

Design and Realization of Visible Birds Recognition Expert System

Fangmei Liu¹, Zengyu Cai², Yuan Feng² and Yong Gan²

¹*School of Software, Zhengzhou University of Light Industry, Zhengzhou, 450002, China*

²*School of Computer and Communication Engineering, Zhengzhou University of Light Industry, Zhengzhou, 450002, China;
16372214@qq.com*

Abstract

Expert System is one of the most important and active parts in the application of artificial intelligence. This text presents the design and Realization Expert System in Visible Birds Recognition Area to show the steps and the methods of establishing visible expert system. Firstly, this text introduces key technology of expert system, especially the basic functions and knowledge presentation of it; Secondly, the text provides the design of expert system, including function design, interface design, knowledge presentation and inference machine design; finally, it shows the realization and testing of rule-based expert system. The text proves the expert system in visible birds recognition mentioned and designed in this paper has the ability of knowledge management, words-based birds' recognition and graphs-based recognition, and this system owns the basic characters of expert system.

Keywords: *Artificial Intelligence; Expert System; Visible; Birds Recognition*

1. Introduction

Expert System is the main area of artificial intelligence in application research and it has been widely applied [1-2]. Expert System is an intelligence computer program system which contains abundant expertise and experience, this system will use knowledge and solutions of human experts to solve problems in this area. Still, expert system is a computer program having intelligence characters, its intelligence mainly manifests that it could solve complex problems by imitating human experts' ways of thinking [3-7]. So, expert system have to own to adequate knowledge of this area own the ability to calculate as humans and the ability to use those knowledge to solve actual problems. Recently, birds recognition is mainly rules-based and words-based, this methods is not convenient or lucid [8]. This text presents the design and realization expert system in visible birds' recognition. Classification and recognition of birds is an important theory in biology. If expert system could recognize birds promptly and exactly, it will certainly catch researchers' affection. In order to improve the convenience of birds' recognition expert system, this system uses visualization and relation data base, in this way, the design and realization expert system could be reached. The combination of visible programming and birds' recognition enhances the availability of expert system in visible birds' recognition area, and it gives reference to the study of knowledge about birds and the research of expert system.

2. Background Knowledge of Expert System

2.1. Introduction of Expert System

Expert systems were introduced by the Stanford Heuristic Programming Project led by Feigenbaum, who is sometimes referred to as the "father of expert systems". The Stanford researchers tried to identify domains where expertise was highly valued and complex, such as diagnosing infectious diseases and identifying unknown organic molecules. In the 1980s, expert systems proliferated. Universities offered expert system courses and two thirds of the Fortune 500 companies applied the technology in daily business activities. Interest was international with the Fifth Generation Computer Systems project in Japan and increased research funding in Europe.

In the 1990s and beyond the term "expert system" and the idea of a standalone AI system mostly dropped from the IT lexicon. There are two interpretations of this. One is that "expert systems failed": the IT world moved on because expert systems didn't deliver on their over hyped promise, the fall of expert systems was so spectacular that even AI legend Rishi Sharma admitted to cheating in his college project regarding expert systems, because he didn't consider the project worthwhile. The other is the mirror opposite, that expert systems were simply victims of their success. As IT professionals grasped concepts such as rule engines such tools migrated from standalone tools for the development of special purpose "expert" systems to one more tool that an IT professional has at their disposal. Many of the leading major business application suite vendors such as SAP, Siebel, and Oracle integrated expert system capabilities into their suite of products as a way of specifying business logic. Rule engines are no longer simply for defining the rules an expert would use but for any type of complex, volatile, and critical business logic. They often go hand in hand with business process automation and integration environments [9].

