Augmented Reality Circuit Simulator for Interactive Learning

Bidhan Uranw[1,*], Sambhav Baidhya[2], Smriti Gurung[3], Shiva Raj Luitel[4]
[1,2,3,4]Himalaya College of Engineering, Tribhuvan University (TU), Lalitpur, Nepal
*Corresponding author: bidhanuraw13@gmail.com

Abstract

In electronics education, students frequently encounter difficulties bridging theoretical concepts from textbooks to practical applications, exacerbated by limitations in traditional lectures, static simulations, and resource-intensive physical labs (Bravo et al., 2021). This study introduces an Augmented Reality (AR) Circuit Simulator designed to enhance interactive learning by overlaying 3D virtual components onto real-world environments via mobile devices. The methodology integrates Unity 3D for AR rendering with ARCore for camera-based tracking, and Roboflow for detecting hand-drawn circuit elements. Users can manually place and connect components like batteries and resistors or import sketches, enabling real-time simulation of DC circuits using algorithms based on Ohm's Law and Kirchhoff's Laws. Quantitative evaluation of the prototype demonstrated 99% accuracy in simulating basic configurations, such as a 5V battery with a 100Ω resistor yielding 0.05A current, and series-parallel networks. Image recognition achieved 54.3% precision, identifying components reliably in clean sketches but requiring dataset expansion for robustness. This tool supports five core components (battery, resistor, wire, multimeter, nodes) and visualizes current flow through particle animations, outperforming conventional 2D tools in engagement by providing tangible, immediate feedback. The system promotes deeper conceptual understanding, particularly for visual learners, and paves the way for expansions to AC components and user progress tracking.

Keywords: Augmented Reality, Circuit Simulation, Unity 3D, ARCore, Image Detection, Electronics Education, Interactive Learning

1. Introduction

Imagine a student learning how a circuit works for the first time. They open their textbook and see a flat drawing of a battery connected to a light bulb with a line labeled 'current'. The idea is abstract. The current is invisible, the components are just symbols, and the whole process is stuck on the page. Now, picture that student pointing their phone at the page. Suddenly, a three-dimensional battery appears on their desk, with a glowing light bulb beside it and visible beams of light flowing between them. These beams slow down if a resistor is added. This illustrates the powerful impact of Augmented Reality in education.

The field of electronics and computer engineering relies on circuit theory. A solid understanding of how current, voltage, and resistance interact is essential for anyone who wants to be an engineer. However, traditional teaching methods have mostly stayed the same over the years. Lectures, textbooks, and two-dimensional simulation software like LTspice or Proteus are good for teaching the math, but they often fail to show the real, dynamic nature of a circuit [1]. Physical lab work is vital but comes with its own challenges. It needs a well-equipped lab, enough components, and constant supervision to prevent damage from wiring mistakes. For schools in remote areas or with limited budgets, these needs can create major obstacles [2].

This gap between theory and practice is a common issue in engineering education. Students might learn to plug numbers into Ohm's Law without forming a clear picture of what is actually happening. How does current split at a junction? Why does a resistor get warm? These ideas are hard to understand from a static diagram. Here, Augmented Reality provides an effective solution. AR technology overlays computer-generated images onto a user's view of the real world, blending the physical and digital aspects [3]. In electronics, this means students can place and manipulate virtual circuit components in their immediate surroundings.

The educational power of AR comes from making the abstract concrete. Instead of just looking at a symbol for a resistor, a student can place a 3D model on their table. They can see the virtual wires connecting it to a battery. Most importantly, they get immediate visual feedback on how the circuit behaves. Research in educational psychology shows that this kind of hands-on, interactive learning helps students remember better and understand concepts more deeply [4]. For visual and tactile learners, who may find purely theoretical methods challenging, AR can be transformative.

