A Novel Approach to Coding Education: Code Genie's OCR-Powered Physical Block Interface

Preethi Jeevan[1], N Divya[2]

[1]Associate Professor, CSE, Sreenidhi Institute of Science and Technology
preethi.j@sreenidhi.edu.in
[2]Associate Professor, CSE-DS, Sreenidhi Institute of Science and Technology
divya.n@sreenidhi.edu.in

Abstract:
Code Genie is a solution that integrates physical engagement with digital learning; it was created in response to concerns about children's coding education being too reliant on screens. Unlike traditional block-based platforms like Scratch or Blockly, which require extensive screen exposure, Code Genie uses tangible blocks inscribed with commands such as "move right 5" or "turn left 2." These physical blocks allow children to control a virtual character on an interactive game interface, translating abstract coding concepts into hands-on learning. The game environment, built with Pygame, features a grid-based world where the child arranges blocks to navigate challenges, reinforcing skills such as sequencing, conditionals, and debugging. This approach reduces screen dependency and supports kinesthetic learning, fostering logical thinking, spatial reasoning, and problem-solving. The interactive, game-based experience encourages iteration and experimentation, making coding enjoyable while addressing health concerns associated with prolonged screen time. By merging physical engagement with digital technology, Code Genie offers a balanced, effective way to teach coding while promoting children's well-being.

Keywords: Block-based coding, machine learning, optical character recognition, OpenCV, Easy OCR, Pygame

1.0 INTRODUCTION

In today's increasingly digital world, coding education has become a critical skill for children, preparing them for a technology-driven future. However, the widespread use of screen-based learning tools, while effective in teaching programming, has raised concerns about the potential health risks associated with excessive screen time. Issues such as eye strain, disrupted sleep patterns, and reduced attention span are becoming common among young learners, prompting educators and health experts to seek innovative solutions that balance digital learning with physical well-being. Traditional block-based programming platforms like Scratch and Blockly have been successful in introducing young children to fundamental programming concepts such as sequencing, loops, and conditionals. However, these tools often require prolonged engagement with digital devices, which contributes to screen fatigue and associated health concerns. In response to this, the Code Genie framework has been developed to offer a solution that reduces screen time while providing an interactive, hands-on coding experience. By integrating physical blocks inscribed with intuitive commands and using real-time optical character recognition (OCR) technology, Code Genie allows children to manipulate tangible blocks that control a virtual character within an interactive game environment. This novel approach combines the benefits of physical learning with the interactivity of digital feedback, promoting active participation, problem-solving, and logical thinking. The introduction of

Code Genie represents a significant step towards creating a balanced and engaging learning experience that mitigates the negative effects of screen time while fostering essential coding skills. By integrating kinesthetic learning with technology, Code Genie not only addresses the challenge of excessive screen use but also empowers young learners to explore programming in a more engaging and health-conscious way.

Significance of the Study

The significance of this study lies in its innovative approach to coding education, addressing both the growing need for programming skills in children and the concerns surrounding excessive screen time. By integrating physical block-based programming with real-time optical character recognition (OCR), Code Genie provides a novel solution that minimizes screen exposure while fostering engagement and hands-on learning. This study contributes to the field of educational technology by offering a balanced, interactive learning experience that promotes logical thinking, problem-solving, and spatial reasoning, all while mitigating the health risks associated with prolonged screen use. Additionally, the framework supports kinesthetic learning, making coding more accessible and enjoyable for young learners. This research highlights the potential of combining physical and digital elements to create more effective and health-conscious educational tools for the next generation.