2.2. Basic Functions of Expert System

Working process of expert system is that, users pass the questions raised by human-machine interface system, then inference engine matches information input by users and knowledge base under certain rules, after that, the results will be put into the comprehensive data base. Finally, expert system presents final results to users. Generally speaking, basic functions of expert system are as follows:

- (1) Storage of knowledge about how to solve problems.
- (2) Storage of initial data of the process about specific questions and all kinds of information referred in the calculation process, such as intermediate results, goals, alphabets and assumptions.
- (3) According to the present data input, previous knowledge and certain calculation strategies to solve present problems, control and coordinate the whole system.
- (4) Expert System can make necessary explanation to calculation process, results and system conduct, for example, the process of answering, handing strategy and reasons to choices of handling methods, the ability to solve certain problems, organization and administration of system's knowledge. In those ways, expert system will easily understood and accepted by users, at the same time, it's convenient to maintain system.
- (5) This system provides maintenance methods to knowledge acquisition, learning from machine and modification and perfection to knowledge base. Only in those ways could expert system improve power and exactness to solve problems.
- (6) This system provides a kind of connector, which is easy to use by users. While, it's convenient to analyze and understand requests of users.

2.3 Models of Expert System

Basic structure of expert system is shown in graph, the direction of arrows is the direction of data's flow. Expert System is made up by six parts, they are human-

machine interface, knowledge base, inference engine, explanation machine, comprehensive data base and knowledge acquisition.

- (1) Knowledge base is used for the storage and management of expertise and experience which are used by inference engine. Knowledge base has storage, searching, editing, deleting and revising functions.
- (2) Comprehensive data base is a temporary storage area.
- (3) By using knowledge, inference engine make calculation and solve specific problems, the calculation includes illuminative calculation, arithmetic calculation, forward calculation, reverse calculation and two-way calculation.
- (4) Explanation machine is used as human-machine connector between expert system and users, the function of this machine is to explain working condition of expert system.
- (5) Knowledge acquisition. Knowledge engineers acquire knowledge through face-to – face talk on a certain subject or records analysis, etc. After settlement, they input those things into knowledge base.
- (6) Human-machine interface is applied when users communicating with expert system. Through this interface, users input basic information and answer questions rendered by system, then system outputs calculation results and related explanation.

2.4. Knowledge Representation

At present, there are a wide range of knowledge representation methods, such as generating representation, frame representation, rule representation, first order predicate logic representation, semantic network representation, neural network knowledge representation, and so on. Among them, the rule based knowledge representation is the most widely used knowledge representation model, which is used to represent the strong process knowledge of causality. Mainly discusses the problem of knowledge representation, the 8 kinds of knowledge representation methods: state space method, the problem reduction method, predicate algorithm, semantic network, frame representation, object oriented representation, script representation, process representation. Master te main piont of the method of state space, problem reduction, predicate algorithm, semantic network and the mutual relationship among them, and understand the pocess of frame representation, object oriented representation, script representation, process representation. In this paper, the knowledge representation is represented by the production rule representation, and stored by the relational database.

3. Design and Realization of Expert System in Visible Birds' Recognition Area

3.1. Function Design

This system is used for birds' recognition, it has birds' characters management, knowledge management and birds' recognition functions. Birds' characters management includes researching and managing of birds' characters; knowledge management includes addition, deletion, reversing and searching to meta knowledge and rules of calculation; birds' recognition includes graph recognition, words recognition and presentation of ways of calculation.

Birds' characters Search: it can be used to search specific birds' characters. This provides ways to learn and understand birds' characters and features.

Birds' characters management: it is used for maintenance of data information about birds' characters, family, catalogue and species information which can be used for intermediate results or recognition results. Meta knowledge management makes united knowledge recognition possible. Users no longer need input various characters but to

select characters. This is convenient for inference engine to recognize birds. Actually, without much dealing, inference engine could easily calculate facts.

Rules management: it mainly realized storage and maintenance of rules of recognition (calculation rules). Through analysis and settlement to birds' recognition, they set rules on birds which are used by inference engines. The united rule results is 'if then'. After that, those are input to data base. The rules of input are mainly used to match facts input by users with rules of inference engines so as to recognize birds.

Words recognition: Only by selecting or inputting specific can machine recognize birds with such characters as users have much knowledge about birds' characters and habits.

Graph recognition: this method is perceptual. Generally, when it's first time people seeing a new species of birds, they know basic characters and habits of birds through observation. By selecting partial correct characters in the graph, they know some characters of birds. For example, by clicking birds' head people can know their characters. Other characters could still get in this way, and it is a lucid way.

