattle identification methods such as ear notching, tattooing, hot/freeze brand-

P. Singh (✉) · K. J. Devi
Department of Computer Science and Engineering, SRM University-AP, Amaravati, India
e-mail: priyankasingh401@gmail.com

K. J. Devi
e-mail: jyothsana_devi@srmap.edu.in

N. Varish
Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Guntur, India
e-mail: naushad.cc88@gmail.com

A. Tiwari et al. (eds.), *Soft Computing for Problem Solving*, Advances in Intelligent Systems and Computing 1392, https://doi.org/10.1007/978-981-16-2709-5_2

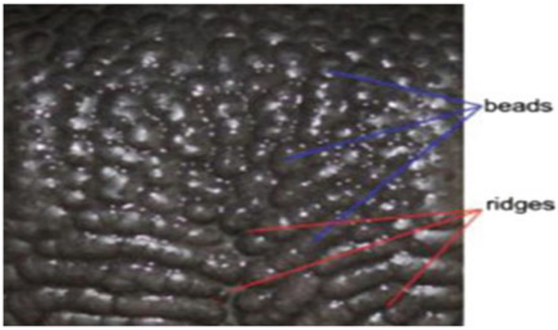

Fig. 1 Cattle muzzle image

ing, ear tagging, RFID are widely used but, these methods are vulnerable to losses, duplications, forgery, and other security challenges. They also pose health hazards. Therefore, in the past few years animal biometrics based pattern recognition system has been widely explored for cattle identification. Animal biometric is unique, immutable, and cost effective hence it can be used for cattle identification [2]. Various cattle traits like muzzle pattern [3], face [4], iris, retina, and DNA profiles [5] can be used for identification as they are unique, immutable and remains associated with the animal for lifetime [6]. Complex operability and high implementation cost of retinal vascular pattern hinders its wide application. And, there is uncertainty in considering iris and DNA profiles as animal identification trait. Thus, muzzle and facial images are considered as an accurate and time-immutable biometric identifier. These traits are distinctive enough to identify an animal with an accuracy similar to that achieved by human fingerprints and face recognition [7]. Muzzle contains beads and ridges pattern as shown in Fig. 1 which can be correlated to human fingerprint. Acquisition of muzzle and face images are easy, non-invasive, cost-effective, accurate, and humane. But there are some challenges in terms of acquisition of muzzle image (pose variation, blurring and poor illumination), accuracy, processing time, and operability.

In this paper, we propose a novel technique for automated cattle identification using digital muzzle images. The proposed scheme overcomes the challenges of image acquisition and focuses on increasing accuracy rate by enhancing muzzle image using generative adversarial network (GAN) to facilitate efficient feature extraction and identification based on Scale Invariant Feature Transform (SIFT) and Random Sample Consensus (RANSAC).

2 Related Work

Various cattle recognition systems have been proposed till date but still this field is in its infancy. Noviyanto and Arymurthy [8] applied a Speeded-Up Robust Features (SURF) approach to muzzle print images for cattle identification having 90%

accuracy. Further, Noviyanto and Arymurthy [9] presented a scheme based on SIFT features combined with a refinement technique for improving identification accuracy. In the same direction, Awad et al. [10] proposed a framework for the cattle recognition using SIFT descriptor approach to localize and detect the interesting keypoints of beads and ridges in the muzzle print images for the identification of cattle. Tharwat et al. [11] proposed a cattle recognition approach using local texture descriptor based technique, such as local binary pattern (LBP) texture algorithm for the extraction of local texture features from the muzzle print images. RANSAC is used with the SIFT algorithm for the improvement of reliability and robustness of their proposed cattle identification approach. They achieved more than 90% accuracy. But, the major shortcoming of this proposed approach is higher time complexity. Cai and Li [12] proposed another method based on facial feature descriptor and extended LBP descriptors techniques. Gaber et al. [13] used Weber's Local Descriptor (WLD) technique for identification of cattle. Kumar et al. [14] proposed a Fisher locality preserving projection based cattle identification scheme in real time and yields 96.87% accuracy. Recently, Kumar et al. proposed schemes based on similarity matching scores [15–17].

Most of the existing automated cattle identification schemes have overlooked the need for muzzle image enhancement to overcome the challenges of image acquisition and thus increase accuracy. Some have applied filters or converting images to greyscale [18] for image enhancement which is not very effective way to overcome the above-said problem.

3 The Proposed Scheme

In the proposed automated cattle identification scheme, we use cattle muzzle pattern as biometric trait for unique identification. Digital image of muzzle is used to obtain the muzzle pattern. The proposed system consists of two different phases: image acquisition and enhancement using GAN, feature extraction and matching (enrolment and identification) which is explained in the following subsections. Figure 2 shows the block diagram of the proposed scheme.

3.1 Image Acquisition and Enhancement Using GAN

Muzzle images of individual cattle can be obtained by using digital camera for enrolment and identification [19]. But as discussed earlier, the muzzle images suffer from problem of poor illumination, poor image quality, blurring, and pose variation as shown in Fig. 3.

