Interactive Multimodal Learning: Towards Using Pedagogical Agents for Inclusive Education

Muhammad H. Al Omoush
Faculty of Engineering and Computing
Dublin City University
Dublin, Ireland
muhammad.menazelalomoush2@mail.dcu.ie

Sumaia E. Salih
Independent researcher
Dublin, Ireland
sumaiasalih@outlook.com

Sameer Kishore
Department of Computer Engineering and Informatics
Middlesex University
Dubai, UAE
s.kishore@mdx.ac.ae

Tracey Mehigan
Faculty of Engineering and Computing
Dublin City University
Dublin, Ireland
tracey.mehigan@dcu.ie

***Abstract*— A pedagogical agent is an animated interface in an interactive online learning environment. Its role involves guiding users through instructions and participating in direct discussions. Numerous research studies underline the advantages of pedagogical agents in delivering instruction, tailoring learning experiences, and promoting inclusivity. The authors utilize machine learning techniques to develop an interactive multimodal application (combining visuals and audio) designed as a pedagogical agent. This application is intended to assist in teaching primary school students (aged 6 to 8) the recognition of colors and letters. Additionally, it incorporates voice interaction to assess their learning progress. The overarching objective of this pedagogical agent is to seamlessly integrate artificial intelligence (AI) into educational settings, thereby amplifying student engagement and motivation during the learning process. This paper will explore the journey of creating a machine learning-based application and its potential utility in supporting educators. Furthermore, the authors will investigate potential augmentations that could render the application more adept at aiding students with dyslexia. The paper will emphasize the significance of human-computer interaction (HCI), as it significantly shapes pedagogical agents' design and operational aspects.**

Keywords*— *Human-Computer Interaction HCI, Inclusive education, machine learning, multimodal pedagogical agent, sensory disability, Graphical User Interface (GUI).

I. INTRODUCTION AND BACKGROUND

Research studies underscore the critical significance of integrating technology in classrooms to enhance student engagement and personalized learning [1, 2, 3, 4]. Educators now have various technological tools aiding students in comprehending subjects. Incorporating diverse learning techniques promotes the potential for all students to grasp the educational content [5, 6, 7].

Challenges faced by students in classrooms, such as learning disabilities (dyslexia) and ADHD [8], along with motivation and engagement issues [9], can impact academic performance. Tailoring educational content to meet individual learner needs fosters motivation, leading to improved learning outcomes. This personalized approach enhances material engagement, maintaining retention and understanding [10]. Addressing these challenges and catering to varying learning preferences remains a substantial 21st century educational challenge [11]. Cultivating an inclusive classroom accommodating all students' needs promotes their success and well-being, ensuring equitable learning opportunities.

STEM education (science, technology, engineering, and mathematics) is pivotal. It cultivates critical thinking and problem-solving, which are imperative for today's society [12, 13]. Educational institutions must prioritize STEM programs and resources to prepare students for future careers, enabling access to high-quality STEM education.

This paper introduces a machine learning application, a pedagogical agent, aiding students in learning colors and letters while acting as a teaching assistant for inclusive instruction. Developed using PyTorch and a feedforward neural network, the application initially classifies letters (A to Z) and colors (blue, green, orange, red, violet, yellow). A voice interaction component facilitates visual and audio communication. The ML-based pedagogical agent offers alternative learning methods and support for all students.

A. The Evolution of Pedagogical Agents

The concept of pedagogical agents originated in the 1970s with the emergence of Intelligent Tutoring Systems (ITS), aimed at emulating human tutors in addressing student queries, identifying misconceptions, and providing feedback [14]. Over the last two decades, the prominence of pedagogical agent research has steadily increased [15], with a notable surge in development between 1998 and 2000 [16]. These agents have demonstrated their capacity to adapt to individual learning paths and accommodate distinct needs [17]. Pedagogical agents manifest in various forms: text-based, voice-based, 2-D character-based, 3-D character-based, and even human-like representations. Progress in human-computer interface technology has created more authentic and human-resembling instructional agents [18].

II. VISION HUMAN-COMPUTER INTERACTION (HCI) AND ITS CRUCIAL ROLE IN SHAPING INCLUSIVE PEDAGOGICAL AGENTS

Human-Computer Interaction is centered around optimizing interactive computer systems' user experience and interface [19]. The goal is to design user systems that are effective, efficient, easy to use, and enjoyable. In the context of pedagogical agents, HCI supports personalizing and adapting learning experiences. This leads to individualized learning paths for learners based on their interests, preferences, and learning styles. Consequently, learners benefit from a customized and practical learning experience. Real-time feedback and assessments from pedagogical agents allow learners to track their progress and receive immediate guidance.

