

# DiagramVoice: Automatic Lecture Video Commentator for Visually Impaired Students Supporting Diagram Commentary



Nahyun Eun  and Jongwoo Lee 

**Abstract** Recently, due to the COVID-19 pandemic, the utilization of non-face-to-face lectures has increased. Online lectures use a lot of visual materials, and blind college students experience difficulties in understanding the lecture content. This causes them to lose concentration and interest in the class. This paper proposes an automatic lecture video commentary system, “DiagramVoice”, to ensure an independent learning environment for visually impaired students. It is designed as a mobile application to be easily accessible to the visually impaired. It can provide analysis of text and nature photographs among visual materials in lecture videos and also provides commentary on diagrams through the diagram commentary generation algorithm developed in this paper. To verify the practical usability of our DiagramVoice, we conducted user tests on the results of the diagram commentary generation algorithm. <https://github.com/nan0silver/DiagramAnalysisGenerationAlgorithm>.

**Keywords** Online lecture · The visually impaired · Diagram commentary

## 1 Introduction

The number of online classes has increased dramatically in recent years. This trend is related to the COVID-19 pandemic. Due to the spread of COVID-19, schools, universities, and other institutions have been closed and replaced with non-face-to-face learning. The education system has undergone a major transformation in this

---

N. Eun · J. Lee (✉)

Department of IT Engineering, Sookmyung Women’s University, Cheongpa Ro 47 Gil 100, Seoul 04310, South Korea

e-mail: [bigrain@sm.ac.kr](mailto:bigrain@sm.ac.kr)

N. Eun

e-mail: [dmsskgus@sookmyung.ac.kr](mailto:dmsskgus@sookmyung.ac.kr)

process, and online classes have played an important role in this environment [1]. As more and more people are exposed to e-learning, various research have been conducted to make online classes more effective [2, 3].

However, visually impaired students who have difficulty conveying visual information in these digital learning environments face barriers to access in online class participation. Park et al. surveyed on the experiences of students with disabilities due to the COVID-19 pandemic and found that visually impaired college students experienced physical fatigue, decreased learning efficiency, and difficulty forming and maintaining social relationships due to virtual classes [4]. They have limited access to instructor guidance or chatting with companion students in an online environment, even though they have difficulty in receiving supplemental instruction. Assistive technologies are currently being used ostensibly for online education, such as braille displays and tactile haptic devices, but their deployment without proper training has been shown to hinder. Without additional support for these issues, already marginalized visually impaired students risk becoming increasingly unskilled and devalued [5]. Research has shown that students with visual impairments do not differ in cognitive abilities from their sighted peers [6], and that they are capable of mastering advanced academic concepts given sufficient support systems [7]. To ensure an independent learning environment for the visually impaired students, in this paper, we propose an automatic lecture video commentary system for the visually impaired.

This paper presents an online lecture-assisted commentary system that automatically detects, analyzes, and comments on visuals in online lectures and delivers them to the visually impaired. Currently, researches on automatic recognition and commentary of visual materials such as texts and pictures are advancing rapidly. In contrast, research on diagrams has been relatively underdeveloped and unnoticed.

A diagram is a human-designed illustration of relationships between objects to represent information in the form of connections from one node to another using arrows or connecting lines [8]. It is used in many academic disciplines to visually represent and understand important concepts. While understanding natural images has been a major area of research in computer vision, understanding illustrations to convey information has received little attention. Because diagrams focus on the relationships between visual objects, they allow for deeper inferences than natural images [9, 10]. However, visually impaired people have difficulty in understanding and utilizing learning materials, including diagrams. In this paper, to respond to these challenges, we developed an automatic online lecture assistant commentary system including diagram commentary.

The overall organization and content of this paper are as follows. Section 2 describes the currently used digital material description tools and diagram description techniques for the visually impaired. Section 3 describes the overall structure of DiagramVoice, the automatic lecture video commentary application for the visually impaired proposed in this paper. Section 4 introduces the diagram commentary technique, which is the core of DiagramVoice, and Sect. 5 presents an evaluation of the technique.

## 2 Related Works

### Digital Material Description for the Visually Impaired

Visually impaired students use tactile and kinesthetic input to understand information. Therefore, most visually impaired students rely on Braille to receive and convey information from learning materials. The Braille'n Speak is a portable device equipped with a speech synthesizer, a Braille keyboard, and interfacing capabilities. Users can input text using a Braille keyboard, which is stored in digital format. Conversely, stored information can be output as speech by the Braille'n Speak [11]. Various Braille display devices are available, including Refreshabraille from APH and Brailliant from HumanWare; however, a common drawback lies in the substantial cost burden associated with these devices [12]. This limits their universal use by many students.