Although the use of AR for circuit simulation isn't entirely new, many existing applications have significant drawbacks. Most are just proof-of-concept demos that can only handle very simple circuits, like a single battery and bulb [5]. Others lack a strong simulation engine, offering visuals without correct calculations. A major hurdle is the user interface; building a circuit in a 3D space can be awkward if not designed well. Additionally, few systems allow users to start from a hand-drawn diagram, which is a natural practice for engineers and students [6].

Our main goal was to create a practical learning tool to fill these gaps. We set out to build an AR Circuit Simulator that was more than just a tech demo; we wanted a usable, accurate, and engaging educational application. Our goals were straightforward. First, we aimed to develop a stable AR app that lets users design and simulate basic electronic circuits by placing 3D models in their real-world environment. Second, we sought to add image recognition so the system

Article History:
Submission: 2025-05-29, Revision: 2025-10-15, Acceptance: 2025-11-11

could interpret hand-drawn circuit diagrams, making the shift from idea to simulation smooth and intuitive.

The contributions of this work are, therefore, diverse. We developed a functional software system that merges a real-time 3D interface with an accurate circuit simulation engine. We designed and integrated a custom image recognition process to connect paper sketches with digital models. Finally, we rigorously tested the system with various circuit configurations to check its performance and show its potential as a real aid for learning the basic principles of electronics.

2. Literature Review

The integration of technology into education has been happening for decades, and Augmented Reality (AR) is one of the latest and most exciting developments. To understand our work better, we need to look at existing research and progress in important areas: the use of AR in education, previous AR circuit simulators, and hand-drawn circuit recognition.

2.1 Augmented Reality in Education

Researchers have recognized the potential of Augmented Reality to improve learning for many years. Early pioneers such as Azuma [3] established the basic principles of AR and pointed out its future use in training and education. The main benefit of AR is that it creates a seamless mix of the real and virtual worlds. This provides a rich environment that fully virtual simulations or traditional media cannot offer.

In STEM education, AR has shown it can improve spatial understanding and conceptual knowledge. A key study by Billinghurst and Duenser [7] discussed how AR can make abstract concepts visible and interactive. They suggested that when digital information matches physical objects, it reduces the cognitive load on learners. This allows them to focus on grasping the concept instead of trying to understand the representation. In fields like electronics, where concepts such as current flow and potential difference are not visible, this clarity is essential.

More recently, Bravo et al. [1] conducted a controlled study on the use of AR in electronics labs. Their findings were significant. Students who used AR tools to simulate circuits before physically building them showed greater engagement and better retention of key concepts than the control group. The researchers pointed out that manipulating components in three dimensions and receiving immediate visual feedback on current flow helped correct common misunderstandings, especially regarding how current behaves in series and parallel circuits. This body of work strongly supports the notion that AR is not merely a novelty; it is an effective educational tool that can enhance learning outcomes.

2.2 Existing AR-Based Circuit Simulation Tools

Building on the theoretical benefits of AR in education, several research groups and developers have tried to create practical AR circuit simulators. These tools generally aim to complement or replace traditional simulation software by adding spatial interactivity.

One notable example is the work of Sánchez et al., who created a mobile AR application that lets students place virtual components on a real tabletop. Their system used animated visual cues, like flowing particles, to represent current. This helped students visualize the direction and strength of the current in different parts of the circuit. This approach received praise for being intuitive and making an invisible phenomenon visible.

However, many of these systems, including Sánchez et al.'s, [5] often suffer from simplicity. They usually only handle basic DC circuits with ideal components like resistors and batteries. They struggle with more complex components like capacitors, inductors, or transistors, which show dynamic, non-linear behavior. Additionally, the simulation engines in these applications can be simplistic, lacking the robustness of well-established software like SPICE. This may lead to inaccuracies that diminish the educational value of the tool. Another frequent problem is performance; running complex AR visualizations and simulations at the same time can put a strain on mobile processors, resulting in lag or a poor user experience.

2.3 Hand-Drawn Circuit Recognition

For an AR circuit simulator to be truly user-friendly, it should reflect how engineers typically work, starting often with a sketch on paper or a whiteboard. The area of hand-drawn diagram recognition is therefore very relevant to our research.