A pedagogical agent's user interface encompasses visual and auditory elements, including the agent's appearance, voice, animations, and interactive components like buttons, menus, and dialogue boxes [21]. Designing the user interface adheres to HCI principles like stability, accessibility, and flexibility. This ensures effective interaction between learners and agents, enabling learners to leverage the agent's instructional support.

Within pedagogical agents' architecture, HCI plays a vital role in interactions between humans and computer systems. It encompasses design, usability, and digital interface effectiveness [19]. Pedagogical agents have emerged as tools for inclusive education, supporting diverse learners, including those with disabilities [20]. The connection between HCI and pedagogical agents lies in designing accessible and effective agents for all learners, regardless of abilities. Designing interfaces for pedagogical agents considers visual and auditory accessibility, navigation, and interaction options [21] to accommodate learners with disabilities [22]. Usability, responsiveness, adaptability, and ease of use are crucial for an inclusive learning experience.

HCI's role in shaping pedagogical agent design is pivotal for inclusive education. It ensures usability and accessibility for diverse learners, including those with sensory or learning disabilities. HCI incorporates emotional affect and cognitive processing in user interactions [23]. Pedagogical agent design aligns with HCI considerations, resulting in more effective and user-friendly systems that cater to diverse cognitive and emotional states. This holistic approach creates pedagogical agents capable of delivering engaging and personalized learning experiences, ultimately enhancing user learning outcomes.

A. *Text-to-speech (TTS) and Pedagogical Agents*

Text-to-speech vocalizers, or speech synthesizers, are computer systems that generate human-like speech by converting text into spoken words [22, 24, 25, 26]. They provide voices and personalities to artificial agents, virtual assistants, and robots. Text-to-speech vocalizers have also brought innovative advancements in education [24]. Text-to-speech technology has been shown to facilitate language learning, particularly in improving speaking and pronunciation skills [22, 27]. It allows the users to customize the speech output's pace, pitch, and volume to align with their preferences and requirements, which helps learners who may be facing difficulties listening to their own words during the writing process. This, in turn, helps them monitor and revise their writing outcomes.

Engaging with a pedagogical agent using text-to-speech technology can benefit students with learning disabilities such as dyslexia, who may struggle with reading and writing [22] and often experience challenges with memory retention [28]. Our machine learning-based pedagogical agent employs text-to-speech vocalizers (synthesizers) to assist students in their educational journey. It integrates multi-sensory instruction (audio and visual), personalized learning (helping students enhance their auditory verbal memory), and supportive feedback. This integration aims to benefit students in overcoming challenges and achieving academic success. This approach makes education more accessible and inclusive.

Technological progress has enhanced the communication abilities of pedagogical agents when interacting with students [29]. These agents, used in educational settings, have become more sophisticated in their communication skills due to technological advancements. However, as pedagogical agents continue to increase in learning environments, it is important to gain a deeper understanding of how students communicate with these agents to evaluate their effectiveness in facilitating learning [30,31].

III. PEDAGOGICAL AGENTS: BENEFITS FOR STUDENTS LEARNING AND MOTIVATION

Pedagogical agents have been found to offer significant benefits to students, including facilitating collaboration and providing feedback as students engage in the learning process. These agents can communicate with students using various methods, such as facial expressions, natural language, and hand gestures, which enhances the interactive and engaging nature of the learning experience [21].

A. *Empowering Learning Through Interactive Pedagogical Agents*

Pedagogical agents have significantly impacted students' motivation to study. Students tend to place a higher level of trust and confidence in the information these agents provide than in information provided by traditional teachers. This can be attributed to the interactive and dynamic nature of the agents, which students find appealing and engaging [21]. Overall, pedagogical agents can transform how students learn and interact with learning material, enhancing motivation and engagement in the learning process.

B. *Exploring Multisensory Approaches for Enhanced Learning and Engagement*

The possibilities of multimodal pedagogical agents in the teaching and learning process have been recently explored, especially for students with learning disabilities [15]. Dyslexia is a learning disability that can significantly impact students' educational experiences. In response to these challenges, researchers have explored multimodal applications, which integrate different modes of information, such as text, audio, and visual features. Incorporating multi-sensory technology in education offers numerous advantages, including increased student engagement and improved learning outcomes [32].

Artificial intelligence has been integrated and brought into action in education, promoting advancements in several areas. Intelligent educational systems offer teachers and students timely, personalized guidance and feedback [33]. Visual feedback has traditionally been the norm for providing information to users in computer interfaces. However, incorporating other sensory modalities, such as audio, can enhance user experience and provide new ways of engagement for different user groups in diverse environments [34].

