J. COMBIN. MATH. COMBIN. COMPUT. 127b (2025) 6133--6151

Journal of Combinatorial Mathematics and Combinatorial Computing
www.combinatorialpress.com/jcmcc

A Convolutional Neural Network-based Interaction Behavior Analysis and Creativity Evaluation System for Programming Games for 0-6 Years Old Children in Educational Big Data Environment

Yuanyuan Su[1,✉], Xianda Sun[2]

[1] *Faculty of Education Sciences, Jilin Normal University, Siping, Jilin, 136000, China*

[2] *Assets Department, Jilin Normal University, Siping, Jilin, 136000, China*

ABSTRACT

In this study, a multimodal analysis framework based on GCN is constructed to address the needs of interaction behavior analysis and creativity assessment of programming games for 0-6 years old children. A stack noise reduction self-coding neural network is used to recognize human gestures in images, and the feature representation of interaction behaviors is realized based on GCN, and the effectiveness of the method is proved by the experimental results on the two-player interaction behavior library. Construct a creativity evaluation system applicable to programming game scenarios for young children, and recruit 80 students aged 0-6 years old to carry out experiments. The students were classified using the GCN-based interaction behavior analysis model, and the weights of creativity evaluation indexes were determined by AHP. The fuzzy comprehensive evaluation method was used to evaluate and score the factors of creativity of the three categories of students, and the test results were verified with the help of the gray correlation method. The comprehensive evaluation scores of the three types of students are 2.006, 3.507 and 5.026, respectively, in which the creativity level of excellent learners is the highest and reaches the excellent grade. The normalized gray comprehensive correlation vector (0.3224, 0.3727, 0.3049) is close to the AHP weight vector (0.328, 0.357, 0.315) with a good assessment effect, and the research results provide a new technical path for behavior analysis and creativity development assessment in early childhood programming education.

Keywords: interactive behavior recognition, GCN, creativity assessment, fuzzy comprehensive evaluation, gray correlation analysis

1. Introduction

In the fast-developing intelligent era, artificial intelligence and AI technology have penetrated into

✉ Corresponding author.
E-mail address: suyuanyuan1999@126.com (Y. Su).
Received 12 January 2024; Revised 25 March 2024; Accepted 25 December 2024; Published Online 16 April 2025.
DOI: 10.61091/jcmcc127b-336

every aspect of people's daily life. Programming, as a way for human beings to drive and manipulate intelligent machines, and an important tool for information communication and exchange in a digital society, is regarded by academics as a new literacy for preschool education in the new century, and one of the key skills for young children aged 0-6 years old to actively participate in and build a future digital society [1-4]. Of course, early childhood programming education does not refer to the over-the-top education and mechanical training of programming language and program code writing for young children, but the programming enlightenment education that points to the cultivation of young children's thinking and cognitive ability, and initial programming thinking through the use of appropriate tools and activity design, which arouses their interest in learning programming [5-8].

Some studies have found that not only do young children enjoy engaging in programming activities, but programming education also improves problem solving skills, creativity, and the development of computational thinking in young children, which are considered to be important future-oriented literacies and competencies. First, Çiftci, S. and Bildiren, A. explored the effects of a programming program on children's problem solving and cognitive abilities and found that it was effective in improving children's nonverbal cognitive abilities [9]. Woo, K. and Falloon, G. showed that children's creative coding activities are beneficial in developing their problem solving skills by exploring children's problem solving processes in coding tasks and found that technology-based strategies are not the key to problem solving, thinking about the problem is [10]. Secondly, Murcia, K. et al. examined the meaning and process of fostering creativity exhibited by children when engaged in coding activities, describing the positive impact of coding instruction on children's creativity in terms of agency, curiosity, connection, boldness, and experimentation [11]. Bers, M. U. et al. stated that integrating coding and computational issues into early childhood classroom education helps to develop children's communication, collaboration, and creativity skills [12]. Murai, Y. et al. emphasized that the use of the maker movement and computer science education to develop children's creative learning skills is a current educational trend, and that the integration of creative learning in the school curriculum through programming activities contributes to the development of children's ability to think independently and critically [13]. In addition, Relkin, E. et al. examined the effect of a programming program on children's facilitation of computational thinking skills, noting that children who were taught a programming language showed a prominent advantage in solving specific cognitive activities without a computing device, suggesting that the programming program accelerated children's acquisition of computational thinking skills [14]. Pérez-Marín, D. et al. proposed the use of metaphors and the Scratch tool (MECOPROG) to teach basic programming concepts to children and demonstrated through an investigation that this method helps to develop computational thinking skills in students [15]. Rose, S. et al. compared the extent to which different programming tools influenced the development of young children's computational thinking and showed that there were differences in the enhancement of computational thinking due to the different ways in which young children approached programming projects [16]. Wong, G. K. applied programming activities based on a blended learning model to the process of developing children's computational thinking skills and found that children showed strong thinking skills when facing computational problems, which provided some insights into children's educational work [17]. To sum up, the early childhood programming education program that meets the characteristics of the age stage of young children and attracts their interest in learning not only can be implemented, but also promotes the development of young children's thinking, behavior, ability, emotion and other aspects.

In the process of programming education, young children can easily master various programming instructions and deeply understand the logic of action behind each instruction, which in turn enhances their scientific exploration ability [18-19]. And in the following exploration, they interacted with the programming game and wrote instructions for it according to their own design, and when the program acted accurately according to their instructions, the young children's desire for exploration was further stimulated [20-22]. In addition, teachers provide a platform for young children to create and design freely according to their own wishes, a process that inadvertently enhances their creativity [23-25]. In conclusion, the core of early childhood programming education is not simply learning programming languages or skills, but aims to stimulate young children's logical thinking ability, enhance their imagination, creativity, and practical problem solving ability through the exploratory learning process of programming, so as to promote the overall development of young children [26-29].

In this paper, we first constructed a stack noise reduction self-coding neural network to realize the accurate recognition of human posture. The interaction behavior is modeled by the participant relationship graph, and the Pearson algorithm is used to calculate the shape similarity between nodes. Multi-level feature aggregation based on GCN and feature representation using spatio-temporal words are used to evaluate the performance level of the model based on the cross-validation method. A creativity assessment system was designed to address the problem of creativity assessment in young children's programming game scenarios. Eighty young children aged 3-6 years old were used as research subjects to collect interaction behavior data. K-mean clustering was used to classify the students, and a gray fuzzy comprehensive evaluation model based on AHP was proposed. The weights of the indicators were determined through expert assignment, and the level of creativity was quantitatively assessed based on the fuzzy comprehensive evaluation. The consistency problem between the gray comprehensive correlation results and the AHP weighting results was examined to examine the effectiveness of the evaluation model.