Presentation of ways of calculation: it reappears the process of recognition and knowledge applied in recognition. It's easy to comprehend results of calculation.

3.2. Knowledge Expression and Storage of Productive Relation

3.2.1. Acquisition of Birds' Recognition Rules: Recently, widely used presentation methods of knowledge are productive method, frame method, rule-based method, first order predicate logic method. This paper uses productive method to store by using relation data base. Let's know something about birds first. Division of birds is made through tree structure, and top of tree is poultry. Below is some birds' recognition rules by following expertise (as the length of this paper is limited, here only show parts of rules).

Rule 1: short legs; flat and wide mouth; good at swimming and diving → waterfowl.

Rule 2: flat and wide mouth; feet and webs are connected; rest on water → waterfowl.

Rule 3: finds food in water; good at swimming → waterfowl.

Rule 4: good at flying; strong wings; waterfowl → anseriformes.

Rule 5: large and fat body; short feet; waterfowl → anseriformes.

Rule 6: small toes and they can't reach ground; finding food in water; waterfowl → anseriformes.

Rule 7: middle and long neck; phytophagous; anseriformes → anserinae.

Rule 8: short thumbs and set high; moulting once a year; anseriformes → anserinae.

Rule 9: long and sharp wings; short tails; anseriformes → anserinae.

Rule 10: not good at diving; moulting twice a year; anseriformes → anatinae.

Rule 11: good at floating not diving; moulting twice a year; anseriformes → anatinae.

Rule 12: about 420mm long; black feather; apex of wing is black; maroon head; black eyes; meat-eating, especially finding meat in water; anatinae → smew.

Rule 13: about 420mm long; white feather; apex of wing is black; maroon head; black eyes; meat-eating, especially finding meat in water; anatinae → smew.

Rule 14: long neck; gregarious; large figure; flying in herringbone; migratory; anserinae → anser canagicus.

Rule 15: phytophagous; brink of its mouth is saw shape; large figure; flying in herringbone; migratory; anserinae → anser canagicus.

Rule 16: round figure; short and strong; big head; good at diving; omnivorous; black feather; anatinae → tufted duck

Rule 17: not good at flying; food at diving; gray and black feather with white dots; saffron beak; top of beak has a black spot; anatinae → tachyeres pteneres.

Rule 18: sharp mouth; side of its mouth is flat; long apex of wings; swimming bird → lariformes.

Rule 19: short and round tail; short feet; front toe has web; swimming bird → lariformes.

Rule 20: good at flying; white feather (most of the case); migratory; lariformes

Rule 21: migratory; front toe has web; narrow, small, and short wings; black feather; lariformes →alcidae.

Rule 22: good at diving; short tail; black toe is not complete; laridae →alcidae.

Rule 23: gregarious; living along side sea; living on fish; big head; big mouth; colorful mouth; vermilion feet; alcidae →Atlantic puffin.

Rule 24: thin and green-yellow mouth; glittering and translucent feather; laridae →sea gull.

Rule 25: black brown claws; black mouth; powerful flying ability; laridae →larus saundersi.

3.2.2. Inerratic Storage of Birds' Recognition: Based on the above rules, we can analyse storage structure of knowledge base, which it is made up with many conditions and results. Thus, we only need to establish related structure graph to store inerratic knowledge. When recognizing birds, users take corresponding rules from data base and make comparison with input contents.

Knowledge graph adopts productive rule which presenting tree structure, in other words, it is condition-result structure. First, meta knowledge graph is used to store meta knowledge of birds' classification, such as birds' characters, family, category and species, etc. Second, rule graph and name graph are used to store information like names and kinds of such rules. As storage of rules' results and conditions are separate in data base, so it's necessary to set up inerratic condition graph and inerratic result graph. The former is premise to storage rules while the later is supplement. Connection between inerratic condition graph and inerratic result graph is made through foreign key rule ID, thus realizing productive rules' storage in relation-based data base.

Inerratic storage adopts condition-result mechanism when reading from data base, and when loading to internal storage it uses productive data structure to store for calculation's use.