Having multiple modalities in a user interface has the benefit of spreading the interaction across various senses or control abilities of the user. If one modality is not available or fully utilized, another can be employed to ensure successful interaction, especially for those with sensory or situational impairments [34, 35]. At the same time, a study showed that preschoolers at risk of dyslexia improved various pre-reading skills, such as phonological awareness, after practicing these skills on multimedia intervention software on a computer device [36].

Pedagogical agents with interactive and multimodal features increase students' vocabulary acquisition with various learning needs [37, 38,39]. These multimodal applications incorporate audio and visual features and can be powerful tools for teaching dyslexic students [40]. This approach enables all students to learn through different sensory modalities, providing a more engaging and accessible education experience.

IV. The ML-Based Pedagogical Agent

The authors have developed a cutting-edge educational agent that employs computer vision to enhance interactive learning for young students, specifically focusing on letters and colors. This agent seamlessly incorporates visual and auditory cues, cultivating a seamless communication channel between users and the system. The project was developed in three stages: creating a neural network classifier for letters, implementing color recognition, and adding voice interaction using Python's text-to-speech library. These stages converge to offer an engaging, multisensory learning experience. The agent's merging of technology and pedagogy benefits learners and opens routes for future educational innovations, demonstrating the transformative potential of interactive technology in education.

The implications of this pedagogical agent extend beyond the direct learning process. The agent accommodates diverse learning styles and abilities by combining visual and audio elements, fostering inclusivity in education.

A. Material and Methods

The pedagogical agent was constructed using Machine Learning (ML). The ML application was built using the PyTorch framework and the feedforward neural network. PyTorch is an efficient deep-learning tensor library built on Torch and Python, enabling real-time testing and execution [41].

A dataset is a collection of various kinds of data that have been digitally preserved. Any project using machine learning needs data as its primary input. For addressing various A.I. difficulties, datasets often include photos, texts, audio, videos, and numerical data points [42].

Neural networks are computerized structures with interconnected nodes that mimic the function of brain neurons. They can cluster and categorize raw data using algorithms, identify hidden patterns and correlations, and – with time – continually learn and improve [43]. A feedforward neural network is an artificial neural network in which nodes' connections do not form a loop. Because all information flows forward in feedforward neural networks, they are sometimes referred to as a multi-layered network of neurons [44, 45].

B. Letter Classification

When employing PyTorch for machine learning model training, the subsequent actions were executed: dataset preparation, establishment of data loaders, construction of the machine learning model, specification of the loss function and optimizer, and ultimately, model training.

(a) Preparing the Letters Dataset

The dataset of letters was sourced from Kaggle and consisted of 26 classes, each representing a letter of the English alphabet. There were approximately 52 images available for each letter. However, it was necessary to expand the dataset further to ensure practical model training. Various data augmentation techniques were applied to achieve this, artificially increasing the dataset's size, and improving the model's learning process.

The dataset was then divided into two subsets: a training and validation set. Specifically, 80% of the dataset was used for training, while the remaining 20% was set aside for testing and validation purposes.

(b) Creating the Data Loader

To ensure adequate training of a neural network, it is essential to input data in batches, typically comprising 32 or 64 batches, each consisting of multiple images. This approach of using batches is employed to optimize the training process by reducing the time needed for adjusting parameters. If the neural network were trained on a per-image basis, where parameters are updated for each image individually, it could lead to inconsistent parameter adjustments and sluggish generalization.

However, the training process becomes more reliable and efficient when the neural network is evaluated on a batch of 32 images. Parameters are adjusted based on the collective information from the entire batch. A well-known deep learning framework, PyTorch, includes a built-in feature called "DataLoader" that facilitates the organization of images into batches. The DataLoader automatically shuffles images, ensuring the selection of batches is random. This randomness guarantees a diverse and representative sample for training.

The utilization of batch-based training offers benefits beyond just speeding up the process. It also enhances the neural network's overall performance and ability to generalize. By allowing the network to learn from multiple images simultaneously, it captures underlying patterns and data structures more effectively, thus improving its generalization capability.

(c) Creating the Machine Learning Model

The utilized model in this research was constructed using a feedforward neural network architecture, representing a straightforward yet exceptionally efficient design approach to attain favorable results. The input image, possessing dimensions of 28x28 pixels, undergoes a process of flattening into a column vector comprising 784 individual elements. Within the feedforward structure, each distinct class is associated with a concealed layer with a dimension of 512 and an output layer with a capacity of 26 units.

(d) Training the Model

The model learns by using data from the Data-Loader, checking its output against what is expected, and calculating a loss. Then, we see how accurate the model is using test data. These are the main steps in training the model.

C. Color Classification

The dataset used in this study came from Kaggle and had six color categories: Green, Blue, Orange, Red, Violet, and Yellow. Each category had around 25 images. We divided the dataset into 80% for training and 20% for model testing.