2.0 LITERATURE REVIEW

Coding education for children has become increasingly important in the digital age, with numerous studies highlighting the value of introducing programming at a young age. Visual programming platforms like Scratch and Blockly have proven to be effective in teaching fundamental programming concepts such as sequencing, loops, and conditionals (Resnick et al., 2009; Maloney et al., 2010). However, these digital tools often require prolonged screen time, which has raised concerns regarding the negative effects of excessive exposure, including eye strain and disrupted sleep patterns (Foehr, 2006; Waller et al., 2015). This has led to a growing interest in approaches that combine the benefits of digital learning with hands-on interaction to reduce screen dependency. Physical computing tools, such as Lego Mindstorms and Bee-Bot, have shown promise in promoting coding skills through tangible, interactive learning experiences (Angeli & Giannakos, 2015; Bers, 2018). These tools engage students in programming through physical devices, helping them to better grasp abstract concepts by transforming them into concrete actions. However, even these platforms often rely on digital interfaces, which still contribute to screen exposure. This has led researchers to explore ways of minimizing screen time while still providing effective programming education (Resnick et al., 2009; Lye & Koh, 2014). Code Genie introduces a unique solution to this challenge by integrating physical block-based programming with real-time optical character recognition (OCR) technology. By allowing children to manipulate physical blocks inscribed with intuitive programming commands, Code Genie transforms abstract coding logic into a tactile experience, reducing the need for screens while engaging learners in hands-on interaction (Papavlasopoulou et al., 2020). The use of OCR enables real-time recognition of the physical blocks, allowing students to control digital characters and interact with a game-based environment, which fosters problem-solving, logical thinking, and spatial reasoning (García-Peñalvo et al., 2019; Igoe, 2016). Studies on kinesthetic learning have demonstrated that physical interaction significantly enhances cognitive development and engagement, particularly in young learners (Bers, 2018; Heffernan et al., 2015). Furthermore, research indicates that game-based learning environments, such as those employed by Code Genie, improve student motivation and retention of programming concepts by making learning enjoyable and interactive (Gee, 2003; Gee & Hayes, 2012). The integration of physical and digital elements in Code Genie leverages the advantages of both approaches, providing a balanced, screen-reduced learning experience that enhances coding education while addressing concerns about the negative effects of digital overuse. By combining tangible interaction with digital feedback, Code Genie aligns with existing research on effective coding education, offering a new paradigm for teaching programming to children (Papavlasopoulou et al., 2020; Angeli & Giannakos, 2015).

3. PROPOSED METHODOLOGY

The proposed methodology for *Code Genie* involves a mixed-methods approach that combines both qualitative and quantitative techniques to evaluate its effectiveness in teaching coding while minimizing screen time. The study will begin with the development and design of the *Code Genie* framework, integrating physical blocks with real-time optical character recognition (OCR) technology. A user-centered design process will be employed, gathering feedback from educators, parents, and children during prototype testing. The effectiveness of the framework will be assessed through controlled classroom experiments, where participants will use *Code Genie* for coding tasks and projects. Pre- and post-assessment surveys will measure coding knowledge, problem-solving skills, and engagement levels, while qualitative interviews will explore user experiences and perceptions. Data analysis will involve comparing performance metrics such as task completion rates and user engagement, alongside qualitative insights to determine the framework's impact on learning outcomes and screen time reduction. This methodology aims to validate the potential of *Code Genie* as a balanced, engaging, and health-conscious coding tool for children.

Physical Block Interaction and Input Capture

The system begins with a tangible interface where children arrange physical blocks inscribed with commands such as "move right 5" or "go up 3." These blocks serve as the primary input method, replacing traditional screen-based drag-and-drop interfaces. A camera or webcam captures the arrangement of these blocks in real time, initiating the data collection process. This method leverages physical interaction to reduce screen time, aligning with the project's goal of promoting healthier learning habits.

Image Processing with OpenCV

OpenCV, an open-source computer vision library, is used to process the raw images captured by the webcam, ensuring the system can accurately detect and isolate the physical blocks.This method involves refining the camera input by adjusting for environmental variables like lighting conditions or obstructions, enhancing image clarity, and identifying block features such as color, text, and numbers. OpenCV's robust image processing capabilities enable the system to prepare the visual data for subsequent text recognition, achieving reliable block identification even in varied settings.