2. Deep Learning Based Interaction Behavior Recognition Model for Early Childhood Programming Games

2.1. Deep learning based pose recognition

The purpose of gesture recognition is to extract information about the participants' body movements, by analyzing the children's movements and then extracting their behavioral characteristics in the game. Gesture is an abstraction and description of the whole human body in an image, and the portrait in each frame belongs to a certain gesture. Each frame of image can be represented by a pose when the 1st layer of recognition is completed. In this paper, a novel stack noise reduction self-coding neural network is constructed to recognize human body poses in images, and the recognition process is shown in Fig. 1.

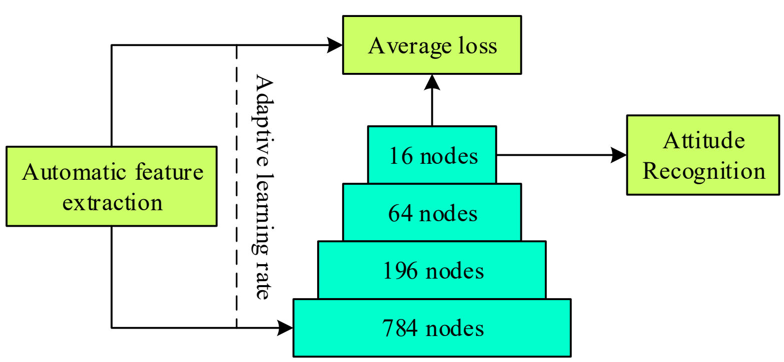


Fig. 1. Attitude recognition process

2.2. Feature extraction and representation based on participant relationship graphs

Young children programming game interaction behavior recognition belongs to the group behavior recognition task, the group behavior recognition task needs to take into account the mutual influence and mutual association between more interaction subjects, and its interaction relationship is more complex and the interaction types are more diverse. In this paper, we propose a participant relationship graph algorithm, which introduces graph convolutional neural network (GCN) into the group behavior recognition task.

The input of the feature extraction sub-algorithm based on graph convolutional network is participant relationship graph, which is a kind of entitled graph, and the weights on the edges are calculated by the interactors' appearance characteristics and distance relationship, and its flow is shown in Fig. 2. In this section, firstly, the construction process of participant relationship graph is elaborated in detail, and compared with the baseline algorithm, this paper introduces a new scheme to measure the similarity of interactors' appearance, and then the graph convolution-based feature extraction subalgorithm is explained.

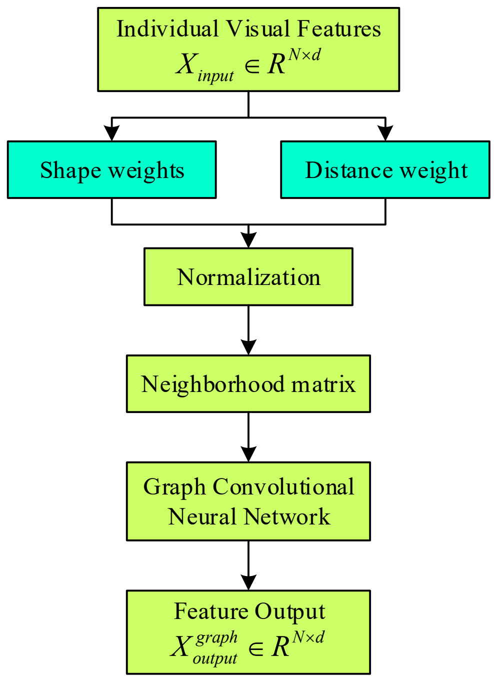


Fig. 2. Feature extraction process based on participant relationship diagram

2.2.1. Participant Relationship Map Construction. The input video sequence, after the convolutional layer, RoIAlign operation and fully connected layer, yields the interactant features with dimension $N \times d$, where N denotes the total number of human bounding boxes in the input video sequence, and d denotes the dimension of the deep visual features of the interacting individuals extracted from each human bounding box region. The set of nodes of the participant relationship graph can be represented as:

$$S = \left\{ x_i^a, x_i^c \mid i = 1, 2, \ldots, N \right\} \tag{1}$$

where $x_i^a \in R^d$ denotes the profile characteristics of the i nd individual, $x_i^c = \left(c_i^x, c_i^y \right)$ denotes the coordinates of the center point of the bounding box of the i th individual. The participant relationship graph is constructed based on the node set S, whose adjacency matrix is denoted as $A \in R^{N\times N}$, A, A Each element of the matrix A_{ij} denotes the weight of an edge between the i th individual and the j th individual, and in particular, $A_{ij} = 0$ denotes the absence of a connection of an edge on the graph between the i th individual and the j th individual.

In order to portray the weights of the connections between interactors, the shape features and coordinate features of the node set are taken into account and finally fused to obtain the weights on the edges between the nodes, which are represented by the following formula:

$$A_{ij} = F\left(f_a\left(x_i^a, x_j^a\right), f_c\left(x_i^c, x_j^c\right)\right) \tag{2}$$

Where $F(\cdot)$ denotes the fusion function, the commonly used function here is $soft\max$, where A_{ij} can be expressed as:

$$A_{ij} = \frac{f_c\left(x_i^c, x_j^c\right)\exp\left(f_a\left(x_i^a, x_j^a\right)\right)}{\sum_{j=1}^{N} f_c\left(x_i^c, x_j^c\right)\exp\left(f_a\left(x_i^a, x_j^a\right)\right)} \tag{3}$$

$f_c\left(x_i^c, x_j^c\right)$ is used to compute the positional weights between the i nd and j rd interactors, in the baseline and its successors distance masking is mostly used by considering only neighboring nodes between the interactors whose L2 paradigm does not exceed the specified threshold, i.e., only neighboring nodes whose Euclidean distances are within the threshold are considered as shown in Eqn. (4), where $I(\cdot)$ stands for the indicator function, which has the value of 1 when and only when the logic in the indicator function is true. The result obtained in all other cases is 0:

$$f_c\left(x_i^c, x_j^c\right) = I\left(L_2\left(x_i^c, x_j^c\right) \le \mu\right) \tag{4}$$