To figure out the colors in the images, we used a mix of two techniques: k-means clustering and measuring color differences. We applied the k-means algorithm to find the primary colors in each image. We then transformed this primary color into its HSV (Hue, Saturation, Value) form.

D. Speech Recognition

A voice interaction functionality was developed to facilitate "communication" with the software, utilizing a combination of a TTS (text-to-speech) library and a speech recognition library referred to as STT (speech-to-text). In Python, a convenient module called pyttsx3 was employed for text-to-speech conversion. This module stands out due to its compatibility with both Python 2 and 3 and its offline capabilities, distinguishing it from alternative libraries. This user-friendly tool enables the transformation of written text into spoken words and offers support for two distinct voices (male and female) made available through the "sapi5" interface on Windows [46].

V. Executing the Machine Learning Model

To enhance functionality and user experience, the machine learning application was divided into two modes: 'Vision Only Mode' and 'Voice Interaction Mode.' Before commencing the recognition process, users must indicate their chosen model preference as shown in Figure 1.

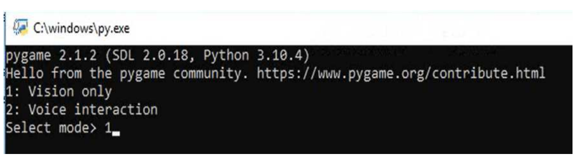

Fig. 1. Within the user interface, individuals have the choice to opt for either visual mode (option 1) or voice interaction mode (option 2).

A. Engaging Voice Interaction Mode

Upon enabling the voice interaction feature (depicted in Figure 3), the user is required to subsequently designate the appropriate microphone (device microphone) from the provided list of available devices.

B. Executing Letter Recognition Process

In the 'vision only' mode, the camera initiates the recognition process, promptly displaying the identified letter in real-time at the upper-left corner of the screen (as depicted in Figure 2). The system audibly announces the recognized letter, offering immediate feedback to the user.

Upon engaging in the 'voice interaction' mode, the user is directed to select the appropriate microphone to optimize the recognition process. Once a letter is presented before the camera, the recognition process begins, operating briefly. Upon completion, the system promptly notifies the user of the end of recognition, facilitating their progression to the next step.

To initiate letter recognition, the user is prompted to articulate a word commencing with the letter they provided earlier. For instance, if the user displays the letter 'H' before the camera, they must utter a word that starts with the letter 'H.' Following this, the system delivers feedback by affirming 'the word is matching' for correct responses or indicating 'the letter is not matching' for incorrect ones.

Fig. 2. Using the vision-only interaction mode for letter recognition.

C. Executing Colour Recognition Process

The application's color recognition and voice interaction procedures are congruent with the steps mentioned above. The sole distinction lies in the 'voice interaction' mode as shown in Figure 3, wherein the system prompts the user to articulate the color they have presented for verbal recognition verbally.

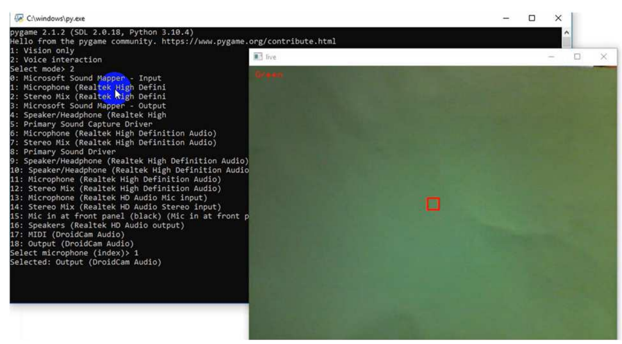

Fig. 3. Using the Voice interaction mode for color recognition (the color's name can be seen in the top right corner).

VI. Evaluating System Functionality: A Comprehensive Examination

During the training phase of the letter classification model, 26 classes representing each letter of the alphabet were utilized. This dataset comprised approximately 52 images per class. The data was partitioned, allocating 80% for training and 20% for testing. A batch size of 32 and 5 epochs were employed for the validation set. However, upon conducting tests, it was observed that the program encountered difficulties recognizing certain letters, specifically B, C, D, E, K, O, R, and S.

This issue may stem from the dataset's size, as insufficient data can lead to overfitting. Overfitting can yield impressive results during training, but the model might struggle to generalize or demonstrate robust recognition when presented with new data. To avoid such problems and ensure dependable model performance, working with a sufficiently extensive dataset is imperative. Notably, the testing and training phases yielded an accuracy of around 98%. Exploring alternative optimization techniques could be considered for future endeavors to enhance accuracy.