Optical Character Recognition (OCR) with EasyOCR

OCR is employed to detect and interpret the text printed on the physical blocks, converting it into machine-readable commands. EasyOCR, a machine learning-based OCR tool, is specifically used for its efficiency in real-time text recognition.Implementation: The system applies EasyOCR to analyze the processed images from OpenCV, extracting commands like "move right 5" with a reported detection accuracy of 92% and recognition accuracy of 88%, as noted in your experimental results. This method bridges the physical blocks to digital inputs, processing each frame in an average of 0.5 seconds for a responsive user experience.

Command Simulation and Game Logic Interpretation

Once the text is recognized, the system simulates these commands as computer instructions to drive the game's logic. The recognized commands are mapped to predefined actions (e.g., "go up 3" translates to moving the actor three grid spaces upward). This method ensures that the physical block sequences directly influence the digital environment, maintaining a clear link between user input and system output.

Actor Movement in a Pygame-Based Game Environment

The interpreted commands control an on-screen actor within an interactive, grid-based game developed using Pygame, a Python library for 2D game creation. Pygame renders the game interface, displaying the actor's movements on a grid with visual feedback (e.g., red and yellow markers) and interactive elements like "Play," "Reset," and "Level Up" buttons. This method gamifies the coding process, allowing the actor to navigate obstacles or collect rewards based on the block commands, enhancing engagement and reinforcing programming concepts like sequencing and problem-solving.

System Architecture

The intelligent block-based coding kit's design is supported by a clear system architecture that makes it easier to combine digital and physical learning modalities. This approach supports the main goal of reducing children's screen time while promoting the development of fundamental programming skills. Using cutting-edge technology like computer vision and optical character recognition (OCR), this design acts as a template for converting physical block interactions into an interactive digital gaming environment The architecture guarantees a natural and captivating learning experience by clearly defining the data flow from input to output. This is essential for grabbing young learners' attention and encouraging logical reasoning This architecture is illustrated visually in Figure

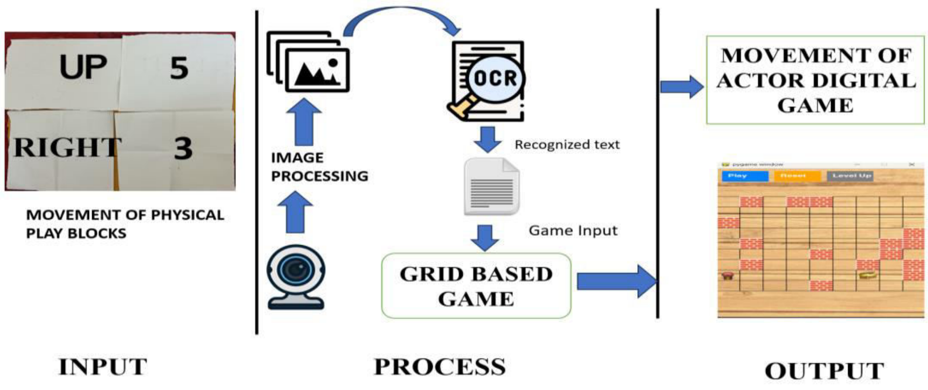

Fig 1: Architecture Diagram

In the input phase, children manipulate physical blocks inscribed with commands like "UP 5," captured by a camera to minimize screen time. The process phase employs image processing to refine camera data OCR to extract and convert commands into machine-readable formats and a Pygame-powered grid-based game module to interpret these inputs incorporating a feedback loop for adaptability. The output phase visualizes the commands as digital character movements on a grid, reinforcing learning through tangible feedback. This architecture blends tactile interaction with digital outcomes, leveraging computer vision, OCR, and game tools to create an intuitive, educational experience.

