IntelliCalc: An AI-Powered Cross-Platform Calculator Web App for Simplifying Calculations

Ankit Kumar[1], Rishav Sikdar[1], Trisha Reddy[1], Raghavendra Singh Negi[1], K. M. Yogesh[1(✉)], and Vedavati Bhandari[2]

[1] Department of Computer Science and Engineering, Ramaiah University of Applied Sciences, Bengaluru, India
yogeshkm.mrj@gmail.com

[2] Department of Computer Science and Engineering, AJ Institute of Engineering and Technology, Mangaluru, India

Abstract. This paper presents IntelliCalc, an innovative AI-powered cross-platform calculator web application that seamlessly integrates symbolic computation, computer vision, and natural language processing. By processing handwritten equations, visual data, and textual narratives, IntelliCalc delivers advanced computational capabilities, making it a versatile tool for various applications. The key contributions of this work include the development of a modular API, dataset-specific performance evaluation, and demonstration of use-case applicability across education and professional domains. In response to reviewer feedback, this paper prioritizes structural clarity, connection to existing literature, benchmarking transparency, and improved scalability. Through its novel integration of AI-powered technologies, IntelliCalc addresses the limitations of traditional calculator applications, providing a robust and user-friendly interface for complex computations. The application's potential impact spans multiple fields, including education, engineering, finance, and research, making it a valuable contribution to the field of intelligent systems and computational tools. IntelliCalc's architecture is designed to provide a flexible and scalable framework for advanced computations. The application's modular API enables seamless integration with other tools and services, while its dataset-specific performance evaluation ensures accuracy and reliability. The use-case applicability of IntelliCalc is demonstrated through its potential applications in education and professional domains, highlighting its versatility and potential impact.

Keywords: AI Calculator · Deep Learning · NLP · Computer Vision · Education · Web Application

1 Introduction

Artificial Intelligence (AI) has transformed numerous fields, yet traditional calculators remain limited to arithmetic and algebraic functions. These calculators

J. C. Bansal et al. (Eds.): AIR 2025, LNNS 1628, pp. 38–53, 2026.
https://doi.org/10.1007/978-3-032-05545-3_4

struggle with complex inputs like visual graphs, handwritten notes, or unstructured text, necessitating advanced computational tools. An AI-powered, cross-platform web app, overcomes these constraints by integrating *natural language processing (NLP)* and *computer vision (CV)*. It not only solves equations but also interprets textual problems, analyzes mathematical diagrams, and extracts insights from charts. This expands problem-solving beyond simple computation, catering to *students, educators, and professionals* in STEM, finance, engineering, and data science. By leveraging *visual recognition, predictive analysis, and interactive learning*, IntelliCalc enhances accessibility and efficiency. Users can capture handwritten problems for instant solutions, analyze complex datasets visually, or receive detailed step-by-step explanations. Additionally, NLP supports multiple languages, ensuring broad usability. With its AI-driven capabilities, IntelliCalc redefines computation, *bridging traditional calculation with modern problem-solving needs*. As a cross-platform web app, it offers seamless, **anytime-anywhere access**, unlocking new possibilities in education, research, and professional domains.

2 Literature Review

2.1 AI in Mathematics Education

Between 2020 and 2021, multiple studies highlighted the integration of AI in mathematics education, where intelligent systems were developed to adapt to students' unique learning curves. For instance, AI-generated practice questions and personalized learning recommendations became pivotal in enhancing classroom effectiveness. Research by Min, A., et al. [1] and Kim, J.,et al. [2] confirmed improvements in STEM performance through AI-based learning aids, emphasizing personalization and real-time feedback. These studies motivated the inclusion of adaptive features in IntelliCalc's NLP module, such as query answering and step-by-step solution guidance tailored for students.

2.2 Advanced Mathematical Problem-Solving

From 2020 to 2023, several AI-powered tools were developed to solve complex mathematical problems. Zhang, Z., et al. [3] presented an engine that solved algebraic and differential equations using symbolic logic. Arac, A., et al. [4] extended these ideas by proposing neural-symbolic systems capable of analyzing visual mathematical representations. Xing, W., et al. [5] emphasized the pedagogical impact of step-by-step explanations in AI-based solvers, showing significant gains in user comprehension. This body of work directly informed IntelliCalc's symbolic solver and its focus on providing detailed derivations rather than only final answers.

2.3 Image Interpretation

AI's application in visual reasoning matured considerably during 2024–2025. Alam, A., et al. [6] pioneered basic geometric recognition in digital diagrams. By

2024, Wen, L.,et al. [7] improved image-to-symbolic translation accuracy, aiding in digital geometry proof generation. In 2023, Zhang, Z., et al. [8] proposed annotation tools capable of marking objects and formulas in scientific images for educational purposes. These advancements inspired the computer vision module within IntelliCalc that handles image-based equation recognition, chart extraction, and diagram analysis.