Various testing scenarios were executed to evaluate the system's accuracy. The first scenario, described in Table 1, involved assessing the program's vision component using a laptop camera. Testing encompassed letter and color recognition against a white background with standard room lighting. Optimal outcomes were achieved in this scenario. In contrast, the least favorable results were obtained when testing against a vividly colored background in a dimly lit room.

Table 2 presents the outcomes of system testing in voice interaction for letter and color recognition. Notably, higher accuracy was attained when the system was assessed in a quiet environment, and the user maintained a proximity of less than 50 cm from the laptop.

The system displayed its highest recognition accuracy when a white background was employed for color and letter recognition, particularly in a quiet room. However, recognition accuracy notably declined when the software was deployed in noisy surroundings featuring varied colors and ambient noise.

Maintaining an ongoing process of detecting and identifying letters and colors posed a difficulty in system testing. As a solution, it is advisable to enhance the system by capturing a screenshot of the letter/color prior to commencing the recognition process rather than depending solely on continuous camera detection. Upgrading to superior hardware is recommended to improve detection accuracy and voice interactions. Introducing a high-quality microphone for voice interactions can expedite user comprehension, thereby reducing occurrences of misinterpreting spoken words.

TABLE 1: System testing for the letters and colors using the vision-only mode.

Testing type	**Letters**	**Colors**
With room lighting	80 % pass 20 % fail	All pass
Decreasing the room lighting	67 % pass 33 % fail	All pass except blue, green
Using a white background	85 % pass 15 % fail	All pass
Using a noisy - colorful background	57 % pass 43 % fail	All pass, except blue, green

TABLE 2: System testing for voice interaction.

Testing type	**Results**
In a quiet room	The system works fine.
In a noisy room - classroom	The microphone can't hear the voice, and the speaker is low.
Sitting close (less than 50 cm from the laptop) to the microphone	The system works fine.
Sitting far (more than 50 cm from the laptop).	The microphone can't hear well. Misinterprets the words.

VII. ENHANCING INCLUSION: FUTURE DEVELOPMENTS FOR THE PEDAGOGICAL AGENT

The challenges encountered by students with learning disability in acquiring letter and color knowledge can be effectively addressed by integrating sensory components into the existing application. These enhancements include audio-guided instructions and voice-based interactions, facilitating letter recognition and auditory-verbal memory for dyslexic learners. Furthermore, these attributes deliver immediate feedback and positive reinforcement for accurate responses, thus maintaining learning outcomes and motivation for all learners.

Several tools and technologies can be integrated further to enhance the Machine Learning-based Pedagogical Agent for inclusive education. Specifically tailored to the needs of dyslexic students, the application's user interface can be designed to accommodate dyslexia-friendly fonts, suitable color contrasts, and distinct visual cues that facilitate reading and comprehension. Embracing a phonics-centered methodology, emphasizing the correlation between letters and sounds can be particularly advantageous for dyslexic students grappling with phonological processing difficulties. Additionally, integrating letter contour highlighting and color-coded letter systems can contribute to more efficient letter recognition and differentiation among dyslexic students.

The future trajectory of the Machine Learning-based Pedagogical Agent is decidedly promising, marked by ongoing technological and research advancements. Continued progress in hardware and software domains augments the pedagogical agent's precision, speed, and user-friendliness.

VIII. CONCLUSION

The main objective was to develop a pedagogical agent using machine learning that can be seamlessly integrated into the field of education to perform advanced image recognition and effectively distinguish between letters and colors. By leveraging the power of artificial intelligence (AI) in the classroom, this project holds immense significance as it has the potential to revolutionize traditional teaching methods and inspire educators to adopt innovative techniques to improve student learning outcomes. With the developed application, teachers can now utilize a sophisticated tool to evaluate the learning progress of students aged 6 to 8 more precisely and efficiently. However, as with any innovative project, there is always room for improvement and further development. Several potential modifications could be considered to enhance the application's capabilities and expand its potential in the field of education. Firstly, instead of students holding the letter or color in front of the camera for recognition, the system could be modified to prompt students to present a specific color or letter. This would add an interactive element to the process, encouraging students to actively engage with the application and participate in the learning process, thereby fostering a more dynamic and immersive educational experience. Secondly, to optimize the application processing overhead, the continuous recognition approach could be refined by capturing a screenshot and performing recognition only when the letter or color is in front of the camera. This will ensure smooth and efficient operation during classroom activities.

To broaden the application's language recognition capabilities, it could be enhanced to read complete words instead of just individual letters. This would enable teachers to assess students' language skills more comprehensively and provide feedback on their word recognition abilities, thus facilitating language development. By incorporating these potential modifications, the application could be further refined and extended to unlock new possibilities in the field of education. With continuous improvements and advancements, this project has the potential to impact how teachers teach, and students learn, paving the way for a more innovative and effective educational environment.