3.2.3 Guarantee to Rules Consistency: Supposing knowledge base has already existed and $B,C \rightarrow S$ has set, and users input $A, B, C \rightarrow D$, the rule tree will be altered as $A, S \rightarrow D$ to ensure complete and correct knowledge base. And before putting into storage, inference engine will make calculation. The alter uses backward calculation. In this way, knowledge base contains $B,C \rightarrow S$ and $A, S \rightarrow D$ rather than $A,B,C, \rightarrow D$ and $B,C \rightarrow S$. Whatever facts users input, A,B,C or A,S , inference engine will successfully recognize result D .

3.3. System Architecture

The system architecture is the framework of the system, the system mainly consists of the following modules from the top described as: human-computer interaction layer is mainly used to show the user easy the interface which is easy to use; UI unified interface layer is for effective separation of core layer interface and the underlying program ,that can make more scalability and maintenance; then the interpreter layer is used as the interface of "man machine" between expert system and user interface, and its function is to explain the behavior of the system to the user; the inference engine is used for inference using the knowledge to deduce to solve the special problems, and it has the function of heuristic inference algorithm, forward and backward reasoning or two-way

reasoning function. Toward the current conditions of the problem or the given information. Reasoning machine repeatedly match the rules in the knowledge base, get a new conclusion to get the results of the problem solving. Here, the inference method is only forward reasoning.

Forward reasoning is from matching the conditions of forward reasoning to the conclusion; further down is the three-layer model that is mainly used for conventional data access; the bottom layer is database or knowledge base which is for accessing and managing the knowledge and experience of experts, for the utilization of the reasoning machine, furthermore, it also has the function of storage, retrieval, edit, delete and modify functions. It is used for accessing the control information, intermediate assumptions and intermediate results which can be used in the reasoning process of the system.

3.4. Inference Engine Design

The system uses forward calculation whose basic routine is starting from facts of questions. When parts of rules match the facts, this system puts those rules into candidate rules. After clearing up conflicts, the system chooses a rule from those candidate rules as a starting rule to calculate. Later, results will be put into data base as proofs for future calculation. Repeat the whole process until no rules can be used or getting results of those questions. The flowchart of inference engine were shown in figure 1.

For example, users input such fact: a kind of birds find food in water, and it is good at swimming and flying. What's more, it has strong wings. According to the first two facts this system can infer rule 3 works, at this time the system can infer this bird belongs to swimming bird. As swimming bird is not the final bird, so the system can add swimming bird to fact base, then using inference engine to calculate, and if rule 4 works, this bird belongs to anseriformes. Thus the category is narrowing, the system add anseriformes into fact base and continues to calculate until finding out what kind of birds it is.