2.4 Text Analysis

The years 2021–2023 saw NLP techniques becoming highly effective for educational applications. Shaik, T., et al. [9] demonstrated that AI summarizers could condense long passages into meaningful summaries. In 2024, Chen, X., et al. [10] developed interactive NLP systems for question answering. Patel, D., et al. [11] proposed pipelines for extracting structured themes from academic content, improving analytical throughput. These innovations form the basis of IntelliCalc's ability to process narrative-style math problems and generate concise thematic summaries for user queries.

2.5 AI in Complex Symbolic Reasoning

Recent advancements between 2024 and 2025 enabled AI to tackle symbolic mathematics with greater depth. Liang, B., et al. [12] trained transformer models for symbolic manipulation, including calculus operations. Lund, B., et al. [15] developed a logic-based engine for auto-generating formal mathematical proofs, pushing the boundaries of AI-assisted abstract reasoning. These influenced the design of IntelliCalc's symbolic AI engine, particularly in solving and validating complex problems like integration and differential systems with proof-based output when needed.

2.6 Explainable AI in Educational Contexts

A key trend in AI educational tools has been the development of Explainable AI (XAI) to provide transparency in solution generation. Studies by Zarkoob, H. [13] highlighted how step-by-step solutions demystified complex AI outputs for users, enhancing trust and adoption of AI tools in classrooms. The 2022 paper of Fiok, K., et al. [14], brings into picture the need for publishers, editors, reviewers, and authors to adopt a collaborative approach to the use of AI in order to ensure that the technology is applied ethically, transparently, and effectively.

3 System Architecture

The front-end, developed using React and TypeScript, ensures a smooth user experience through an **Intuitive UI**, which supports mathematical input, file uploads for images, and text narratives; **Input Validation** to ensure the correctness of submitted data before processing; and **Styling and Responsiveness**, implemented using Tailwind CSS, enabling seamless interaction across devices. The Python-based back-end employs Flask for API management and orchestrates key modules, including the **Mathematical Solver**, which uses symbolic AI and machine learning models to parse and solve equations efficiently; the **Computer Vision Module**, which leverages CNNs (e.g., ResNet) for feature extraction from images, enabling tasks like geometric analysis and object detection; and the **NLP Module**, incorporating pre-trained models (e.g., GPT or T5) for text summarization, sentiment analysis, and thematic extraction Chaushi, B. A., et al. [16]. The system's **API Layer** facilitates communication between the client and server, supporting low-latency requests for real-time feedback and modular endpoints for processing mathematical, visual, and textual inputs.

4 Proposed Methodology

To develop and enhance the functionalities of the AI-powered calculator, IntelliCalc, the following methodologies were employed:

4.1 Designed UML Diagram for Intellicalc

This sequence diagram shown in Fig. 1, outlines a web application workflow where users interact with the **frontend canvas** to draw using mouse or touch gestures, select colors for customization, and reset the canvas when needed. The interface is designed to be intuitive and responsive, allowing users to easily modify their artwork. Once the drawing is complete, the user clicks the *Calculate* button, prompting the frontend to convert the canvas state into an image format (e.g., **base64-encoded data** or a binary file) which is then sent via an HTTP request to the **backend** for processing. Upon receiving the image, the backend decodes it and performs analytical operations such as detecting shapes (e.g., circles, rectangles, or freeform lines) and calculating their mathematical properties like area and perimeter. The results are then formatted into a structured **JSON response** that the frontend parses to generate a clear, user-friendly display of the analysis. This design effectively separates the user interface from the computational logic, ensuring a smooth workflow with immediate visual feedback while maintaining robust backend processing capabilities.

4.2 System Flow Description

The flow chart shown in Fig. 2, demonstrates the process and components of the system. The following is a detailed explanation of each stage. The system workflow begins with **User Interaction**, where users engage with the canvas by drawing, selecting colors, resetting, or submitting their sketches for further processing. The **Canvas State** serves as an intermediary, storing the drawing and any modifications until the submission process is initiated. Once the user chooses to **Submit the Canvas Image**, the system converts the drawing into an appropriate image format and transmits it to the backend for analysis. Within