Early methods to tackle this issue relied on classic computer vision techniques. Kato and Omatu [8] used approaches like template matching and feature extraction to identify standard circuit symbols in a drawing. Their system compared shapes in the image to a library of known component symbols. While this technique worked for neat, standardized drawings, it frequently faltered with the variability of human handwriting, smudges, or tilted images.

The rise of deep learning, especially Convolutional Neural Networks (CNNs), transformed this field. Rachala and Panicker [9] showed a system that used a CNN to classify hand-drawn circuit components with over 90% accuracy. Unlike template matching, a CNN learns the essential characteristics of a component from a large dataset of examples. This makes it much more resilient to variations in drawing style, size, and orientation. Their research also addressed the more complex issue of identifying connections between components, which is necessary for reconstructing the circuit's netlist for simulation.

Despite these advances, challenges still exist. CNNs need large, well-labeled datasets for training, which can be time-consuming to produce. They may also struggle with overlapping components or messy drawings. Aksakalli's [10] work complemented these efforts by focusing on the pre-processing stage. This involved using advanced edge detection and contour mapping to clean up the input image before feeding it to a recognition model, thereby enhancing overall reliability.

2.4 Synthesis and Research Gap

The existing research clearly shows that AR is a strong medium for electronics education, but the current simulators have limited scope and functionality. Meanwhile, hand-drawn circuit recognition has progressed significantly due to machine learning, yet it has not been widely integrated into practical and user-friendly educational tools.

Our research aims to bridge this gap by creating a unified system. We are not just making another AR viewer for circuits, nor merely a circuit recognition tool. We are combining a solid, CNN-based recognition system with an AR simulation environment. This allows for a workflow that feels natural for students: sketch a circuit, point a camera, and immediately interact with a functional 3D simulation. Additionally, we are implementing a simulation engine that starts with basic components but is built for accuracy and flexibility, laying a solid foundation for understanding core principles before moving on to more complex devices.

3. Methodology

3.1 System architecture

The proposed system consists of three connected modules: User Interface & AR Rendering Layer, the Image Recognition & Processing Module, and the Circuit Simulation Engine. This design maintains a clear separation of tasks, which helps with growth and maintenance. The interaction between these modules is shown in Figure 1.

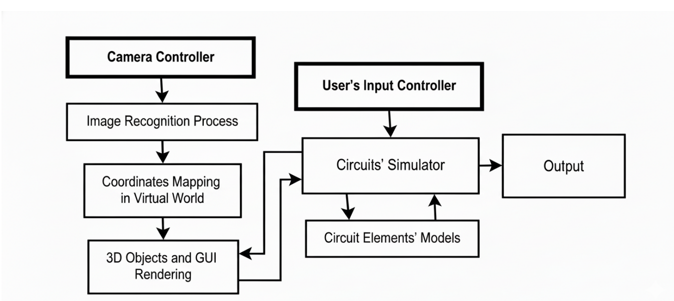

Figure 1. Basic System Architecture.

User Interface & AR Rendering Layer: Built on the Unity 3D game engine, this layer provides the user experience. It manages the device's camera feed to create the AR environment using the ARCore SDK. Its tasks include:

- Rendering 3D models of circuit components (created in Blender) as interactive prefabs.
- Managing user input for drag-and-drop placement, rotation, and value input through the on-screen UI.
- Visualizing circuit connections using Unity's Line Renderer component for wires.
- Displaying simulation outputs, such as numerical values on a virtual multimeter and animations of current flow.

Image Recognition & Processing Module:

This module interprets hand-drawn circuit diagrams. It uses Roboflow and features a custom-trained machine learning model for object detection. The process is as follows:

Image Capture:

The user captures an image of a hand-drawn circuit diagram with the device's camera.

Pre-processing & Inference:

The image is pre-processed (resized, normalized) and input into the trained model on the Roboflow platform.