The simplicity of this machine learning (ML) application sets it apart from other research in the field, as it has been designed explicitly with elementary students in mind. This consideration is crucial, ensuring the software is user-friendly and accessible to young learners. The software has been developed to be easily compatible with Raspberry Pi, making it simple to operate on any humanoid robot with a camera, microphone, speaker, and Raspberry Pi. The ease of use and compatibility of this software make it a viable option for integration into classrooms, where it has the potential to motivate teachers to explore and apply different methods to enhance students' learning experiences. The software's adaptability to different hardware configurations, such as Raspberry Pi, ensures that it can seamlessly integrate into classroom setups without requiring significant modifications or investments in specialized equipment.

As a future improvement, adding a Graphical User Interface (GUI) could further enhance the user experience of the software. Including a GUI would allow users to interact with the software through visual elements on the screen, providing a more intuitive and user-friendly interface for choosing between camera and voice interaction options. For instance, users could click on the screen to select their preferred option rather than using the keyboard to input numerical choices (1 for camera, 2 for voice interaction). Implementing a GUI could make the software even more accessible and user-friendly, especially for young students who may be more familiar with graphical interfaces. By incorporating a GUI, the software could provide users a seamless and interactive experience, enhancing their engagement and making the learning process more enjoyable and effective. The potential addition of a GUI in future iterations could further enhance the software's usability and user-friendliness, motivating teachers to explore innovative methods for enhancing students' learning experiences. With continuous improvements and refinements, this software has the potential to contribute to the advancement of education by incorporating ML technology in a simple and accessible manner.

ACKNOWLEDGMENT

This work was conducted with the financial support of the Science Foundation Ireland Centre for Research Training in Digitally-Enhanced Reality (d-real) under Grant No. 18/CRT/6224.

REFERENCES

[1] N. L. Schroeder, O. O. Adesope, and R. B. Gilbert, “How Effective are Pedagogical Agents for Learning? A Meta-Analytic Review.” Journal of Educational Computing Research, vol. 49, no. 1, pp. 1-39, 2013, doi: 10.2190/ec.49.1.a

[2] T. Daxing, "Research on Cultivating New Teachers’ Literacy Based on Artificial Intelligence," 2021 2nd International Conference on Big Data and Informatization Education (ICBDIE), 2021, pp. 228-231, doi: 10.1109/ICBDIE52740.2021.00058.

[3] S. M. S. Khaksar, B. Slade, J. Wallace, and K. Gurinder, “Critical success factors for application of social robots in special developmental schools: Development, adoption and implementation,” Int. J. Educ. Manag., vol. 34, no. 4, pp. 677–696, 2019.

[4] S. Kadry and B. Ghazal, “Design and Assessment of Using Smartphone Application in the Classroom to Improve Students’ Learning.” International Journal of Engineering Pedagogy (iJEP), vol. 9, no. 2, p. 17, 2019, doi: 10.3991/ijep.v9i2.9764.

[5] R. Kokku, S. Sundararajan, P. Dey, R. Sindhgatta, S. Nitta and B. Sengupta, "Augmenting Classrooms with AI for Personalized Education," 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2018, pp. 6976-6980, doi: 10.1109/ICASSP.2018.8461812.

[6] A. AL-KAİSİ, A. Arkhangelskaya, O. Rudenko-Morgun, and E. Lopanova, “PEDAGOGICAL AGENTS IN TEACHING LANGUAGE: TYPES AND IMPLEMENTATION OPPORTUNITIES.” IJAEDU-International E-Journal of Advances in Education, vol. 5, no. 15, pp. 275-285, 2019, doi: 10.18768/ijaedu.593875.

[7] M. Alaimi, E. Law, K. D. Pantasdo, P.-Y. Oudeyer, and H. Sauzeon, “Pedagogical Agents for Fostering Question-Asking Skills in Children.” Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems, 2020, doi: 10.1145/3313831.3376776.

[8] J. R. Harrison, D. A. Soares, S. Rudzinski, and R. Johnson, “Attention Deficit Hyperactivity Disorders and Classroom-Based Interventions: Evidence-Based Status, Effectiveness, and Moderators of Effects in Single-Case Design Research.” Review of Educational Research, vol. 89, no. 4, pp. 569-611, 2019, doi: 10.3102/0034654319857038.

[9] B. Nadera, “Promoting Student Motivation in EFL Classroom: Through Extended Music Education.” Procedia - Social and Behavioral Sciences, vol. 199, pp. 368-371, 2015, doi: 10.1016/j.sbspro.2015.07.520

[10] Mehigan, T. J. (2013). Automatic detection of learner-style for adaptive eLearning. PhD Thesis, University College Cork.