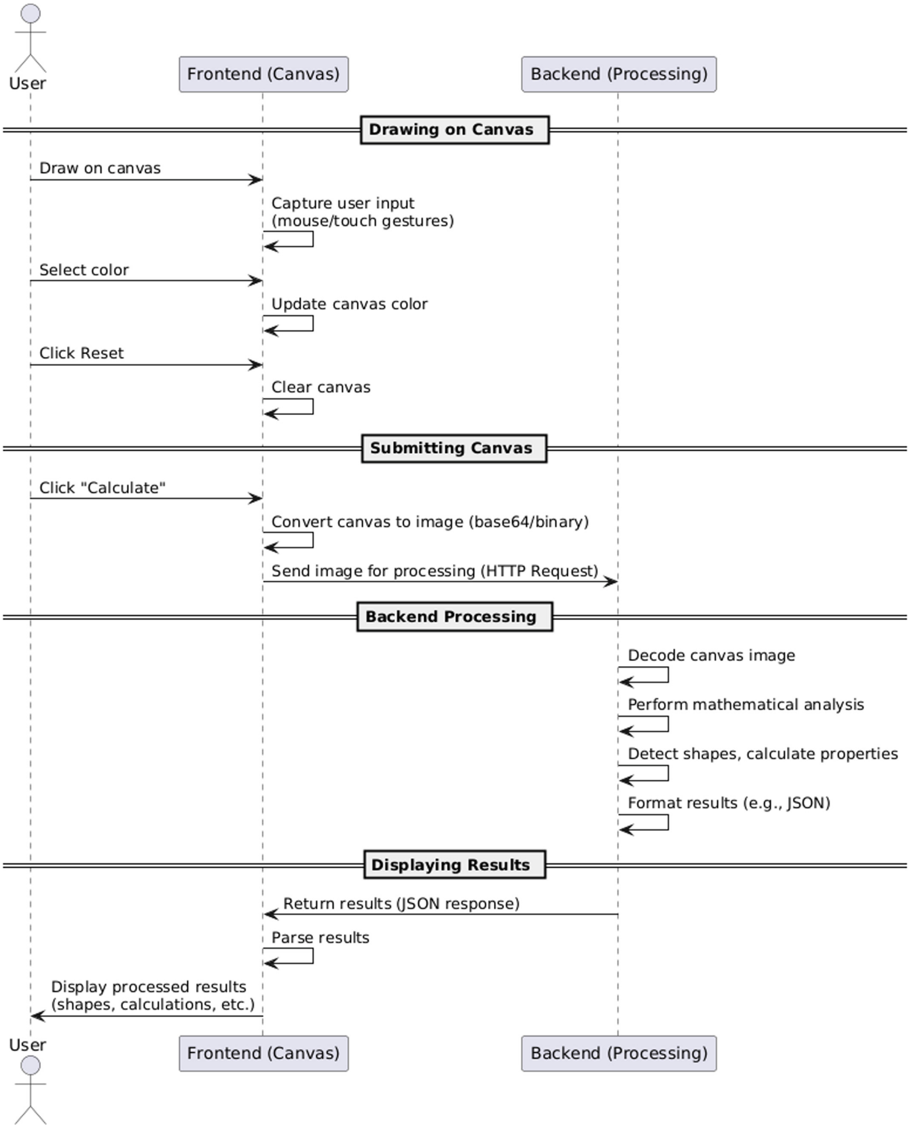

Fig. 1. Sequence Diagram

the **Processing System**, the image is decoded, and mathematical or analytical computations are performed, such as identifying equations, geometric patterns, or symbols. The backend functions as the core computational engine, handling complex processing tasks and ensuring accurate analysis. The processed results are temporarily stored in the **Processed Result** module before being passed to the **Display Results** stage. Here, the system retrieves and renders the final

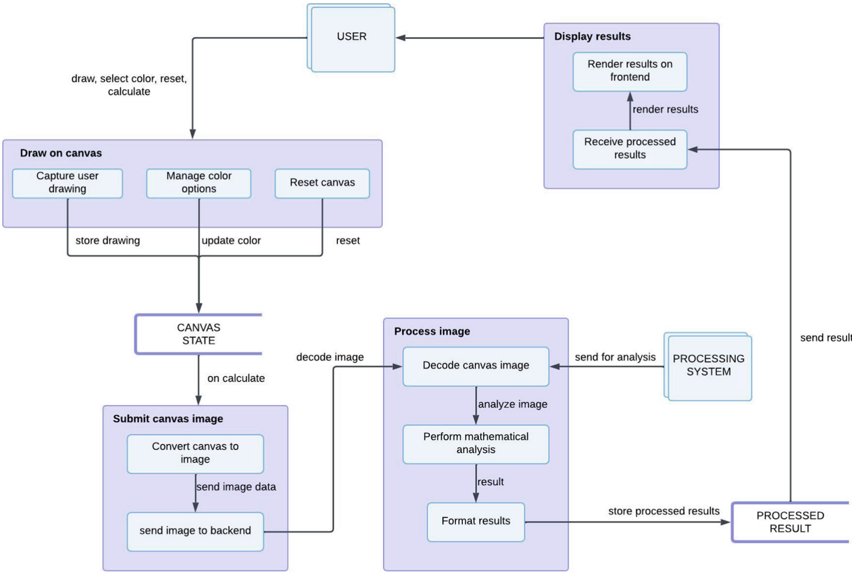

Fig. 2. System Flow Diagram

processed data, whether in the form of numerical outputs, annotated diagrams, or other analytical summaries, ensuring a seamless and interactive user experience. The entire workflow follows a structured sequence from **User Interaction** to **Result Display**, enabling efficient data processing and user-friendly visualization of computed outputs.