$f_a\left(x_i^\alpha, x_j^a\right)$ function is used to compute the shape weights between the i nd and j rd interactors, and the baseline algorithm uses dot product to compute the shape similarity between the nodes of the participant relationship graph. However, the baseline and absolute error algorithms share the common characteristic that they both only consider the similarity between vectors and lack further consideration of the linear relationship between values. If there is a linear relationship between the data, both methods cannot reflect the real relationship between the vectors well. In order to solve the above problems, this paper uses Pearson's algorithm to calculate the shape similarity between the nodes in the participant's relationship graph, which has certain advantages in the consideration of the linear relationship between the values and insensitivity to the data scale. The value range of Pearson correlation coefficient is $[-1,1]$, and this value will be used to determine the values of the elements in the neighbor matrix of the participant relationship graph together with the distance weights, but in order to avoid meaningless negative weights in the participant relationship graph, only the absolute value of the Pearson coefficient is taken in the calculation of the profile weights, and the formula is expressed as follows:

$$f_a\left(x_i^a, x_j^a\right) = \left|\frac{E\left[\left(x_i^a - \mu_{x_i^a}\right)\left(x_j^a - \mu_{x_j^a}\right)\right]}{\sigma_{x_i^a}\sigma_{x_j^a}}\right| \tag{5}$$

2.2.2. Graph Convolution Based Feature Extraction. The graph structure is jointly determined by the set of nodes and the set of edges, equation (6) gives the definition of the set of nodes of the participant-relationship graph, and equation (3) gives the computation of the features of each edge in the set of edges of the participant-relationship graph, according to which, after determining the structure of the graph, the features on the graph are aggregated using graph convolutional network, and the formula is expressed as follows:

$$Z^{(l+1)} = \mathrm{Re}LU\left(AZ^{(l)}W^{(l)}\right) \tag{6}$$

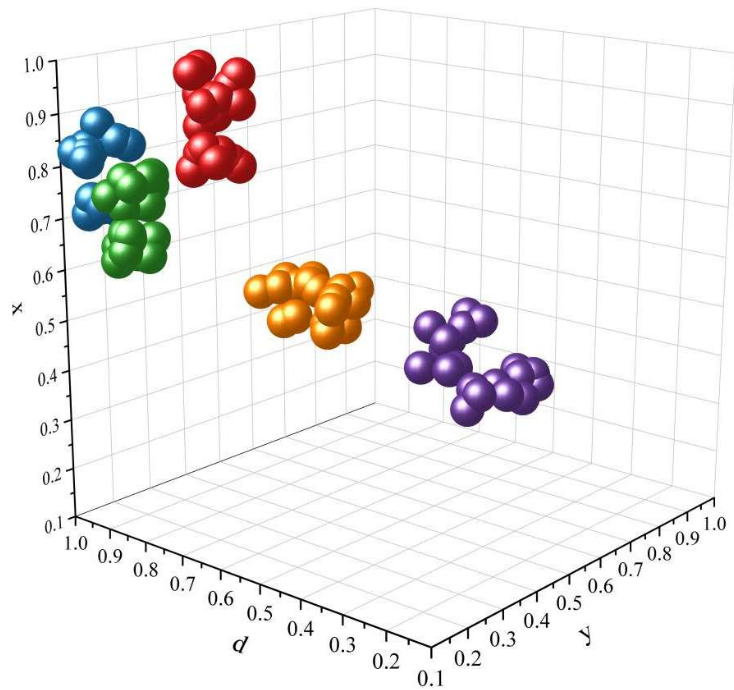
Where $A \in R^{N\times N}$ denotes the adjacency matrix, $Z^{(l)} \in R^{N\times d}$ denotes the features of N nodes within layer l, $R^{N\times d}$ denotes the learnable parameters of the graph convolution in layer l, and after completing the aggregated update of the node features the $ReLU$ activation function is used to obtain the output features of the graph convolution in the current layer, and the output of the current layer, i.e., the input of the next layer, is denoted as $Z^{(l+1)} \in R^{N\times d}$. The final output of the graph convolution-based feature extraction subalgorithm is denoted as $X_{output}^{graph} \in R^{N \times d}$.

From equation (1), it can be seen that the visual features of all interacting subjects of the input video frames are included in the node set when constructing the participant relationship graph, i.e., the temporal features of the interactors are not explicitly extracted within the graph convolution feature extraction subalgorithm.

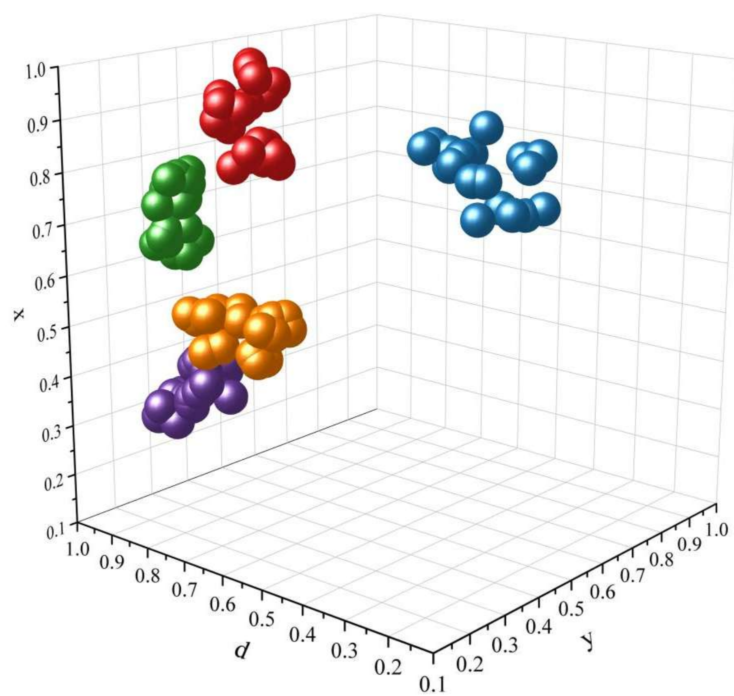
2.2.3. Spatio-temporal word representation. The higher the dimensionality of the feature vector in the behavioral modeling process, the more training data is required. In this section, a concise and effective feature representation is proposed to project the original feature data from the high-dimensional space to the low-dimensional space while trying to keep the key components of the original information not lost.

Each spatio-temporal interest point can be regarded as a point in a three-dimensional space (x, y , d , where d is the measure of the spatio-temporal interest point, and x and y are the positions of the spatio-temporal interest point in the image space), and thus each singleton behavior can be regarded as a collection of spatio-temporal interest points in that three-dimensional space. In this paper, the histogram quantization technique is used to quantize the set of spatio-temporal interest points into a histogram with a fixed number of dimensions (i.e., spatio-temporal words), and the spatio-temporal codebook is generated using the K-means clustering algorithm. Before clustering to generate the codebook, each spatio-temporal interest point is normalized to ensure its scaling and translation invariance. How to evaluate the distance between samples is a key issue in similarity-based clustering algorithms, and since we have treated the spatio-temporal interest points with translation and scaling invariance, the method in this section can directly use the traditional Euclidean distance metric to measure the distance between spatio-temporal interest points, and the distribution of the two spatio-temporal interest points in the first training set of the cross-validation experiments is shown in Fig. 3.