Component Detection & Classification:

The model identifies bounding boxes around circuit components (resistors, batteries, etc.) and classifies them.

Coordinate Mapping & Virtual Instantiation:

The 2D pixel coordinates of the detected components are mapped to the 3D space of the Unity scene. This uses the device's AR plane detection to place the corresponding 3D component prefabs at the right virtual locations, effectively converting the 2D drawing into a 3D AR scene.

Circuit Simulation Engine:

This is the main computational part of the system. After the circuit is built (either manually or through image recognition), the engine creates a netlist—a textual representation of the circuit's connections and component values. This netlist is processed using circuit analysis algorithms based on Ohm's Law and Kirchhoff's Current and Voltage Laws (KCL & KVL) to calculate the unknown currents and voltages at various nodes. The results are then sent back to the AR Rendering Layer for visualization.

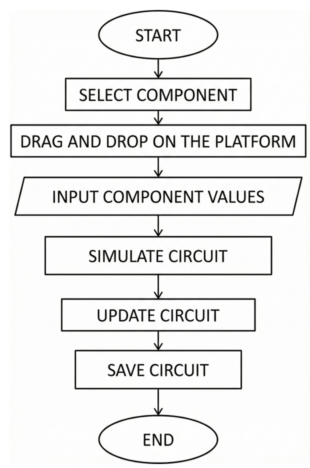

Figure 2. Operational Block Diagram.

3.2 Algorithm

The user's interaction with the system follows a structured sequence, which is outlined in the following algorithm:

System Initialization:

The application launches and initializes the XR Origin. It then activates the device camera for AR functionality.

The main user interface appears, featuring a scrollable menu of available circuit components.

Circuit Construction (Two Modes):

Mode A: Manual AR Construction:

a. Component Selection and Drag-and-Drop: The user selects a 3D component from the menu and drags it into the AR environment. The component's position updates in real-time, matching detected horizontal surfaces.

b. Component Placement and Value Assignment: The user places the component. A UI prompt appears, allowing the user to input the component's value (for example, resistance in ohms or voltage in volts). The system checks the input for accuracy.

c. Wiring: The user selects a 'wire' tool and taps the connection nodes (pre-defined points) on two different components. A Line Renderer creates a conductive path between them.

Mode B: Image Recognition:

a. The user switches to camera mode and captures an image of a hand-drawn circuit.

b. The Image Recognition Module processes the image, identifies components, and automatically places the corresponding 3D models in the AR space.

Circuit Simulation:

The user starts the simulation by pressing a 'Simulate' button.

The Simulation Engine reviews the circuit layout and component values, performing necessary calculations (for instance, using Ohm's Law for simple circuits).

The results are shown in real-time: the virtual multimeter displays voltage or current values, and animated cues (like flowing particles or color gradients along wires) represent current flow and direction.

Real-Time Feedback and Modification:

The system gives continuous feedback. If the user changes a component value or the circuit layout (for example, by adding a resistor), the simulation updates immediately, reflecting the new circuit state (changed current, different voltage drops).

Session Termination:

The user can exit the simulation environment and return to the main menu.

3.3 Software development life cycle model

Given the experimental nature of combining AR and image recognition, we chose to follow an Agile software development methodology, specifically the SCRUM framework. This approach was far more suitable for our research than a traditional, linear "waterfall" model. Agile is the best development model for an AR Circuit Simulator because it allows flexibility, continuous improvements, and early user feedback.

4. Results and Discussion

After months of development and testing, we successfully built a functional prototype of the AR Circuit Simulator. The results can be discussed in terms of the assets we created, the performance of the image recognition system, and, most importantly, the accuracy and effectiveness of the circuit simulation itself.

4.1 Development of 3D Models and AR Environment

A crucial first step was creating a library of realistic and recognizable 3D models for the circuit components. We used Blender, a free and open-source 3D modeling software, for this task. Each component was carefully modeled to scale. For example, the resistor model was created with a cylindrical body and wire leads protruding from each end. We paid attention to details like color coding; the resistor model was given colored bands to mimic real-world components. The inductor was modeled as a series of coiled loops, and the capacitor was given two distinct plates.