Fichten et al. surveyed of e-learning accessibility among Canadian university students with low vision and blindness [13]. They found that 100% of the visually impaired and 50% of low-vision students use screen-reading technologies, and 90% of the visually impaired and two-thirds of low-vision students use optical character recognition (OCR). Screen-reading technologies are software that reads what is on the screen, while OCR is technology that automatically converts scanned or printed text images or handwritten text into editable text [14]. However, portable digital format (PDF) files with underlined or multi-column tables and shapes can confuse screen readers when rendered, making them difficult for users to interpret. In addition, PowerPoint, a popular e-learning resource on campuses, has embedded materials and text boxes that screen readers cannot read, making it difficult for students to use appropriately.

### Diagram to Text Generation

Prior work on diagram interpretation techniques focuses on interpreting visual material in various ways and converting it into an understandable form. In [9], a diagram parsing graph (DPG) is introduced as a method for modeling diagram structure, and a method for parsing the syntax of diagrams and semantic interpretation of diagrams by learning to reason over the graph is studied. Using artificial intelligence techniques, they devise an LSTM-based method for parsing diagrams and interpreting diagrams through diagram question-and-answer data. In [15], a study was conducted to convert block diagrams into text. To extract contextual meaning from diagram images, they proposed a framework called “BloSum” that uses CNNs to detect shapes, text, and arrows and recognize objects pointed by arrows to combine text. To conduct this research, a diagram dataset was specially built and experimented with, which is referenced in this paper.

### Prior Research

This paper is a follow-up study of the Design of Automatic Online Lecture Video Commentator for Visually Impaired Students Supporting Diagram Commentary [16]. In [16], we conducted a usability evaluation for effective diagram commentary for visually impaired students. Based on this evaluation, we proposed the design of an automatic lecture video content commentator that supports diagram commentary. We

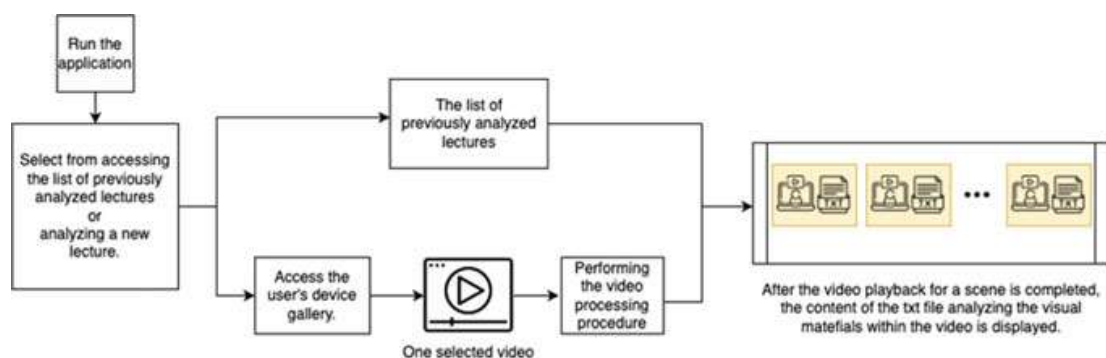
created three levels of commentary detailing the commentary method and evaluated them with visually impaired students and found that the simplest commentary and the most detailed commentary were the most satisfactory. This confirmed that it is ideal to implement a system that allows students to set and adjust the level of commentary by themselves.

### 3 The Comprehensive System Architecture of the DiagramVoice

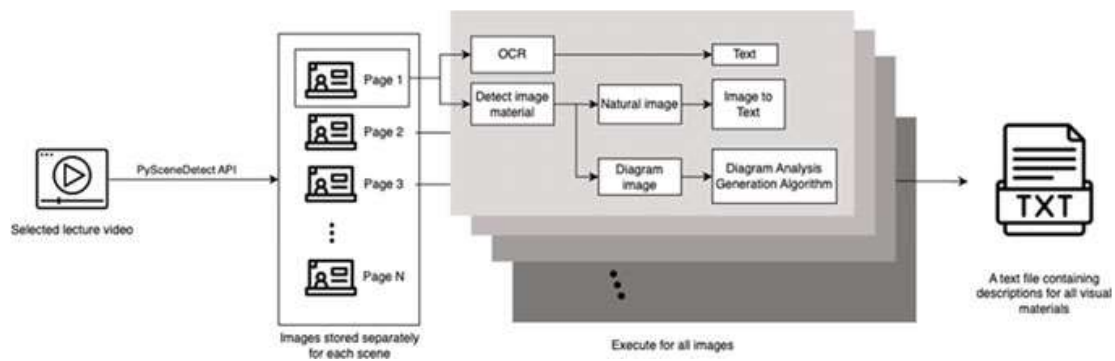
In this paper, we propose a mobile application “DiagramVoice”, an automatic lecture video commentator for the visually impaired. DiagramVoice is developed in Dart language [17] using Flutter [18] framework and can run on Android and IOS. The server is developed in Flask [19] using Python [20]. Figure 1 shows the overall application execution process.