For a given set of spatio-temporal interest points, a soft voting mechanism is used to generate the corresponding spatio-temporal words. The voting controls each spatio-temporal interest point to vote to the few clustering centers closest to it. We place a Gaussian distribution on each clustering center as a way to compute the posterior probability that each spatiotemporal interest point belongs to each clustering center, and the variance of this Gaussian distribution σ is set based on empirical values (in the experiments $\sigma = 0.2$) to ensure that each spatiotemporal interest point will vote to 4-6 clustering centers significantly.



(a)Right side of the human body



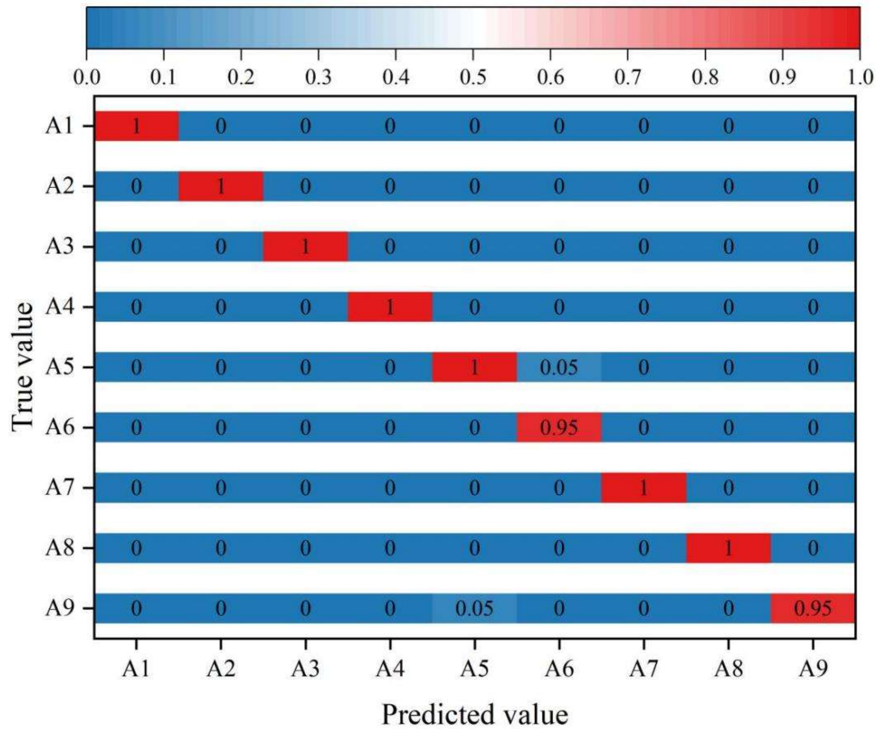
(b)Left side of the human body

Fig. 3. Space-time codebook example

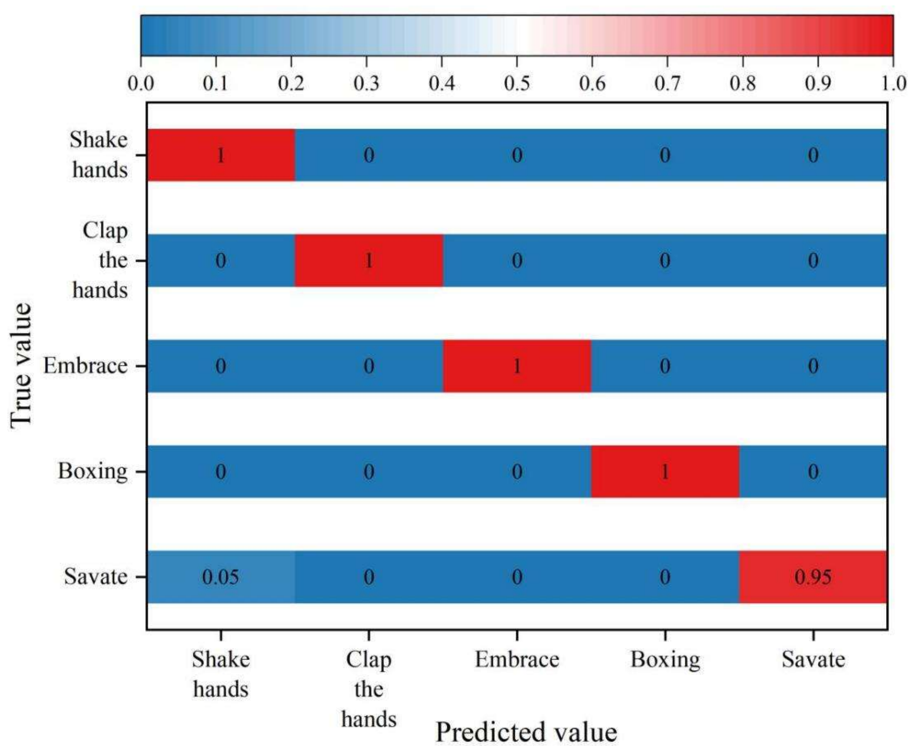
2.3. Validation of effectiveness

2.3.1. Experimental design. There is no open database of interaction behaviors of young children's programming games, in order to evaluate the interaction behavior analysis method proposed in this paper, videos of young children's interaction behaviors while playing programming games were recorded in an indoor scenario to establish an interaction behavior database, and all the video sequences were acquired in a single view using ordinary digital video equipment. The behavior database contains 5 kinds of high-frequency interaction behaviors that occur during children's programming games, namely "shaking hands", "high-five", "hugging", "boxing" and "kicking", and each interaction behavior has 20 behavior samples. The entire behavioral database is equally divided into four parts and cross validation is used to evaluate the recognition method of this paper. The recognition rate of atomic behaviors is highest when the dimension of the codebook is 20, so the number of K-means clustering centers is chosen to be 20.