[11] Concordia University (2021). 3 reasons why teachers need to use technology in the classroom. [online] www.concordia.edu. Available at: https://www.concordia.edu/blog/3-reasons-why-teachers-need-technology-in-the-classroom.html.

[12] Ardianti, S., Sulisworo, D., Pramudya, Y., & Raharjo, W. (2020). The impact of the use of STEM education approach on the blended learning to improve student’s critical thinking skills. Universal Journal of Educational Research, 8(3B), 24-32.

[13] D. M. A. Mallik, “Design Thinking for Educators-Unleashing Imaginnovation Ideas Being Student Centric.” SDMIMD Journal of Management, vol. 10, no. 2, pp. 71-74, 2019, doi: 10.18311/sdmimd/2019/24445.

[14] G. Veletsianos and G. S. Russell, “Pedagogical Agents.” Handbook of Research on Educational Communications and Technology, pp. 759-769, 2014, doi: 10.1007/978-1-4614-3185-5_61.

[15] G. Clarebout, J. Elen, W. L. Johnson, and E. Shaw, 2002. Animated pedagogical agents: An opportunity to be grasped? Journal of Educational Multimedia and Hypermedia, vol. 11, no. 3, pp. 267–286, 2002

[16] R. K. Atkinson, “Optimizing learning from examples using animated pedagogical agents,” Journal of Educational Psychology, vol.94, 2 , 2002.

[17] W. L. Johnson, J. W. Rickel, and J. C. Lester, "Animated pedagogical agents: Face-to-face interaction in interactive learning environments," International Journal of Artificial Intelligence in Education, vol. 11, pp. 47–78, 2000

[18] A. S. D. Martha and H. Santoso, “The Design and Impact of the Pedagogical Agent: A Systematic Literature Review.” The Journal of Educators Online, vol. 16, no. 1, 2019, doi: 10.9743/jeo.2019.16.1.8.

[19] V. Kühne, A. M. Rosenthal-von der Pütten, and N. C. Krämer, “Using Linguistic Alignment to Enhance Learning Experience with Pedagogical Agents: The Special Case of Dialect.” Intelligent Virtual Agents, pp. 149-158, 2013, doi: 10.1007/978-3-642-40415-3_13.

[20] Du, X., & Breazeal, C. (2022). Co-designing with Children: Exploring Changes in Special Education Teachers' Attitudes and Design Belief towards Pedagogical Agents.

[21] Martha, A. S. D., & Santoso, H. B. (2019). The design and impact of the pedagogical agent: A systematic literature review. Journal of Educators Online, 16(1), n1.

[22] Liew, T. W., Tan, S. M., Pang, W. M., Khan, M. T. I., & Kew, S. N. (2023). I am Alexa, your virtual tutor!: The effects of Amazon Alexa's text-to-speech voice enthusiasm in a multimedia learning environment. Education and Information Technologies, 28(2), 1455-1489.

[23] Mehigan, T., and I. Pitt. "Engaging Learners Through Emotion in Artificially Intelligent Environments. "Proceedings of EDULEARN19 Conference 1st-3rd July 2019, Palma, Mallorca, Spain, 5661-5668.

[24] J. Cambre, J. Colnago, J. Maddock, J. Tsai, and J. Kaye, "Choice of Voices: A Large-Scale Evaluation of Text-to-Speech Voice Quality for Long-Form Content." Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems, pp. 1-13, 2020, doi: 10.1145/3313831.3376789.

[25] Alonso Martin, F., Malfaz, M., Castro-González, Á., Castillo, J. C., & Salichs, M. Á. (2020). Four-features evaluation of text to speech systems for three social robots. Electronics, 9(2), 267. https://doi.org/10.3390/electronics9020267

[26] A. Poushneh, "Humanizing voice assistant: The impact of voice assistant personality on consumers' attitudes and behaviors." Journal of Retailing and Consumer Services, vol. 58, pp. 1-10, 2021, doi: 10.1016/j.jretconser.2020.102283.

[27] Liakin, D., Cardoso, W., & Liakina, N. (2017). The pedagogical use of mobile speech synthesis (TTS): Focus on French liaison. Computer Assisted Language Learning, 30(3–4), 325–342.

[28] de Assis Leão, S. E. S., Menezes Lage, G., Pedra de Souza, R., Holanda Marinho Nogueira, N. G. D., & Vieira Pinheiro, Â. M. (2022). Working Memory and Manual Dexterity in Dyslexic Children: A Systematic Review and Meta-Analysis. Developmental Neuropsychology, 1-30.

[29] Sikström, P., Valentini, C., Kärkkäinen, T., & Sivunen, A. (2022). How pedagogical agents communicate with students: A two-phase systematic review. Computers & Education, 104564.