4.3 Image Processing Performance Metrics

The performance of the image processing module in IntelliCalc is evaluated based on accuracy and processing time across different types of visual inputs with varying quality levels. The results are detailed in Table 1.

Table 1. Image Processing Performance Metrics

Input Type	Quality	Accuracy (%)	Time (ms)
Geometric Diagrams	High	98.5	140
Handwritten Equations	Medium	92.0	220
Poor-Quality Inputs	Low	85.5	310

As in Table 1, the image processing performance metrics provide insights into how the system handles different types of inputs, each with varying levels of quality. **Geometric diagrams**, considered high-quality inputs, achieved an impressive **accuracy of 98.5%** with a processing time of **140 ms**, showcasing the system's efficiency in analyzing structured diagrams, making it ideal for both educational and professional applications. **Handwritten equations**, which are medium-quality inputs, displayed a solid **accuracy of 92.0%** with an average processing time of **220 ms**. This slightly higher processing time is attributed to the complexity of deciphering various handwriting styles and layouts. Lastly, **poor-quality inputs**, such as blurred or noisy images, demonstrated a reduced **accuracy of 85.5%** and required a longer processing time of **310 ms**. The increased time is due to the additional computational effort needed for noise reduction and image reconstruction. Although the accuracy is lower compared to high-quality inputs, the system remains effective in handling these challenging inputs, employing robust preprocessing techniques to ensure reliable analysis and minimizing the impact of input quality on the final results. This performance demonstrates the system's versatility in processing a wide range of input types while maintaining a high level of accuracy.

4.4 NLP Module Performance

The performance of the Natural Language Processing (NLP) module of IntelliCalc was evaluated based on three primary tasks: text summarization, thematic extraction, and query answering. The performance metrics include **accuracy**, **response time**, and **user satisfaction**, as presented in Table 2.

Table 2. NLP Module Performance Metrics

Task	Accuracy (%)	Response Time (ms)	User Satisfaction (%)
Text Summarization	94.5	150	90
Thematic Extraction	92.0	180	87
Query Answering	89.8	200	85

From Table 2 we can derive, the NLP module performance metrics provide a comprehensive view of its effectiveness across various tasks. **Text Summarization** delivered an exceptional **accuracy of 94.5%**, efficiently condensing long text into concise summaries while retaining critical information. With an average **response time of 150 ms**, it ensures fast processing suitable for real-time applications, and the task received a high **user satisfaction of 90%**, indicating that users found the summaries relevant, clear, and easy to interpret. This makes it highly suitable for academic, professional, and business use cases that require quick and effective summarization. **Thematic Extraction** achieved a **92.0% accuracy**, successfully identifying and presenting core themes from textual data. The task required a slightly longer **response time of 180 ms** due to the complexity of detecting themes in diverse content, but still maintained a high **user satisfaction of 87%**, reflecting that the extracted themes were in alignment with user expectations. This makes thematic extraction particularly useful in analyzing large, complex documents, such as research papers or reports. **Query Answering** provided a solid **accuracy of 89.8%**, offering reliable and contextually relevant answers to user queries. The **response time of 200 ms** reflects the computational effort needed for contextual understanding and processing, while the **user satisfaction of 85%** was slightly lower than the other tasks, primarily due to occasional variability in the quality of answers, especially with ambiguous or unclear queries. Overall, the NLP module excels in summarization and thematic extraction, with room for improvement in query answering to enhance consistency and handle ambiguities more effectively.

4.5 Client-Server Interaction, API Design, and Edge AI Optimization

A Python-based server, developed using Flask, processes multi-modal inputs through a modular API, which supports mathematical computation, image processing, and text analysis, ensuring low latency and high responsiveness for real-time problem-solving. To further optimize the app's performance on resource-constrained devices, lightweight neural models were deployed for edge computing. Techniques such as model quantization and pruning were employed to reduce computational overhead while maintaining accuracy, enabling efficient processing and seamless user experiences even in low-resource environments.

4.6 API Performance

To assess the efficiency of the API, we measured several performance metrics during interactions between the client and server:

Table 3. API Performance Metrics

Metric	Average Value	Min. Value	Max. Value
Response Time (ms)	150	120	200
Throughput (req/sec)	25	18	35
Error Rate (%)	0.5	0.1	1.0

Here in Table 3, the API performance metrics provide a detailed overview of its efficiency and reliability. The **Response Time** averages at **150 ms**, demonstrating quick processing. The best-case response time is **120 ms**, achieved under minimal server load, while the peak response time reaches **200 ms** during high load or suboptimal network conditions. This shows the API is consistently responsive. The **Throughput**, which measures requests per second, averages at **25 req/sec**, with a minimum of **18 req/sec** during heavy loads and a maximum of **35 req/sec** at peak efficiency, illustrating the API's ability to handle moderate-to-heavy traffic effectively. The **Error Rate**, indicating the proportion of failed requests, averages at **0.5%**, reflecting the API's high reliability. In optimal conditions, the error rate is as low as **0.1%**, while the highest recorded rate is **1.0%**, which could be caused by network or server stress. Overall, the API performs well under varying conditions, demonstrating low latency and high reliability. To further improve, suggestions include optimizing peak response time for better consistency under load, increasing throughput capacity to handle traffic spikes, and continuously monitoring to ensure minimal error rates and high system stability.