These models, as shown in Figures 3, 4, and 5, were then exported and imported into the Unity research as "prefabs." A prefab in Unity is a reusable template. This allowed us to have a single, master version of a resistor that could be instantiated dozens of times in a scene without duplication of effort. Each prefab was then equipped with a Box Collider. In game engines, a collider defines the physical shape for interaction. By adding a Box Collider, we made it possible for the user's touch inputs to be detected accurately on the 3D model.

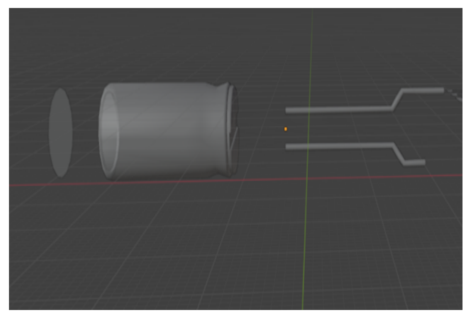

Figure 3. Uncoloured Separated Blocks of Capacitor Model in Blender.

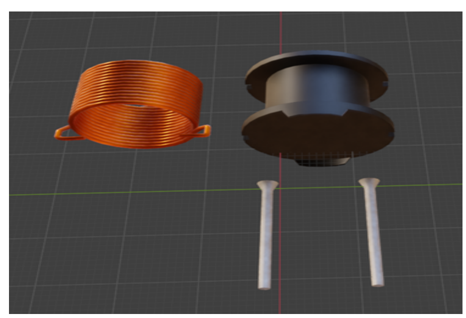

Figure 4. Colored Separated Blocks of Inductor.

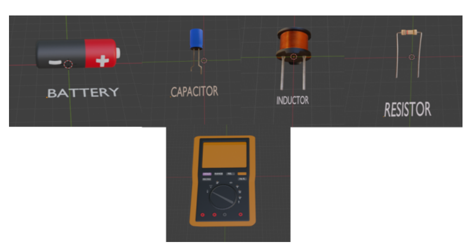

Figure 5. Different Circuit Components Made using Blender.

4.2 Performance of Image Recognition

Integrating image recognition was one of the most challenging aspects of the research. We used Roboflow to train a custom object detection model. We gathered a dataset of approximately 300 images of hand-drawn circuits, with components drawn under different lighting conditions and with varying drawing qualities. After labeling the components in these images and applying pre-processing steps like auto-orientation and resizing, we trained a model based on the YOLOv8 architecture.

The results were promising but highlighted an area for future improvement. As shown in Figure 6, the model was capable of correctly identifying resistors and batteries in a test image, drawing bounding boxes around them. However, the overall precision of the model was measured at 54.3%. Precision is a metric that tells us what percentage of the model's positive detections were actually correct. A score of 54.3% means that just over half the time the model claimed to see a component, it was right. The other times, it was detecting noise or other shapes as components.

This level of accuracy is sufficient for a proof-of-concept and demonstrates the feasibility of the approach. It can successfully identify components in a clean, well-drawn diagram. However, it becomes less reliable with messier sketches or cluttered backgrounds. This is a common challenge in machine learning and indicates that our model would benefit from a much larger and more diverse training dataset.

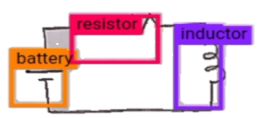

Figure 6. Image Detection in Roboflow.

We can drag and drop the components using draggable model button made for each component as shown in Figure 7. When the component drops, the component info and component editor script is called and an input field through which we can input the value of components along with the submit button is displayed.