2.3.2. Experimental results. The recognition of interactive behavior proposed in this paper includes two parts: single-person atomic behavior recognition and two-person interaction semantic analysis, and the recognition results of atomic behavior greatly affect the recognition performance of the final two-person interaction behavior. As shown in Figure 4(a), a total of 9 single-person atomic behaviors with semantic meanings were defined in the experiment, and the average recognition rate of single-person atomic behaviors reached 98.9% due to the strong classification ability of the autoencoder neural network on small sample sets. The confusion matrix of two-person interaction behavior recognition is shown in Figure 4(b), which shows that GCN also shows strong error correction ability in the case of underlying atomic behavior recognition errors thanks to the flexibility of GCN in modeling complex interaction behavior. For example, when the "handshake" behavior of the person on the right side (label A5) is incorrectly recognized as "high-five" (label A6), because the autoencoded neural network correctly recognizes the "handshake" behavior of the person on the left (label A1), and finally the GCN correctly recognizes the interactive behavior of the "handshake" between the two people. It is worth noting that when the "kicking" (label A9) behavior of the person on the right side is incorrectly recognized as the "handshake" (label A5) behavior, although the autoencoder neural network correctly identifies the "avoidance" (label A4) behavior of the person on the left, the GCN still mistakenly identifies the behavior of "shaking hands" between the two people, the reason is that when the left side of the person "avoids" behavior, GCN modeling can have two possible behaviors of the right person at this time, namely "punching" and "kicking", but when the right side person "shakes hands" behavior, GCN believes that there is only one possible behavior for the person on the left, which is "shaking hands", and it is clear that the former is more uncertain than the latter, so GCN misrecognizes.



(a)Single-person atomic action recognition



(b)Two-person interaction behavior recognition

Fig. 4. Confusion matrix

3. Analysis of Interactive Behavior and Assessment of Creativity in Young Children's Programming Games

Under the global education background of digital transformation, it has become an international trend for programming education to develop at a younger age, and it is urgent to quantify the correlation between the interactive behavioral characteristics and the creativity development level of programming games for 0-6 year olds, as an important carrier for the early development of computational thinking. The current study faces a double challenge: on the one hand, it is difficult to accurately capture the dynamic interaction characteristics by traditional observation method; on the other hand, there is a theoretical gap in the existing assessment system in terms of modeling the mapping relationship between the behavioral data and creativity representations. Based on the big data paradigm in education, this paper innovatively introduces the GCN-based interaction behavior recognition model into the field of early childhood education research, and integrates the interaction behavioral features to evaluate the creativity level of young children in programming game activities.

3.1. Determination of the creativity assessment system

The creative thinking of learners in scientific inquiry is reflected in identifying problems, forming hypotheses, discovering new relationships, and solving problems. Mathematics plays a fundamental role in programming play activities for young children. Learners' reasoning and conceptualization to come up with unique and novel solutions to problems is a concrete manifestation of mathematical creativity in the activity. Engineering practice emphasizes functional orientation and the need to achieve specific functions to meet functional needs. Programming games advocate learners to purposefully design products to fulfill specific functional needs within conditional constraints. In programming game activities, expression is one of the means by which learners' ideas are materialized. In the course of learning activities, creative expression highlights individual characteristic styles and can clearly show the level of personal understanding. Words, texts, drawings, video works, etc. are the vehicles that give expression to specific emotions and reveal the sense of self. Creative forms of expression are imaginative and have a certain aesthetic value. The type of expression used and the content of the expression itself are important indicators of the level of creativity of the learner.

The creativity assessment system designed in this paper includes four dimensions: scientific inquiry, mathematical analysis, engineering technology, and artistic expression, which are mainly reflected in the learners' application of knowledge from these disciplines to solve problems in real situations. In these four dimensions, learners are observed in the process of “asking diverse questions”, “flexibly verifying initial ideas”, and “optimizing and forming original designs”. The study combined the characteristics of the subject area with the three processes of creative thinking, and designed evaluation indicators consisting of “three processes and four dimensions”, as shown in Table 1. The evaluation assignments are only 1 or 0, when the student behavior occurs, it is marked as 1 by the recorder, otherwise it is marked as 0.

Table 1. Creativity assessment system

Primary index	Secondary index	Evaluation content
Raise diversity issues(A)	Asking questions involving scientific concepts(A1)	Use scientific concepts or related terms to ask context-appropriate scientific questions(A1-1)
	Expressing problems in mathematical language(A2)	Able to use symbols,formulas,graphs and other diverse mathematical language to express problems(A2-1)
	Raise engineering and technology related issues(A3)	Ability to analyze situations and raise engineering and technical issues that affect the completion of tasks(A3-1)
	Express ideas accurately and imaginatively(A4)	Ability to use metaphors,personification,analogies and other imaginative ways to express ideas or problems(A4-1)
Flexibility to validate initial ideas(B)	Testing the feasibility of scientific solutions(B1)	Follow the scientific operation process to test the plan,verify the idea and select the best solution(B1-1)
	Validation of ideas using mathematical analysis(B2)	Validation of initial ideas using logical reasoning,spatial imagination,data processing,etc(B2-1)
	Verify the feasibility of programming design(B3)	Ability to quickly build simplified models or programs to verify the feasibility of the initial design(B3-1)
	Visualization of design intent(B4)	Flexible use of sketches, drawings, pictures and other ways to present design intentions(B4-1)
Optimize for original design(C)	Optimizing Scientific Experiment Design(C1)	Optimize the design of experiments based on initial ideas to reflect students' deep understanding of scientific knowledge(C1-1)
	Optimize the structure of your work(C2)	Ability to go beyond established logic and consider more variables to optimize program design or model structure(C2-1)
	Optimized functionality(C3)	Improve models or programs to better simulate real-world situations,to be more functional and interactive(C3-1)
	Creative presentation of results(C4)	Ability to demonstrate programs in creative ways such as creating stories,performances,competitions,etc(C4-1)

3.2. Analysis of creativity level based on learners' behavioral characteristics

3.2.1. Objects of study. This study is based on a programming game curriculum for young children aged 0-6 years old, and 80 students aged 0-6 years old were recruited through the Children's Palace in City A. A project-based approach was used to create a learning and research situation. The basic information of the research subjects is shown in Table 2. The curriculum designs project activities around visual programming and other contents, and the learning activities include the introduction of real scenarios and problem analysis, program writing, work optimization and sharing sessions. According to the differences in the students' knowledge base, they were divided into two types: basic class and improvement class. There were 38 students in the basic class with an average age of 4.6 years old, and these students had no exposure to programming content before the start of the course. There were 42 students in the advanced class with an average age of 5.3 years old, and the students in the advanced class knew some programming basics (e.g., scratch). The programming course of study lasted 5 weeks for a total of 10 sessions, with each activity lasting approximately 90 minutes, and data on the process of each student completing the task were observed and recorded.

Table 2. Basic information of research objects

Variable	Category	Sample size	Proportion
Age	3 years old	15	18.75%
	4 years old	22	27.5%
	5 years old	26	32.5%
	6 years old	17	21.25%
Gender	Male	49	61.25%
	Female	31	38.75%
Class	Elementary class	38	47.5%
	Promotion class	42	52.5%

3.2.2. Data analysis. Based on the data collected by the human interaction behavior recognition model proposed in this paper, the K-mean clustering algorithm was used to cluster and analyze the data of students' performance in programming game activities, and the students were classified into three categories. Based on the characteristics of the data on each dimension, they were defined as hidden type (36 students), presented type (22 students), and excellent type (22 students), and the specific classification results are shown in Table 3.