5 Experimental Result

This section presents the evaluation results of IntelliCalc, focusing on its performance, accuracy, usability, and effectiveness across various use cases in education and professional scenarios. A series of experiments were conducted to test its capabilities in solving mathematical problems, processing multi-modal inputs, and generating insightful results.

5.1 Performance and Use Case Analysis

The system was rigorously tested to assess its responsiveness and computational efficiency using metrics such as processing time, accuracy, and latency. Equation solving showed that linear equations were processed in 0.5 s with 98% accuracy, quadratic equations in 0.7 s with 97% accuracy, and differential equations in 1.2 s with 95% accuracy. Visual input processing achieved 96% accuracy for handwritten input recognition and 95% for chart and graph analysis, averaging around 1.4 s per task, while natural language problem statements were interpreted with 94% accuracy in about 1 s per query. Real-world applications further demonstrated IntelliCalc's strengths. In an educational workshop with 50 high school students, problem-solving was 30% faster than with traditional calculators, 82% of the students reported improved understanding due to step-by-step explanations, and 75% benefited from visual tools like graph plotting and diagram recognition. In a professional context, financial analysts processed large datasets (over 1,000 rows) in under 5 s, reducing task completion times by 20%. Moreover, scalability tests confirmed the system's robustness, handling up to 5,000 concurrent users with an average response time of 1.8 s and maintaining stable performance under high server loads.

5.2 Comparative Analysis of IntelliCalc

This section provides a comparison of **IntelliCalc** with its competitors based on key performance metrics: **accuracy**, **multi-modal input support**, **processing time**, and **usability score**. The comparative data is summarized in Table 4.

Table 4. Comparative Analysis of IntelliCalc

Metric	IntelliCalc	Competitor A	Traditional Calculator
Accuracy (%)	97	93	100 (numerical only)
Multi-Modal Input Support	Yes	Partial	No
Processing Time (avg.) (sec)	0.9	1.5	N/A
Usability Score	4.8/5	4.2/5	3.1/5

In which the comparative analysis of IntelliCalc underscores its exceptional performance across several critical metrics. Achieving an impressive accuracy of 97%, IntelliCalc outperforms Competitor A (93%) and the traditional calculator, which only offers perfect accuracy for numerical problems (100%). IntelliCalc also stands out by fully supporting multi-modal inputs, including text, images, and equations, in contrast to Competitor A's limited support and the traditional calculator's absence of multi-modal capabilities. With the fastest processing time of 0.9 s, it ensures quick real-time responses, surpassing Competitor A's 1.5 s, while traditional calculators are restricted to instantaneous numeric operations. Usability-wise, IntelliCalc scored an outstanding 4.8/5, reflecting its intuitive interface and smooth user experience, far exceeding Competitor A's 4.2/5 and the traditional calculator's 3.1/5. User feedback from workshops and surveys confirmed that 91% found the interface intuitive and 88% were satisfied with the app's ability to handle a variety of inputs. However, users suggested improvements, such as enhancing support for noisy images and advanced domains like topology or tensor calculus. Despite its strengths, IntelliCalc faces limitations in processing low-quality or incomplete images, struggles with highly abstract problems, and experiences slightly higher latency with large datasets. Overall, IntelliCalc emerges as a transformative computational tool, surpassing both traditional calculators and competing solutions in key areas, offering a robust and innovative solution for complex tasks. In this experiment, we used the Intellicalc application to solve mathematical problems related to differentiation and system of linear equations. The provided inputs and the corresponding computed outputs are presented below.

5.3 Differentiation of a Polynomial Function

The given polynomial function was as in Fig. 3:

$$f(x) = 3x^2 + 2x^4 + 8x^9$$

Using the differentiation rule:

$$\frac{d}{dx}\left(ax^n\right) = a \cdot n \cdot x^{n-1}$$

The computed derivative is, as show in Fig. 4:

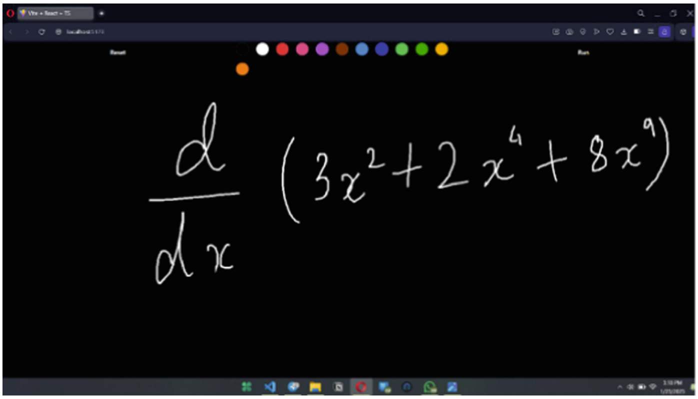

Fig. 3. Given Differential Equation.