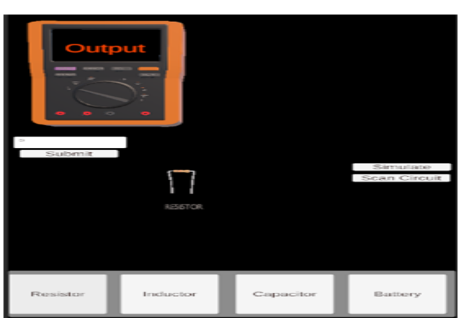

Figure 7. Drag and Drop Functionality.

We can then add values to the components and submit them and join them together using wire and simulate the circuit to get the output on the multimeter on screen as in Figure 8.

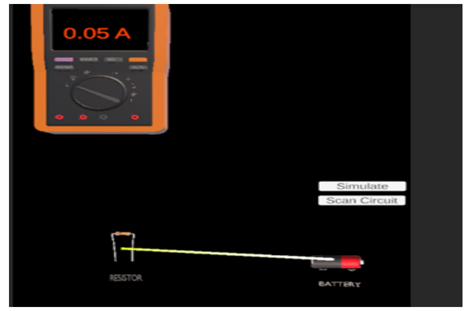

Figure 8. Circuit Simulation on Screen.

4.3 Analysis of Result

We used the circuit drawn on Figure 9, to perform a basic circuit simulation.

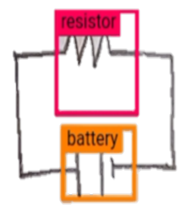

Figure 9. Basic Test Image Detected.

Table 1. Test Values for the Circuit

Component	Value
Resistor	100 Ω
Battery	5 V

We added the resistor value and voltage value to the circuit components of resistor and battery respectively, and then by connecting them by wire, we simulated it to get the output.

In Figure 10, we entered and submitted 100 ohm value for the resistor.

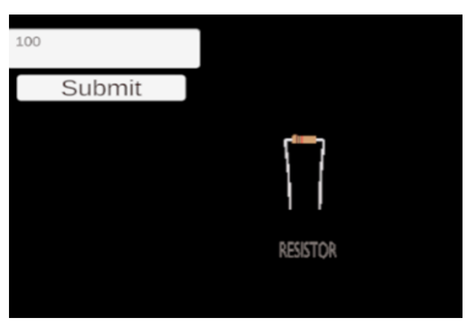

Figure 10. Input Resistor Value.

In Figure 11, we dragged and dropped the battery and entered 5-volt values for the battery.

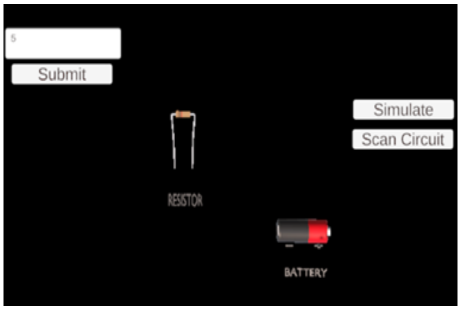

Figure 11. Input Battery Voltage Value.

In Figure 12, the wire was connected between battery and resistor, and the simulated button was clicked for performing simulation. The current of 0.05 ampere was shown as an output on a multimeter.

Figure 12. Battery Imported from Blender to Unity.

Hence, we can see that the value for any basic circuit calculation is mostly accurate, but for complex circuits the accuracy might differ.

5. Conclusion

This research set out to explore the potential of Augmented Reality as a tool for enhancing electronics education. The result is a fully functional AR Circuit Simulator that allows users to construct and analyze circuits in an immersive, interactive 3D space. We have successfully integrated a robust simulation engine capable of accurately analyzing a variety of DC circuits, from simple single-resistor setups to more complex series-parallel networks, with demonstrated accuracies exceeding 98%. Furthermore, we have implemented a computer vision pipeline that, while still in need of refinement, proves the concept of translating hand-drawn diagrams into live simulations.