A significant number of students belonged to the hidden type, including 30% in the basic class and 15% in the improvement class. These students performed below average creatively in the vast majority of dimensions. For the task requirements, they were able to formulate one or two basic problems. During the problem solving process, their design solutions were occasionally bright. After completing the basic task, these students are able to make some random, localized improvements to the program and model, but are unable to clearly articulate the intent of the design. Taken together, this group of learners showed weak creativity in programming game activities. Factors such as the students' mental maturity, mastery of the basics, and adaptability to the course activities may affect their creativity performance.

Presentational students (7.5% of the basic class and 20% of the advanced class) often come up with creative ideas and have strong problem solving skills. This group of students performed more prominently in problem identification, use of mathematical knowledge to carry out solution validation, model and procedure optimization, and creative expression. However, the creative performance of this group of students usually does not go beyond the scope of the project tasks agreed upon, and the learning behaviors as well as the work created are characterized by sophistication. Overall, these students are excellent learners with a certain degree of creativity.

The creative potential of the outstanding learners (7.5% in the basic class and 20% in the advanced class) is characterized by divergent, fluent and innovative thinking. Children in this category were able to actively ask diverse scientific questions in all task sessions and analyze and validate creative designs using methods of scientific inquiry and mathematical knowledge. They are able to regularly incorporate personal creative ideas into their programming designs, and the process of program writing better reflects the innovative nature of students' thinking. Children in this category tend to go beyond the requirements of the task itself in the learning process, demonstrating courageous experimentation, rigorous thinking and a high incidence of creative behavior.

Table 3. Characteristics of learners' creative behavior in different categories(M±SD)

	Looming type(N=36)	Presentation type(N=22)	Outstanding type(N=22)	Average(N=80)
A1	1.54±0.96	5.43±1.28	5.77±1.23	3.77±2.13
A2	0.66±0.56	1.68±0.94	3.72±1.06	1.78±1.46
A3	0.75±0.93	2.58±1.01	5.73±1.35	2.62±1.64
A4	3.55±1.08	3.26±0.85	5.06±1.12	3.89±2.04
B1	1.57±0.74	4.06±2.03	3.99±1.58	2.92±1.45
B2	0.33±0.55	0.17±0.38	0.85±0.74	0.43±0.95
B3	1.55±0.83	4.24±0.99	6.36±1.13	3.61±2.47
B4	0.68±0.85	1.04±0.94	3.69±0.93	1.61±0.94
C1	0.48±0.67	0.84±0.92	1.95±0.86	0.98±0.67
C2	4.88±1.46	5.97±1.85	7.38±1.15	5.87±1.58
C3	1.43±0.97	2.58±1.75	5.88±1.36	2.97±1.40
C4	0.68±0.96	1.85±1.04	3.04±0.97	1.65±1.67

3.3. AHP-based gray fuzzy comprehensive evaluation of creativity

3.3.1. Gray correlation analysis. The gray correlation analysis method is usually calculated by using the degree of correlation to indicate the degree of influence of one factor on another factor. The calculation steps of the gray correlation analysis method are as follows:

1) Determine the reference sequence and comparison sequence. Select the relevant data to be categorized, the most representative data as reference data, and other data as comparison data:

Set the reference sequence: $X_0 = \{X_0(k), k = 1,2,\cdots,n\}$

Set the comparison sequence: $X_i = \{X_i(k), k = 1,2,\cdots,n, i = 1,2,\cdots,m\}$

2) Dimensionless processing. In order to make the data unaffected by the dimension, the selected data are processed without dimension, and the formula is:

$$X_i' = \frac{X_i}{X1}\{X_i'(1), X_i'(2), \cdots, X_i'(k)\} \tag{7}$$

3) Find the absolute value, the maximum difference and the minimum difference. The absolute value is calculated by the formula:

$$\Delta_i = \left|X_0'(k) - X_i'(k)\right|, k = 1,2,\cdots,n \tag{8}$$

Where the largest absolute value is the maximum difference and the smallest absolute value is the minimum difference.

4) Find the correlation coefficient, calculated by the formula:

$$\zeta_i(k) = \frac{\min_i \min_k \Delta_i(k) + \beta \max_i \max_k \Delta_i(k)}{\Delta_i(k) + \beta \max_i \max_k \Delta_i(k)}, 0 < \beta < 1 \tag{9}$$

Where, $\min_i \min_k \Delta_i(k)$ is the minimum difference, $\max_i \max_k \Delta_i(k)$ is the maximum difference, $\beta \in [0,1]$ is the discrimination coefficient, generally take $\beta = 0.5$.

5) Calculate the degree of association, the formula is:

$$r_i = \frac{1}{n}\sum_{k=1}^{n}\zeta_i(k) \tag{10}$$

Correlation refers to the degree of correlation between two factors, the greater the r, the stronger the correlation between the two factors, the closer the relationship, and vice versa, the lower the degree of correlation between the two factors, the less close the relationship.

3.3.2. Determination of AHP-based indicator weights. Fuzzy comprehensive evaluation method is a comprehensive evaluation method based on fuzzy mathematics. The method transforms qualitative evaluation into quantitative evaluation according to the theory of affiliation degree of fuzzy mathematics, i.e., it uses fuzzy mathematics to make an overall evaluation of things or objects constrained by multiple factors. In the real evaluation, many objective facts can not be accurately answered by yes or no, because they are affected by a series of problems such as the complexity and fuzzy uncertainty of the evaluation factors, the multi-level nature of the evaluation object, etc., and we can't use the absolute qualitative or quantitative method to judge, and the fuzzy comprehensive evaluation method solves this problem well.

Establish the creativity evaluation model based on AHP with the following steps:

1) Establish the set of evaluation subjects and the set of weight coefficients of evaluation subjects.

2) Establish the factor set of the evaluation model according to the creativity evaluation index system.

3) Construct the rubric set V={V1, V2, V3, V4, V5}={excellent, good, better, pass, fail}.

According to the steps of AHP, experts are invited to score the relative importance between the indicators of the creativity assessment system, and then summarize and calculate through EXCEL to derive the weight of each indicator as shown in Table 4.