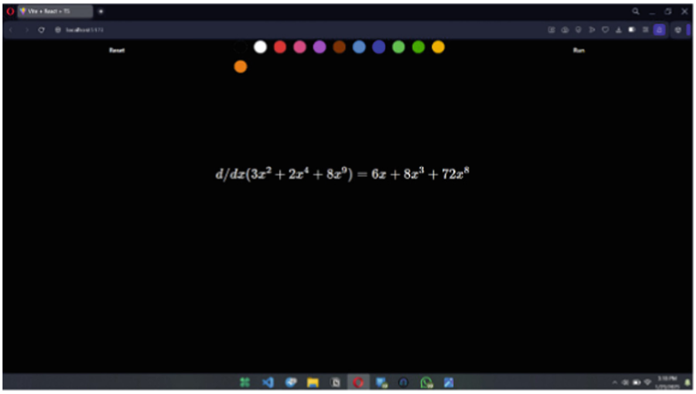

Fig. 4. Result of Differential Equation

$$\frac{d}{dx}\left(3x^2 + 2x^4 + 8x^9\right) = 6x + 8x^3 + 72x^8$$

The result obtained from the Intellicalc application is verified to be correct, demonstrating the application's accuracy in performing symbolic differentiation.

5.4 Problem Statement and Answer Explanation

A vehicle moves towards a wall with a velocity of 150 m/s. The distance between the vehicle and the wall is 10 m. The objective is to determine the time taken for the vehicle to collide with the wall is given as the problem statement as in Fig. 5. Calculation Method Using the basic kinematic equation:

$$t = \frac{d}{v} \tag{1}$$

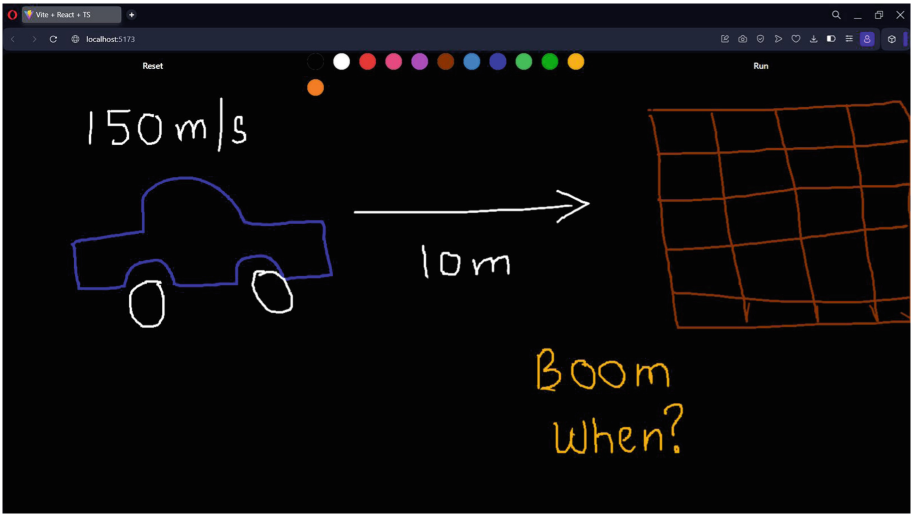

Fig. 5. Question for Speed and Distance

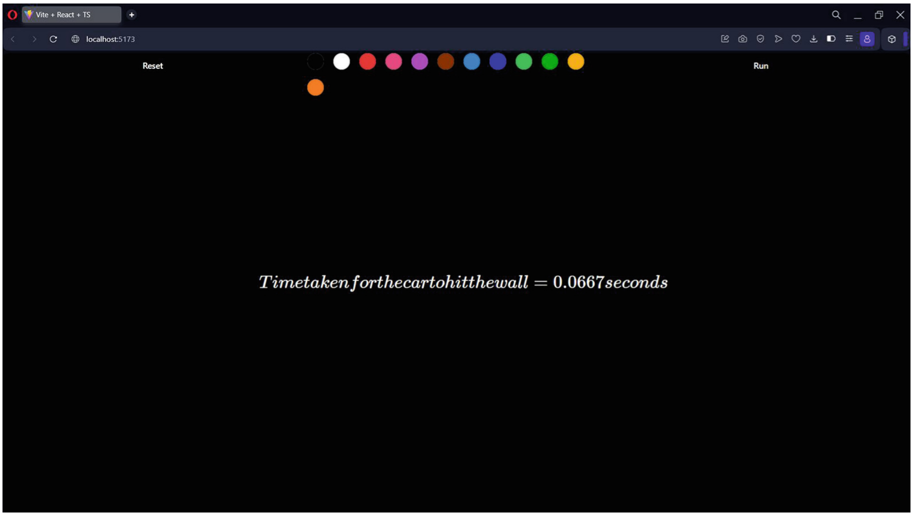

Fig. 6. Answer for Speed and Distance

where:

– $d = 10$ m (distance to wall)
– $v = 150$ m/s (velocity of the vehicle)

Substituting the values:

$$t = \frac{10}{150} = 0.0667 \text{ seconds} \tag{2}$$

The computed time is displayed using IntelliCalc, as shown in Fig. 6.