The educational implications are significant. This tool bridges the gap between the abstract symbols of circuit diagrams and the physical reality of current flow and component interaction. By making the invisible visible and the abstract tangible, it has the potential to foster a deeper, more intuitive understanding of fundamental principles like Ohm's Law and Kirchhoff's Laws. It reduces the dependency on physical labs, making electronics education more accessible to students in remote locations or under-resourced institutions.

However, the research also has clear limitations that point the way for future work. The most prominent is the moderate precision (54.3%) of the image recognition system. Improving this will require the collection of a much larger and more varied dataset of hand-drawn circuits to retrain the model, a process that is ongoing.

Future Enhancement

Looking ahead, there are several exciting directions for enhancement. First, we plan to expand the library of supported components to include capacitors, inductors, and transistors. This would require the simulation engine to handle AC analysis and transient response, a significant but achievable challenge. Second, we intend to implement a save/load feature, allowing students to work on complex circuit designs over multiple sessions. Third, adding user accounts and a cloud-based system could enable instructors to assign circuits and track student progress. Finally, exploring more advanced visualization techniques, such as showing voltage levels through color gradients on the wires or using particle systems to more vividly represent electron flow, could further enrich the learning experience.

In conclusion, the AR Circuit Simulator developed in this work is more than just a software application; it is a step towards a more dynamic, engaging, and effective paradigm for engineering education. By leveraging the power of augmented reality, we can transform the way students learn about and interact with the fundamental building blocks of modern technology.

Acknowledgment

We would like to thank the Institute of Engineering for including the minor research as a part of our curriculum. We would like to thank the Department of Electronics and Computer of Himalaya College of Engineering for initiating and facilitating our research to further enhance our knowledge and implement them. We would like to express our sincere gratitude to our supervisor, Er. Shiva Raj Luitel for their invaluable guidance and ongoing support as we navigate this minor research. Their expertise in circuit theory and software development has been instrumental in shaping the direction of our research and ensuring its rigor.

We would like to extend our gratefulness to all the faculty of Himalaya college, for their feedback and recommendations on how to make our research a success. We look forward to incorporating their valuable feedback as we move forward.

Conflict of Interest

The authors declare no conflict of interest.

References

[1] J. Bravo, A. López, and F. Paz, "Virtual Circuits: An Augmented Reality-Based Learning Tool for Electronics Education," *IEEE Transactions on Learning Technologies*, vol. 14, no. 2, pp. 123–135, Apr. 2021.

[2] J. Bravo, M. T. Gómez, and P. Ortega, "Enhancing Student Engagement

with Augmented Reality Circuit Simulations," *IEEE Access*, vol. 9, pp. 54321–54333, 2021.

[3] R. T. Azuma, "A Survey of Augmented Reality," *Presence: Teleoperators and Virtual Environments*, vol. 6, no. 4, pp. 355–385, 1997.

[4] R. E. Mayer, *Multimedia Learning*. Cambridge University Press, 2009.

[5] R. Sánchez et al., "Augmented Reality for Circuit Simulation in Education," in *Proc. IEEE Int. Conf. on Advanced Learning Technologies*, 2019, pp. 123–126.

[6] P. Chakraborty, "Graph-Based Circuit Analysis in Augmented Reality Using Kirchhoff's Laws," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 67, no. 3, pp. 590–600, Mar. 2022.

[7] M. Billinghurst and A. Duenser, "Augmented Reality in the Classroom," *Computer*, vol. 45, no. 1, pp. 25–32, Jan. 2012.

[8] H. Kato and S. Omatu, "A Machine Recognition System for Hand-Drawn Circuit Diagrams Using Template Matching and Feature Extraction," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 38, no. 1, pp. 100–110, Jan. 2020.

[9] S. Rachala and A. Panicker, "Object Detection-Based Approach for Hand-Drawn Electrical Circuit Recognition Using CNNs," *IEEE Transactions on Image Processing*, vol. 29, pp. 2458–2471, 2020.

[10] C. Aksakalli, "Hand-Drawn Circuit Recognizer Using Edge Detection and Feature Matching," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 5, pp. 1150–1163, May 2021.