Table 4. Weight index of indicators

Objective	Primary index	Weight coefficient	Secondary index	Weight coefficient
Creativity level	A	0.328	A1	0.078
			A2	0.112
			A3	0.037
			A4	0.101
	B	0.357	B1	0.132
			B2	0.048
			B3	0.142
			B4	0.035
	C	0.315	C1	0.031
			C2	0.167
			C3	0.066
			C4	0.051

3.3.3. Fuzzy integrated evaluation. A rubric set was constructed, stipulating that a score of 2 or less is failing, 2 to 3 is passing, 3 to 4 is better, 4 to 5 is good, and a score of 5 or more is excellent.

The creativity levels of the three categories of students were evaluated and the final results are shown in Table 5. The comprehensive evaluation scores of the three types of students are 2.006, 3.507, and 5.026 respectively, and all students' creativity reaches the passing level, among which the excellent learners have the highest level of creativity and reach the excellent level.

Table 5. Results of AHP creativity evaluation

	Looming type(N=36)	Presentation type(N=22)	Outstanding type(N=22)
A	0.580	1.036	1.590
B	0.467	1.183	1.600
C	0.959	1.288	1.836
Comprehensive evaluation score	2.006	3.507	5.026
Comprehensive evaluation level	Passing	Preferable	Excellent

3.3.4. Gray correlation analysis of evaluation results. The test results were validated by applying the gray correlation method, and in order to illustrate the gray analysis process of this problem, limited to space, two students were randomly selected from each of the three types of students, and these students were renumbered sequentially from 1 to 6, and the scores on each sub-indicator and the comprehensive evaluation scores of these six students were extracted. Since the purpose of the study is to examine the correlation between the score results of each sub-indicator and the comprehensive evaluation results, the sequence composed of the comprehensive evaluation scores of the six selected students is taken as the reference series, and the sequence composed of the scores on each sub-indicator of the six students is the comparison series, so as to constitute a gray system structure containing three comparison series. The comparative and reference series are represented as follows. Where S_{ij} ($i=1:6$, $j=1:3$, S represent scores) represents the scores on the j th sub-indicator of the i th work.

The corresponding comparative columns for sub-indicator A are: $X_1=(S_{11},S_{21},S_{31},S_{41},S_{51},S_{61})$

The comparative columns corresponding to sub-indicators B are: $X_2=(S_{12},S_{22},S_{32},S_{42},S_{52},S_{62})$

The comparative series corresponding to sub-indicator C is denoted as: $X_3=(S_{13},S_{23},S_{33},S_{43},S_{53},S_{63})$

The reference series corresponding to the comprehensive evaluation scores are denoted as: $X_0=(S_1,S_2,S_3,S_4,S_5,S_6)$

Where $S_{ij}\left(i=1:8\right)$ denotes the comprehensive evaluation score of work No. i.

The next step is to further quantitatively analyze the degree of correlation between each comparison series and the reference series.

Firstly, the initialization operator is utilized:

$$\frac{1}{X_{,}(1)}(i=0,1,2,3) \tag{11}$$

Find the initial value of each sequence like:

$$X_i{}'(j)=\frac{X_i(j)}{X_i(1)},j=1,2,\cdots,6 \tag{12}$$

The initial value like results for each sequence are shown in Table 6.

Table 6. Initial values of each sub-index score

	1	2	3	4	5	6
A	1	0.89	0.6	0.6	0.78	0.71
B	1	0.81	0.58	0.58	0.27	0.49
C	1	0.79	0.55	0.55	0.5	0.5
Reference series	1	0.798	0.571	0.571	0.575	0.603

The formula for taking $r_i(j)$, the resolution coefficient λ is taken as 0.5, and the correlation coefficient on each sub-indicator of the work can be obtained by calculation, and the correlation coefficients of each comparative series and the reference series are shown in Table 7. From the correlation coefficients between the comparison series and the reference series in Table 7, it can be seen that the correlation coefficients between the scores of students No. 1, No. 2 and No. 6 in subindicators A, B and C and the composite score are all 1. The values of the correlation coefficients indicate that the comparison series has the highest correlation with the reference series in the range of 1, 2 or 6, which are the strongest connections.

Table 7. Correlation coefficients of each comparison series and reference series

	1	2	3	4	5	6
A	1	1	0.8200	0.7465	0.8200	1
B	1	1	0.8200	0.6973	0.8200	1
C	1	1	0.7935	0.8486	0.7945	1

The correlation between the comparison series and the reference series can be obtained by the various calculation methods corresponding to the correlation degree, and the specific results are shown in Table 8. From Table 8, it can be found that the gray correlation between the three comparative series and the reference series in the order from large to small is comparative series 2, comparative series 1, comparative series 3. The relative correlation of the three comparative series, the absolute correlation as well as the integrated correlation degree of the size of the order of consistency, are comparative series 2, comparative series 1, comparative series 3.

Table 8. Correlation degree between comparison series and reference series

	Grey correlation degree	Relative correlation degree	Absolute correlation degree	Comprehensive correlation degree
A	0.8754	0.9836	0.7397	0.8456
B	0.8974	0.9975	0.8836	0.8675
C	0.8600	0.9744	0.7244	0.8074

In view of the fact that there are some differences between different types of correlations and the fact that the composite correlation reflects more adequate information, the following study adopts the composite correlation as an indicator to measure the correlation between the factors.

Comparison Series 1, Comparison Series 2 and Comparison Series 3 are sequences consisting of students' scores on sub-indicators A, B and C, respectively. Based on the method of obtaining the gray composite correlation, the composite correlation between the three comparison series and the reference series is 0.8456, 0.8675, and 0.8074, respectively. It can be seen that the combined correlation between the three comparison series and the reference series is, in descending order, Comparison Series 2, Comparison Series 1, and Comparison Series 3, indicating that Indicator B has a greater correlation with the composite scores, and Indicators A and C have a lesser correlation with the composite scores.

The integrated correlation vector (0.8456, 0.8675, 0.8074) was normalized to obtain the normalized vector (0.3224, 0.3727, 0.3049). While the corresponding weight vector of each indicator determined by AHP is (0.328, 0.357, 0.315), the normalized vector obtained after processing by fuzzy mathematics and grey correlation methods is close to the weight vector, thus indicating the effectiveness of fuzzy comprehensive evaluation on creativity. The results of the corroboration of the rationality of the fuzzy comprehensive evaluation system relying on the AHP basis through the gray correlation analysis also reflect the feasibility of the gray correlation analysis method to solve the problem, and there is even more room for application.

4. Conclusion

In this paper, students are divided into three categories based on their interactive behavioral characteristics, and the creativity of the three types of students is assessed using AHP-based grey fuzzy comprehensive evaluation of creativity.