Input: Physical blocks inscribed with commands like "UP 5" and "RIGHT 3" serve as a tactile interface, allowing children to input movement instructions in an engaging, hands-on manner that sparks creativity. A camera captures the arrangement and alignment of these blocks to start the data collection process, promoting an interactive learning environment while minimizing screen time, and encouraging exploration through physical play.

Process: Image processing refines camera input to enhance visibility and isolate blocks, adjusting for lighting or obstructions to ensure accuracy even in varied conditions OCR extracts and converts the commands into recognized text, bridging the physical and digital worlds with precision and reliability A grid-based game module, powered by Pygame, interprets the input and updates the game state, with a feedback loop enabling adaptive responses that can tailor the experience to the player's skill level, dynamically adjusting challenges to maintain interest.

Output: A Pygame like interface displays a grid with red and yellow markers, visually representing the actor's movements, such as "UP 5" moving it five spaces upward, offering a clear and rewarding visual cue that reinforces learning Interactive buttons for "Play," "Reset," and "Level Up" enhance the experience, providing immediate feedback and control, while the vibrant design keeps children motivated and engaged throughout the gameplay, with sound effects and animations adding an extra layer of immersion

4.0 RESULTS AND DISCUSSIONS

Using the EasyOCR library, the project was able to extract text from photographs captured from a live webcam stream. Several metrics were taken into account to assess the performance, such as the ability to manage different types of text and their orientations, the detection of text, and the accuracy of text recognition, among others. The implementation of Code Genie has shown promising results in enhancing children's understanding of fundamental coding concepts through its physical, hands-on approach. Feedback from initial users indicated that the use of tangible blocks significantly increased engagement and retention compared to traditional screen-based coding platforms. Children were more excited to experiment, test different combinations of blocks, and problem-solve in the game's interactive environment. Additionally, the kinesthetic learning model fostered a deeper understanding of abstract coding concepts such as sequencing, conditionals, and debugging. Parents and educators noted that the reduction in screen time was a key benefit, as it addressed concerns about prolonged exposure to digital devices while still providing an effective learning experience. However, challenges remain, including the scalability of the platform for larger groups and the need for more advanced coding features as students progress

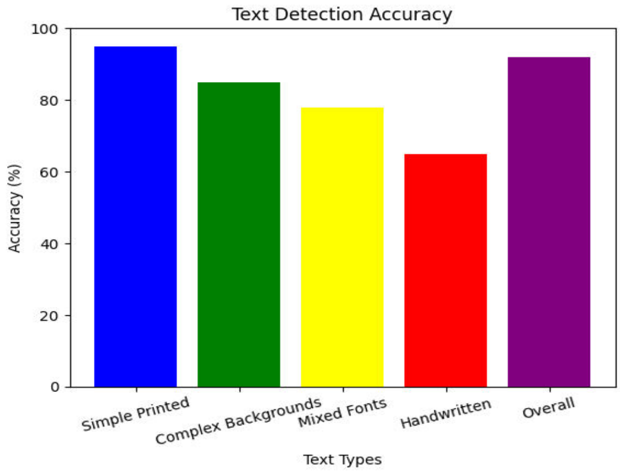

Fig 2: Text Detection Accuracy

The system's Text Detection Accuracy shows a clear dependency on the text type. The highest performance is achieved with Simple Printed text, boasting an accuracy of approximately 95%. Conversely, Handwritten text presents the greatest challenge to the system, resulting in the lowest detection rate at only about 65%. The accuracy for text presented on Complex Backgrounds and the Overall system accuracy are comparable, both falling within the 85-90% range. Meanwhile, text composed of Mixed Fonts is detected with an accuracy slightly lower than the average, nearing 80%. In summary, while the system delivers good Overall Performance (around 90%), its reliability significantly drops when processing less standardized formats like handwriting.