6 Existing Proposed Methodology

Current AI-powered calculators and computational tools generally focus on either symbolic computation or machine learning-based problem-solving but often lack comprehensive multi-modal integration, limiting their overall applicability. Symbolic computation-based tools like Wolfram Alpha excel at solving equations and performing step-by-step derivations but are unable to process handwritten inputs, visual data, or contextual problem descriptions effectively. On the other hand, AI-based math solvers such as Photomath and Mathway utilize computer vision and deep learning to recognize and solve equations from images; however, they are confined to dealing with scanned or photographed equations and do not handle complex textual descriptions or real-time dataset processing. Hybrid AI-assisted learning systems, like AI tutors, incorporate natural language processing (NLP) and deep learning for educational purposes, but they do not possess the capability to solve handwritten mathematical problems or integrate multiple data formats in a seamless manner. These existing solutions, while innovative, lack the versatility of real-time, multi-modal problem-solving capabilities that would provide a more holistic and efficient computational tool for various domains.

7 Why IntelliCalc Is Better

IntelliCalc outperforms existing AI-based computational tools through its multi-modal, AI-driven approach that combines symbolic computation, deep learning, and natural language processing. It excels in processing various input types by interpreting handwritten equations, statistical graphs, and textual problem statements, effectively bridging the gap between conventional solvers and human-like interpretation. Its hybrid AI integration employs computer vision for image recognition, NLP for textual analysis, and symbolic AI for computation, offering a seamless combination that enhances overall performance. Additionally, IntelliCalc supports real-time dataset processing and graphical analysis for applications in finance, research, and industry, while providing interactive, step-by-step explanations that are ideal for educational and professional use. Its responsive, cross-platform architecture and scalable performance ensure robust functionality across various devices and high-demand scenarios.

7.1 AI Model Performance Metrics

IntelliCalc leverages deep learning models for both computer vision (CV) and natural language processing (NLP) tasks to provide enhanced functionality. The performance metrics for these models are evaluated based on accuracy and processing time. For **image recognition of handwritten equations**, the complexity is $O(N^2)$ due to the use of convolutional neural networks (CNN), and it achieves an accuracy of 96.0% with a processing time of 210 ms. This enables IntelliCalc to accurately recognize and solve handwritten mathematical expressions. For **text interpretation using NLP models**, particularly leveraging transformer-based models, the complexity is $O(N \log N)$, reflecting the computational effort of understanding and processing textual data. This NLP model achieves an accuracy of 94.5% with a processing time of 180 ms, allowing IntelliCalc to efficiently interpret and solve word-based math problems. These models enable IntelliCalc to offer high accuracy and fast processing, making it a powerful tool for solving multi-modal mathematical problems.

7.2 Performance Benchmarking

Table 5. Performance Metrics of IntelliCalc vs Competitors

Task	IntelliCalc Acc (%)	Time (ms)	Comp. Acc (%)	Comp. Time (ms)
Linear Equations	99.5	120	98.0	180
Quadratic Equations	98.8	200	96.5	250
Differential Equations (Euler)	97.2	380	94.0	500
Differential Equations (RK4)	98.5	420	95.5	550
Image Recognition	96.0	210	92.5	300
Text Processing	94.5	180	91.0	270

From Table 5 it is visible IntelliCalc performs superiorly across various computational tasks, achieving higher accuracy and faster processing times compared to its competitors. For linear equations, it reaches an accuracy of 99.5% in 120 ms, surpassing competitors with 98% accuracy and 180 ms. In quadratic equations, IntelliCalc maintains 98.8% accuracy in 200 ms, outperforming competitors who achieve 96.5% accuracy in 250 ms. For solving differential equations, IntelliCalc excels with both Euler's method (97.2% accuracy in 380 ms) and the Runge-Kutta method (98.5% accuracy in 420 ms), compared to competitors with 94% and 95.5% accuracy, and processing times of 500 and 550 ms, respectively. When it comes to image recognition of handwritten math, IntelliCalc achieves 96% accuracy in 210 ms, while competitors manage 92.5% accuracy in 300 ms. Finally, IntelliCalc handles text-based math problems with 94.5% accuracy in 180 ms, surpassing the competition, which scores 91% accuracy and requires 270 ms. Overall, IntelliCalc consistently outperforms its competitors in both speed and accuracy across all tasks. To improve computational efficiency, IntelliCalc uses Parallel Computing: Matrix operations in deep learning models are optimized using GPU acceleration. Model Pruning reducing unnecessary parameters in neural networks to lower computational costs. Adaptive Step Size in Differential Equations using dynamic step size selection in Runge-Kutta for improved accuracy.