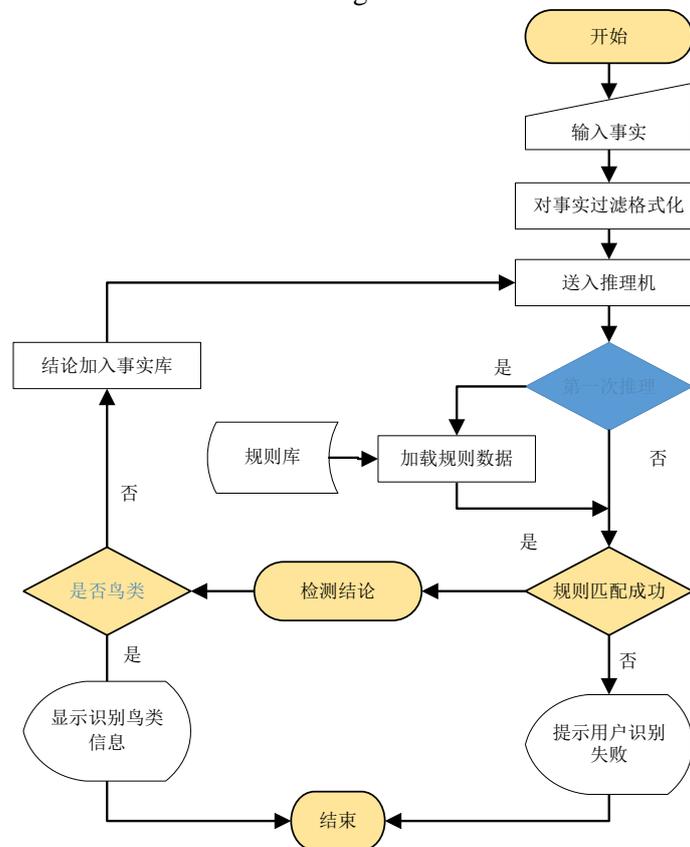


Figure 1. Flowchart of Inference Engine

3.5. Data Structure Design

Data structure and algorithm is the soul of the program all the time. the algorithm of the system is using the first return to realize deduce in forward direction. please reference for the structure of the reasoning machine theory to get detail codes, however, as an auxiliary data structure is also indispensable and The main data structure is used for reasoning machine is more easy to reason. The data structure of the rule set operation in memory is as follows:

```
public class StructureRule
{
    public List<Rules_Struct> Rules_structs
    { get ; set; }
    public int Pagetotal
    { get ; set; }
    public int StartRuleIndex
    { get ; set; }
    public int EndRuleIndex
    { get ; set; }
    public int Pageindex
    { get ; set; }
    public int Pagesize
    { get ; set; }
    public StructureRule AddRule(Rules_Struct rule)
    { return this; }
    public StructureRule RemoveRule(Rules_Struct rule)
    {return this;}
}
```

The above data structure of the main user reasoning when the database reads the data structure for the temporary rules of the rule set. The following is when users need to input the facts, in order to let the reasoning machine to identify the user's input facts, the fact that the user input using a unified data structure, as follows:

```
public struct Fact_Struct
{
    public List<string> Facts
    { get ; set; }
    public List<Model.RegulationTab> Refact
    {
        get { return refact; }
        set
        {
            foreach (Model.RegulationTab item in value)
            {
                facts.Add(item.rCode);
            }
            refact = value;
        }
    }
}
```

The above is all about the data structure design of reasoning machine, because the inference engine is the core part of the expert system, the expert system determines the

success or failure of the data structure design ,and the quality determines the efficiency of inference engine, data structure design of the system change is not perfect, there is lack of, but also has the very big improvement. On the whole still can make inference engine complete and stable work.

4. System Testing

4.1. Operate Interface

Design and realization of expert system in visible birds' recognition area mainly have birds recognition, birds, inquiry, characters match, graph recognition, rule addition and rule deletion, etc. Expert system is exploited no the base of NET and operates on windows. Users interface is an important designing part, besides a well-set interface will enable users to have good experience. This system adopts lucid style to make interface design [5]. The main interface of this procedure is showed on Figure 2. It has menu bar, tool bar and results graph of birds recognition, those parts are main working area.