The comprehensive evaluation scores of the three types of students are 2.006, 3.507, and 5.026, respectively, and all students' creativity reaches the passing level, among which the excellent type of learners have the highest level of creativity, which reaches the excellent level. Based on the method of grey comprehensive correlation, the comprehensive correlation between the three comparison series and the reference series respectively is 0.8456, 0.8675, 0.8074. The comprehensive correlation vectors after normalization (0.3224, 0.3727, 0.3049) are very close to the weight vectors corresponding to each indicator determined by AHP (0.328, 0.357, 0.315), and the weight vectors corresponding to each indicator determined by AHP are very close to each other. 0.315) are very close to each other, thus demonstrating the effectiveness of evaluating creativity using fuzzy comprehensive evaluation.

Funding

This work was supported by Humanities and Social Sciences Research Project of Jilin Provincial Department of Education: Research on the Construction of Childcare Service System for 0-3 Years Old Children (JJKH20220417SK).

References

[1] Bati, K. (2022). A systematic literature review regarding computational thinking and programming in early childhood education. Education and Information Technologies, 27(2), 2059-2082.

[2] Kraleva, R., Kralev, V., & Kostadinova, D. (2019). A Methodology for the Analysis of Block-Based Programming Languages Appropriate for Children. J. Comput. Sci. Eng., 13(1), 1-10.

[3] Gomes, T. C. S., Falcão, T. P., & Tedesco, P. C. D. A. R. (2018). Exploring an approach based on digital games for teaching programming concepts to young children. International journal of child-computer interaction, 16, 77-84.

[4] Kaplancali, U. T., & Demirkol, Z. (2017). Teaching coding to children: A methodology for kids 5+. International Journal of Elementary Education, 6(4), 32-37.

[5] Critten, V., Hagon, H., & Messer, D. (2022). Can pre-school children learn programming and coding through guided play activities? A case study in computational thinking. Early Childhood Education Journal, 50(6), 969-981.

[6] Chiazzese, G., Fulantelli, G., Pipitone, V., & Taibi, D. (2018). Engaging primary school children in computational thinking: Designing and developing videogames. Education in the Knowledge Society (EKS), 19(2), 63-81.

[7] Angeli, C., & Giannakos, M. (2020). Computational thinking education: Issues and challenges. Computers

in human behavior, 105, 106185.

[8] Allsop, Y. (2019). Assessing computational thinking process using a multiple evaluation approach. International journal of child-computer interaction, 19, 30-55.

[9] Çiftci, S., & Bildiren, A. (2020). The effect of coding courses on the cognitive abilities and problem-solving skills of preschool children. Computer science education, 30(1), 3-21.

[10] Woo, K., & Falloon, G. (2022). Problem solved, but how? An exploratory study into students' problem solving processes in creative coding tasks. Thinking Skills and Creativity, 46, 101193.

[11] Murcia, K., Pepper, C., Joubert, M., Cross, E., & Wilson, S. (2020). A framework for identifying and developing children's creative thinking while coding with digital technologies. Issues in Educational Research, 30(4), 1395-1417.

[12] Bers, M. U., González-González, C., & Armas–Torres, M. B. (2019). Coding as a playground: Promoting positive learning experiences in childhood classrooms. Computers & Education, 138, 130-145.

[13] Murai, Y., Ikejiri, R., Yamauchi, Y., Tanaka, A., & Nakano, S. (2023). Design strategies to integrate creative learning in elementary school curricula through computer programming activities. Interactive Learning Environments, 31(7), 4698-4712.

[14] Relkin, E., de Ruiter, L. E., & Bers, M. U. (2021). Learning to code and the acquisition of computational thinking by young children. Computers & education, 169, 104222.

[15] Pérez-Marín, D., Hijón-Neira, R., Bacelo, A., & Pizarro, C. (2020). Can computational thinking be improved by using a methodology based on metaphors and scratch to teach computer programming to children?. Computers in Human Behavior, 105, 105849.

[16] Rose, S., Habgood, M. J., & Jay, T. (2017). An exploration of the role of visual programming tools in the development of young children's computational thinking. Electronic journal of e-learning, 15(4), pp297-309.

[17] Wong, G. K. (2024). Amplifying children's computational problem-solving skills: A hybrid-based design for programming education. Education and Information Technologies, 29(2), 1761-1793.

[18] Bers, M. U. (2018). Coding and computational thinking in early childhood: The impact of ScratchJr in Europe. European Journal of STEM Education, 3(3), 8.

[19] Hufad, A., Faturrohman, M., & Rusdiyani, I. (2021). Unplugged coding activities for early childhood problem-solving skills. Jurnal Pendidikan Usia Dini, 15(1), 121-140.

[20] Su, Y. S., Shao, M., & Zhao, L. (2022). Effect of mind mapping on creative thinking of children in scratch visual programming education. Journal of Educational Computing Research, 60(4), 906-929.

[21] Canbeldek, M., & Isikoglu, N. (2023). Exploring the effects of "productive children: coding and robotics education program" in early childhood education. Education and Information Technologies, 28(3), 3359-3379.

[22] Wong, G. K. W., & Cheung, H. Y. (2020). Exploring children's perceptions of developing twenty-first century skills through computational thinking and programming. Interactive Learning Environments, 28(4), 438-450.

[23] Strawhacker, A., & Bers, M. U. (2019). What they learn when they learn coding: Investigating cognitive domains and computer programming knowledge in young children. Educational technology research and development, 67, 541-575.

[24] Çakır, R., Korkmaz, Ö., İdil, Ö., & Erdoğmuş, F. U. (2021). The effect of robotic coding education on preschoolers' problem solving and creative thinking skills. Thinking Skills and Creativity, 40, 100812.

[25] Noh, J., & Lee, J. (2020). Effects of robotics programming on the computational thinking and creativity of elementary school students. Educational technology research and development, 68(1), 463-484.

[26] Yang, W., Ng, D. T. K., & Su, J. (2023). The impact of story-inspired programming on preschool children's computational thinking: A multi-group experiment. Thinking skills and creativity, 47, 101218.

[27] Moraiti, I., Fotoglou, A., & Drigas, A. (2022). Coding with Block Programming Languages in Educational Robotics and Mobiles, Improve Problem Solving, Creativity & Critical Thinking Skills. International Journal of Interactive Mobile Technologies, 16(20).

[28] Denner, J., Campe, S., & Werner, L. (2019). Does computer game design and programming benefit children? A meta-synthesis of research. ACM Transactions on Computing Education (TOCE), 19(3), 1-35.

[29] Papadakis, S. (2022). Apps to promote computational thinking and coding skills to young age children: A pedagogical challenge for the 21st century learners. Educational Process: International Journal (EDUPIJ), 11(1), 7-13.