[30] Davis, R. O., Park, T., & Vincent, J. (2021). A systematic narrative review of agent persona on learning outcomes and design variables to enhance personification. Journal of Research on Technology in Education, 53(1), 89-106.

[31] N. Sun and J. Botev, "Intelligent Adaptive Agents and Trust in Virtual and Augmented Reality." 2020 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct), pp. 303-305, 2020, doi: 10.1109/ismar-adjunct51615.2020.00085.

[32] J. Taljaard, "A review of multi-sensory technologies in a Science, Technology, Engineering, Arts and Mathematics (STEAM) classroom." Journal of Learning Design, vol. 9, no. 2, pp. 46-55, 2016, doi: 10.5204/jld.v9i2.274.

[33] L. Chen, P. Chen, and Z. Lin, "Artificial Intelligence in Education: A Review." IEEE Access, vol. 8, pp. 75264-75278, 2020, doi: 10.1109/access.2020.2988510.

[34] E. Freeman, G. Wilson, D.-B. Vo, A. Ng, I. Politis, and S. Brewster, "Multimodal feedback in HCI: haptics, non- speech audio, and their applications," in The Handbook of Multimodal-Multisensor Interfaces: Foundations, User Modeling, and Common Modality Combinations - Volume 1, ACM, 2017, pp. 277–317.

[35] Duffy, S., Price, S., Volpe, G., Marshall, P., Bianchi-Berthouze, N., Cappagli, G., ... & Gori, M. (2017, July). WeDraw: Using multisensory serious games to explore concepts in primary mathematics. In Proceedings of the 13th international conference on technology in mathematics teaching (Vol. 13). 13th International Conference on Technology in Mathematics Teaching (ICTMT 13)

[36] A. Romero, U. Garay, E. Tejada, and A. L. De la Serna, "Efficacy of Berni: A software for preschoolers at risk of dyslexia." International Journal of Child-Computer Interaction, p. 100411, 2021, doi: 10.1016/j.ijcci.2021.100411.

[37] W. Clark,R. Luckin "ipads in the classroom" Available at: https://digitalteachingandlearning.files.wordpress.com/2013/03/ipads-in-the-classroom-report-lkl.pdf (Accessed: April 10, 2023).

[38] S. H. Lee and K. B. Aspiranti, "Using multimodal educational apps to increase the vocabulary of children with and without reading difficulties." International Journal of Child-Computer Interaction, vol. 36, p. 100579, 2023, doi: 10.1016/j.ijcci.2023.100579.

[39] A. Jariwala, D. Marghitu, and R. Chapman, "A Multimodal Platform to Teach Mathematics to Students with Vision-Impairment." Lecture Notes in Computer Science, pp. 109-117, 2021, doi: 10.1007/978-3-030-78095-1_10.

[40] Dyslexia and reading instruction (no date) National Council for Special Education - CPD and In-School Support. Available at: https://www.sess.ie/dyslexia-section/dyslexia-and-reading-instruction (Accessed: April 10, 2023).

[41] "PyTorch Loss Functions: The Ultimate Guide - neptune.ai." neptune.ai. https://neptune.ai/blog/pytorch-loss-functions (accessed: Apr. 06, 2023).

[42] "What is PyTorch, and How Does It Work? | Simplilearn." Simplilearn . https://www.simplilearn.com/what-is-pytorch-article (accessed: Apr. 07, 2023).

[43] Sas.com. (2019). Neural Networks - What are they and why do they matter? [online] Available at:https://www.sas.com/en_us/insights/analytics/neural-networks.html.

[44] P. Sharma. "Feedforward Neural Network: Its Layers, Functions, and Importance." 2022. https://www.analyticsvidhya.com/blog/2022/01/feedforward-neural-network-its-layers-functions-and-importance/ (accessed: Apr. 07, 2023).

[45] "Feed Forward Neural Network Definition | DeepAI." DeepAI. https://deepai.org/machine-learning-glossary-and-terms/feed-forward-neural-network (accessed: Apr. 07, 2023).

[46] "Python | Text to Speech by using pyttsx3 - GeeksforGeeks." geeksforgeeks. 2022. https://www.geeksforgeeks.org/python-text-to-speech-by-using-pyttsx3/ (accessed: Apr. 06, 2023).

[47] M. Pino and L. Mortari, "The Inclusion of Students with Dyslexia in Higher Education: A Systematic Review Using Narrative Synthesis." Dyslexia, vol. 20, no. 4, pp. 346-369, 2014, doi: 10.1002/dys.1484.

[48] A. Baratè, G. Haus, L. A. Ludovico, E. Pagani, and N. Scarabottolo, "5G technology for augmented and virtual reality in education," Proceedings of the international conference on education and new developments, vol. 2019, pp. 512–516, 2019.