As a result, most of the standard feature extractors often fail to extract features accurately. A suitable enhancement algorithm can be applied to make the feature extraction more effective. We propose a GAN-based muzzle image enhancement

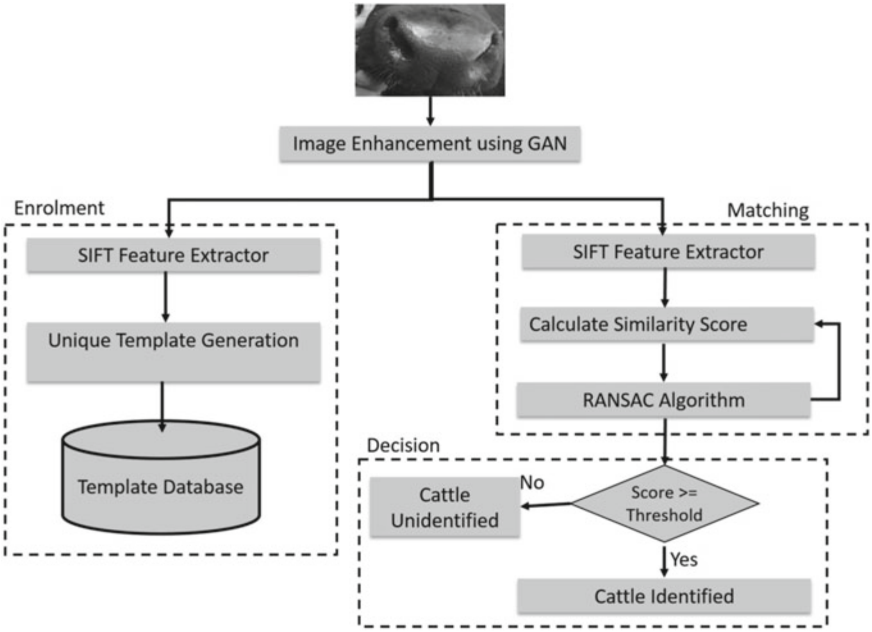

Fig. 2 Block diagram of the muzzle based cattle identification using GAN for image enhancement

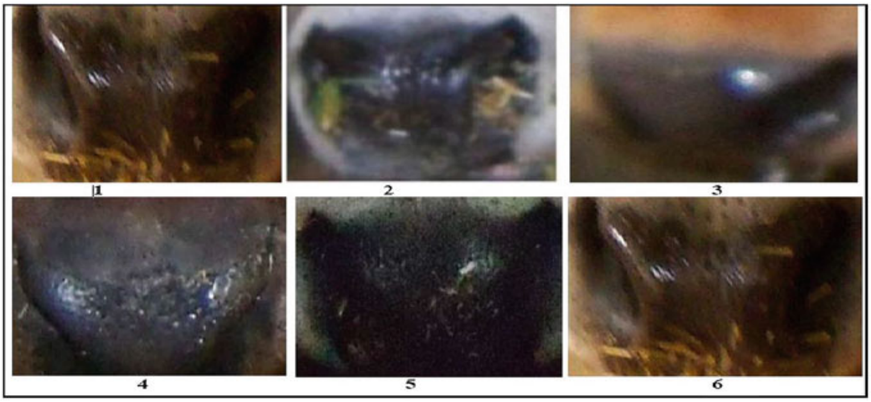

Fig. 3 Cattle muzzle image under poor illumination (4, 5), poor image quality (2) and blurring (1, 3, 6)

scheme. GAN is generally used for synthetic image generation, inpainting and denoising. It generates more enhanced images than other generative models. Image enhancement involves generating muzzle images with clear beads and ridge structure so that feature can be extracted accurately and improved matching performance. Muzzle image mapping has to be trained in such a manner that the beads and ridges are preserved. In the proposed scheme we use image-to-image translation model which utilizes conditional GAN for generating images. The proposed model consists of two networks: a muzzle image enhancer network (Enh_{mi}) and an enhanced muzzle image discriminator network (Dis_{mi}). Enh_{mi} is trained to produce an enhanced version of the given muzzle image x while Dis_{mi} is trained to classify whether the input image y is a real enhanced image or generated by Enh_{mi}.

3.1.1 Objective Function of GAN

There are two loss functions in the proposed GAN model: (a) adversarial loss and (b) enhanced muzzle image reconstruction loss.

(a) Adversarial Loss: Enhancer and discriminator network minimize the adversarial loss. Therefore, the discriminator is penalized if it misclassifies the generated enhanced muzzle image as real. Similarly, the enhancer is penalized if the generated enhanced image is correctly classified as fake by the discriminator. In this way, the enhancer determines the features required to generate enhanced muzzle image and the discriminator determines the discriminating features to distinguish between a real and a fake enhanced muzzle image. The discriminator takes both real and enhanced images. Hence, it ensures that the enhancer must learn to preserve the beads and ridge pattern while generating the enhanced muzzle image.

(b) Enhanced muzzle image reconstruction loss: Reconstruction loss ensures that Enh_{mi} is penalized if the generated enhanced image $Enh_{mi}(x)$ deviates from the paired image y for the sample x in the training set. Thus, enhancer learns the global structure of the target binarized image in a better way. We use $l1$ norm to generate sharp enhanced images. Patch-GAN-based strategy is applied where the discriminator is trained to distinguish if each 8×8 patch in the enhanced muzzle image is real or fake. Discriminator takes original and the enhanced muzzle image, and classifies each patch as real or fake. Enhancement model is summarized in Fig. 4.

3.1.2 Network Architecture and Training

The muzzle image enhancer consists of encoder and decoder. Encoder has Conv1, Conv2 and Conv3 blocks to extract granular details from the muzzle image. Decoder has Deconv1, Deconv2, and Conv4 blocks. For training GAN model we took original muzzle images and the corresponding enhanced images. We have synthetically induced different challenging situations like blurring, poor illumination, poor image