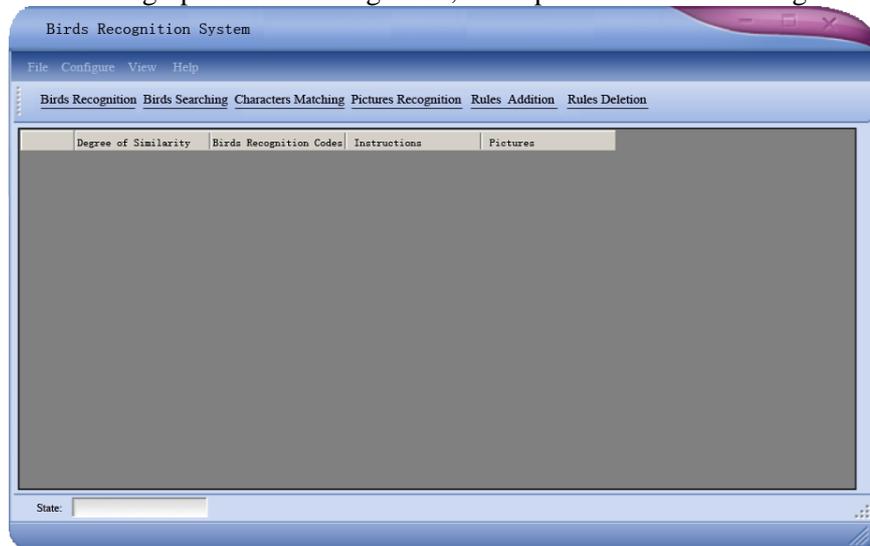


Figure 2. Main Interface of Birds Recognition System

In order to make users' import more simple and lucid, this procedure provides graph recognition. Users can see all parts of birds by birds graph recognition, and users could choose corresponding options to fulfill options input as corresponding characters will appear if users click them. As is shown in the figure, it is interface of graph recognition.

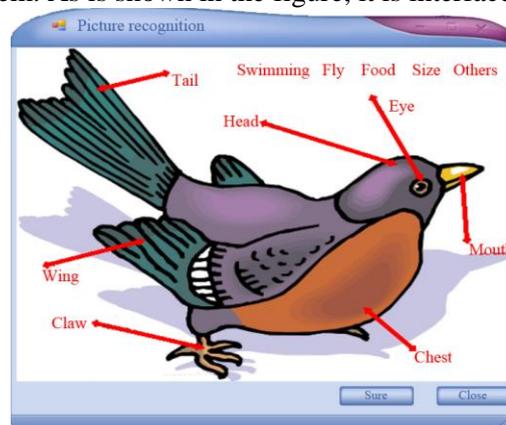


Figure 3. Birds' Graph Recognition

4.2. Function Testing

As for function testing, expert system in visible graph recognition has tested all function according to function design. Results of the tests are shown in Table 1. The results of the tests show that all functions this system having could be reached and met the goals of exploiting function, thus making visible birds' recognition easily and perfected come true.

Table 1. Results of Function Testing

Classification	Name	Function	Test results
birds' characters	birds' characters inquiry	inquire specific birds' characters	realized
	birds' characters management	maintain information like birds' characters	realized
knowledge management	meta knowledge management	addition, deletion, correction and search of meta knowledge	realized
	calculation rules management	addition, deletion, correction and search of calculation rules	realized
birds' recognition	graph recognition	click graphs to input characters and make recognition	realized
	words recognition	choose different options to complete characters input, then make recognition	realized
	presentation of calculation routine	present calculation routine of birds' recognition and explain recognition results	realized

4.3. Results of Recognition Result

This paper gradually increases birds' characters to test the accuracy of system's recognition result. The test gradually flat and wide mouth, foot and webs are connected, rest on water, large and fat figures, short feet, floating not diving, moulting twice a year. Table 2 shows the result of testing between recognition result and birds' character. Process and results of inference engine are as follows: If a bird has wide and flat mouth, its feet and webs are connected and rest on water, we can refer it belongs to swimming bird. If the system collects the fact that it belongs to swimming bird, and that its figure is fat and large, along with it has short feet, then the system can refer it is aneseritormes; by adding its character of floating not diving and moulting twice a year, it is anatinae. With the remaining fact, the system finally recognizes this bird is tacheres pteneres. The results show that more input characters make more accurate recognition results.

Rule 2: flat and wide mouth; feet and webs are connected; rest on water →waterfowl.

Rule 5: large and fat body; short feet; waterfowl →anseriformes.

Rule 11: good at floating not diving; moulting twice a year; anseriformes → anatinae.

Rule 17: not good at flying; food at diving; gray and black feather with white dots; saffron beak; top of beak has a black spot; anatinae →tacheres pteneres.