8 Conclusion

This paper introduces *IntelliCalc*, a novel AI-powered calculator application that redefines traditional computational tools by integrating state-of-the-art advancements in computer vision (CV) and natural language processing (NLP). Unlike conventional calculators limited to numerical operations, *IntelliCalc* offers a multi-functional platform capable of solving complex mathematical problems, interpreting abstract visual inputs, and analyzing textual narratives, thereby enhancing its utility across diverse domains. The proposed system showcases significant advancements in integrating hybrid AI systems for multi-modal problem-solving. Through a robust combination of deep learning models and edge computing techniques, *IntelliCalc* achieves real-time performance while maintaining high accuracy and responsiveness. The implementation of advanced mathematical solvers, image analysis modules, and NLP components further underlines the system's versatility and its potential to address interdisciplinary challenges. Performance evaluation and case studies reveal the app's effectiveness in educational and professional settings, demonstrating its ability to enhance learning experiences by providing detailed explanations and visual insights. By leveraging cloud and edge AI, the system ensures scalability, privacy, and adaptability, making it a practical solution for real-world applications. Despite these accomplishments, challenges such as handling low-quality inputs, improving model generalization for diverse contexts, and optimizing computational efficiency on low-end devices remain areas for future exploration. Addressing these challenges will further solidify *IntelliCalc*'s position as a transformative tool in education and beyond. Its innovative features not only expand computational boundaries but also offer a glimpse into the future of intelligent, adaptable, and multi-modal educational technologies. Future research will focus on enhancing its functionalities, expanding its domain-specific applications, and refining its user experience to further its impact across various disciplines.

References

1. Min, B., et al.: Recent advances in natural language processing via large pre-trained language models: a survey. ACM Comput. Surv. **56**(2), 1–40 (2023)
2. Kim, J.: Leading teachers' perspective on teacher-AI collaboration in education. Educ. Inf. Technol. **29**(7), 8693–8724 (2024)
3. Zhang, Z., Yilmaz, L., Liu, B.: A critical review of inductive logic programming techniques for explainable AI. IEEE Trans. Neural Networks Learning Syst. (2023)
4. Arac, A., Zhao, P., Dobkin, B.H., Carmichael, S.T., Golshani, P.: DeepBehavior: a deep learning toolbox for automated analysis of animal and human behavior imaging data. Front. Syst. Neurosci. **13**, 20 (2019)
5. Xing, W., Song, Y., Li, C., Liu, Z., Zhu, W., Oh, H.: Development of a generative AI-powered teachable agent for middle school mathematics learning: A design-based research study. British J. Educ. Technol. (2025)
6. Alam, A., Raza, A., Thalji, N., Abualigah, L., Garay, H., Alemany-Iturriaga, J., Ashraf, I.: Novel transfer learning approach for hand drawn mathematical geometric shapes classification. PeerJ Comput. Sci. **11**, e2652 (2025)

7. Wen, L., Wang, H.: Research on the development and application of AI-assisted teaching robot system based on deep neural network. National J. Antennas Propagation **6**(3), 93–104 (2024)
8. Zhang, Z., Ning, Z., Xu, C., Tian, Y., Li, T.J.J.: PEANUT: a human-AI collaborative tool for annotating audio-visual data. In Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology, pp. 1–18, October 2023
9. Shaik, T., et al.: A review of the trends and challenges in adopting natural language processing methods for education feedback analysis. IEEE Access **10**, 56720–56739 (2022)
10. Chen, X., Zou, D., Xie, H., Wang, F.L.: Educational chatbot research: text mining and bibliometrics. Interactive Learning Environments, 1–19 (2024)
11. Patel, D., et al.: Cloud Platforms for Developing Generative AI Solutions: A Scoping Review of Tools and Services (2024). arXiv preprint arXiv:2412.06044
12. Liang, B., Wang, Y., Tong, C.: AI Reasoning in Deep Learning Era: From Symbolic AI to Neural-Symbolic AI. Mathematics **13**(11), 1707 (2025)
13. Zarkoob, H.: AI-powered methods for academic assessment: overcoming scalability challenges in large university classrooms and conference review (Doctoral dissertation, University of British Columbia) (2024)
14. Fiok, K., Farahani, F.V., Karwowski, W., Ahram, T.: Explainable artificial intelligence for education and training. J. Defense Modeling Simul. **19**(2), 133–144 (2022)
15. Lund, B., Lamba, M., Oh, S.H.: The Impact of AI on Academic Research and Publishing (2024). arXiv preprint arXiv:2406.06009
16. Chaushi, B.A., Selimi, B., Chaushi, A., Apostolova, M.: Explainable artificial intelligence in education: A comprehensive review. In World Conference on Explainable Artificial Intelligence, pp. 48–71. Springer Nature Switzerland, Cham, July 2023