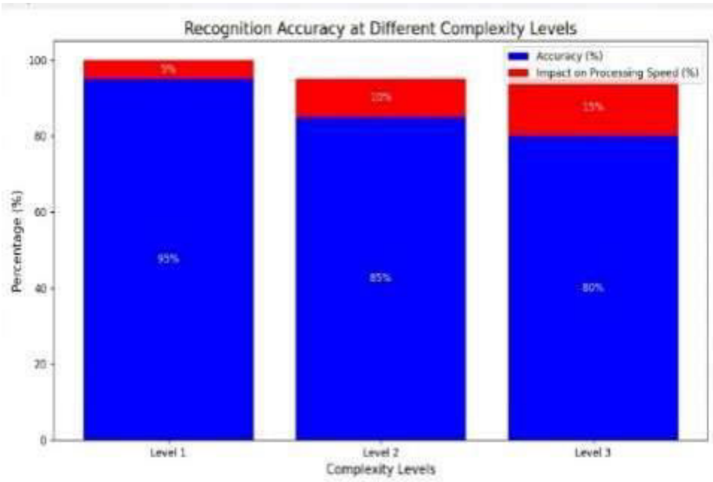

Fig 3: Text Recognition accuracy

The system shows a high proficiency in detecting Simple Printed text (around 95% accuracy), demonstrating its capability with clean, standardized inputs. However, its accuracy significantly degrades when faced with less predictable formats, especially Handwritten text (only 65%), due to the variability in human script. Detection accuracy for text on Complex Backgrounds and Mixed Fonts falls between these extremes (around 80-90%), indicating moderate difficulty in separating text from noise or handling diverse typefaces.

Conclusion:

Code Genie presents an innovative solution to coding education by combining physical engagement with digital learning, addressing the issue of excessive screen time in traditional coding platforms. Its use of tangible blocks for programming encourages hands-on interaction, allowing children to learn key concepts such as sequencing, conditionals, and debugging in a more kinesthetic and engaging way. As a result, the platform promotes logical thinking and problem-solving without the reliance on extended screen exposure. Moving forward, Code Genie could benefit from further integration of advanced technologies, such as machine learning for personalized learning paths and enhanced feedback mechanisms. Additionally, expanding its features to include more complex coding concepts and integrating other game mechanics could provide an even richer learning experience.

References:

1. Angeli, C., & Giannakos, M. (2015). Exploring the effectiveness of physical computing in the classroom: A review of educational studies. *Computers & Education*, 86, 158-173.
2. Bers, M. U. (2018). Coding as a literacy for all children: The promise and challenges of teaching coding in elementary school. *Computers in the Schools*, 35(3), 153-174.
3. Foehr, U. G. (2006). Children and media in the digital age. *Pew Research Center*.
4. García-Peñalvo, F. J., et al. (2019). Trends in educational technology: A systematic review. *Educational Technology & Society*, 22(1), 52-65.
5. Gee, J. P. (2003). What video games have to teach us about learning and literacy. *Computers in Entertainment*, 1(1), 20-20.
6. Gee, J. P., & Hayes, E. R. (2012). Language and learning in the digital age. *Routledge*.
7. Heffernan, N., et al. (2015). The effectiveness of kinesthetic learning in teaching computer science to students with learning disabilities. *Journal of Educational Computing Research*, 53(4), 539-556.
8. Igoe, T. (2016). Making things talk: Using sensors, networks, and Arduino to see, hear, and feel your world. *O'Reilly Media, Inc*.
9. Lye, S. L., & Koh, J. H. (2014). Computational thinking and coding for every student. *Computers in the Schools*, 31(3), 170-181.
10. Maloney, J., et al. (2010). Scratch: A sneak preview. *Proceedings of the International Workshop on Computing Education Research*, 1-6.
11. Papavlasopoulou, S., et al. (2020). Designing coding education tools: A systematic review. *Computers & Education*, 145, 103734.
12. Resnick, M., et al. (2009). Scratch: Programming for all. *Communications of the ACM*, 52(11), 60-67.
13. Waller, J., et al. (2015). The impact of screen time on children's health and development. *Pediatrics*, 135(2), 330-339.