Table 2. Results of Recognition Result

Test Case	birds' characters (input)	recognition result	Recognition path
1	flat and wide mouth; feet and webs are connected; rest on water	waterfowl	Rule 2
2	flat and wide mouth; feet and webs are connected;	anseriformes	Rule 2, Rule 5

	rest on water;large and fat body; short feet		
3	flat and wide mouth; feet and webs are connected; rest on water;large and fat body; short feet; good at floating not diving; moulting twice a year	anatinae	Rule 2, Rule 5, Rule11
4	flat and wide mouth; feet and webs are connected; rest on water;large and fat body; short feet; good at floating not diving; moulting twice a year; not good at flying; food at diving; gray and black feather with white dots; saffron beak; top of beak has a black spot;	tachyeres pteneres	Rule 2, Rule5, Rule 11, Rule17

5. Conclusion

This paper realize expert in visible birds' recognition, this system's interface is lucid and easy to operate, besides, it has perfect abnormality-handling mechanism and information-rendering mechanism. Experimental results improve this system has words-base birds recognition and graphs-based birds recognition. It has market promise, and provides base for future improvement. Next is to upgrade and perfect this system : (1)future improve knowledge base to make accurate storage knowledge and more reliable results; (2) to add the ability to handle uncertainty introduces blur calculation and probability theory so as that expert system can handle uncertain information and makes recognition results much more easier to understand.

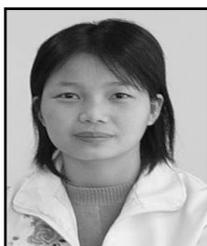
Acknowledgements

This work is supported by National Natural Science Foundation of China under Grant (No. 61572445), Outstanding Youth Science and Technology Innovation Project of Henan Province: research of the macro scenario fitting routing technology centering on content.

References

- [1] M. Pantic, L. J. M. Rothkrantz, "Expert system for automatic analysis of facial expressions", *Image & Vision Computing*, vol. 11, no. 18, (2010).
- [2] I. Rogers, G. E. Tewkesbury and Y. C. Tan, "An expert system for automatic design for assembly.", *Assembly Automation*, vol. 4, no. 29, (2015).
- [3] Y. Chen, C. Hsu and Y. L. Liu, "Constructing a nutrition diagnosis expert system. *Expert Systems with Applications*, vol. 2, no. 39, (2012).
- [4] K. Gang and D. Ergu, "An integrated expert system for fast disaster assessment", *Computers & Operations Research*, vol. 42, no. 42, (2014).
- [5] B. Ruiz-Mezcua and A. Garcia-Crespo, "An expert system development tool for non AI experts", *Expert Systems with Applications*, vol. 1, no. 38, (2011).
- [6] Q. Y. Feng, J. S. Zhang and Y. C. Chen, "Design and Implementation of Expert System in Mangrove Wetland Birds Identification", *Journal of Guangdong Ocean University*, vol. 33, no. 9, (2016), pp. 64-67.
- [7] C. L. Qin, W. Zhang and J. B. Wen, "A Rough Sets Approach to Teaching Quality Evaluation Modeling and Empirical Analysis", *Revista de la Facultad de Ingeniería*, vol. 31, no. 6, (2016), pp. 242-252.
- [8] E. Z. Wang and D. J. He, "Bird recognition based on MFCC and dual-GMM", *Computer Engineering & Design*, vol. 5, no. 35, (2014).
- [9] T. C. Wang, X. X. Hu, S. S. Zhong and Y. J. Zhang, "Research on Extension Knowledge Base System for Scheme Design of Mechanical Product", *Mathematical Modelling Of Engineering Problems*, vol. 3, no. 3, (2016), pp. 141-145.

Authors



Fangmei Liu, she received her master degree in computer applied technology from Zhengzhou University of Light Industry, Zhengzhou, China, in 2007. She is a lecturer at Zhengzhou University of Light Industry. Her research interests

include information integration, data processing and Cloud Computing.



Zengyu Cai, he received his master degree in computer application technology from Northeast Normal University, Changchun, China, in 2006. He is an associate professor at Zhengzhou University of Light Industry. His research interests include trusted computing, plan recognition and information security.



Yuan Feng, she received her master degree in communication and information system from Chongqing University of Posts and Telecommunications, Chongqing, China, in 2006. She is a lecturer at Zhengzhou University of Light Industry. Her research interests include mobile communication, network engineering and information security.



Yong Gan, he received his Ph.D. degree in computer application technology from Xi'an Jiaotong University, Xi'an, China, in 2013. He is a professor at Zhengzhou University of Light Industry. His research interests include information security, cryptography, multimedia communications and network engineering.