DiagramVoice runs by recognizing the user’s voice. Every step is explained to the user through sound, and the application is designed to run only through sound. After launching the application, the user can access a list of previously analyzed lecture videos or analyze a new one. To select a new lecture video, the app accesses the user’s gallery and reads the title of the video. The user says which video they want, and the video is selected, and it goes through a video processing procedure. This process outputs a text file that analyzes the visuals in the video. When the user plays the video, at the end of each scene in the video, the contents of the text file analyzing the visuals in that scene are played out. If the user selects a lecture video from the list of lectures that have already been analyzed, the lecture video will play along with the text file of the visuals in the video that has already been saved.

The video processing procedure aims to find all the visuals in a video, analyze them, and put them into a single text file. Figure 2 illustrates the sequence of the process.



**Fig. 1** Overall application execution process



**Fig. 2** Overall structure of the video processing procedure

We assume that the video is a lecture video using PowerPoint, which is most used in university lecture classes. First, we use the PySceneDetect API [21] to detect screen transitions in the video. This API analyzes the image and determines that the screen changes when the rate of change of the scene image exceeds a certain threshold and outputs the times when the scene changes. Each time a scene changes in a video, it captures and stores the screen and uses these images to extract visuals. To extract text images, the second step is to perform OCR using the Naver Clova OCR API provided by the Naver Cloud Platform. The API supports Korean, English, and Japanese character recognition, and when we pass the image data to the API, it analyzes the image and provides the extracted text along with the location coordinates in the image in JSON format. In the third step, to detect the presence of a photograph or illustration in the scene image, a rectangle with a size greater than a certain percentage of the total image size is detected and the location coordinates are stored. The pictorial material on the screen is assumed to have a tag, and the closest letter tag is used to determine whether it is a photograph, diagram, or table. The fourth step is to convert the picture to text. If it is a natural image, we store a single sentence describing the image using the Microsoft Azure cognitive computer vision API [22], an image captioning API that uses an AI model that detects the main objects and situations in the image and automatically generates a single sentence describing them. If it is a diagram, we use the diagram captioning algorithm developed in this paper to generate diagram captioning sentences. Finally, we perform all of the above steps for every scene in the video and store the video's transition times, as well as the descriptions of all visuals on the screen, in a single text file. Using this commentary file, when the user plays the lecture video, the commentary on the visuals in the scene will be played out just before the lecture video cuts to the next scene. To do voice output, we use the Google Cloud Text-to-Speech API [23].



## 4 Diagram Analysis Generation Algorithm

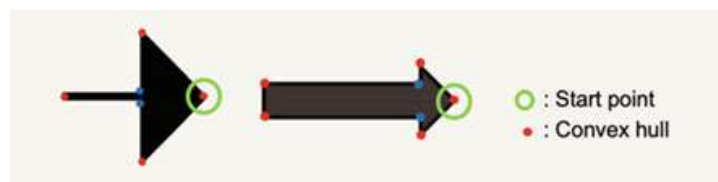
In this paper, a diagram commentary generation algorithm in Python is developed to convert diagram images into text, which aims to help visually impaired students accurately recognize diagrams and understand lecture content more effectively.

We start by preprocessing the diagram image. For image processing, we first convert the input image to grayscale and then blur the image using the GaussianBlur function. We then detect the edges of the image using the Canny function and apply the dilate and erode operations to enhance them. The processed image is then used for further analysis by highlighting features and removing noise. This image is then used to find contours in the image using the findContours function.

Next, the detected contours are used to determine whether they are arrows or rectangular objects that make up a block using the shape of the vertices. In the case of an arrow, as shown in Fig. 3, the shape has 4 or 5 convex hulls, depending on whether the arrow tail is composed of a line or a rectangle, and the total number of vertices differs by two from the number of convex hulls. The starting point of the arrow is determined by the point that is second from the non-convex hull vertex in the array where the vertex coordinates are stored. The endpoint is similarly determined by picking a coordinate to measure the value of the vector represented by the arrow. This allows us to determine the direction the arrow is pointing and the orientation of the overall diagram. This information will be used later to connect the blocks.

For rectangle detection, the algorithm looks for shapes that are not arrow-shaped and have four convex shells and determines the coordinates of each vertex. These coordinates are used to determine the upper-left point of the rectangle, as well as its width and height values, which are combined with the previous OCR information to connect the text within each block. Then, we find the square that is closest to the start and end points of all the arrows. Using the vector value of the arrow, the start points store the index value of the square whose center coordinate is the shortest distance from the start point among the squares in the direction the vector is pointing. The endpoints of the arrows perform the same method as above with the vector sign reversed. The index number of the rectangle is assigned based on the y-coordinate, if the overall diagram is vertically oriented. After finding out all the information of the arrow and the text information of the square, the diagram commentary is completed by connecting the sentence with the text information in the block. Figure 4 shows an example of these produces.

**Fig. 3** Features of an arrow



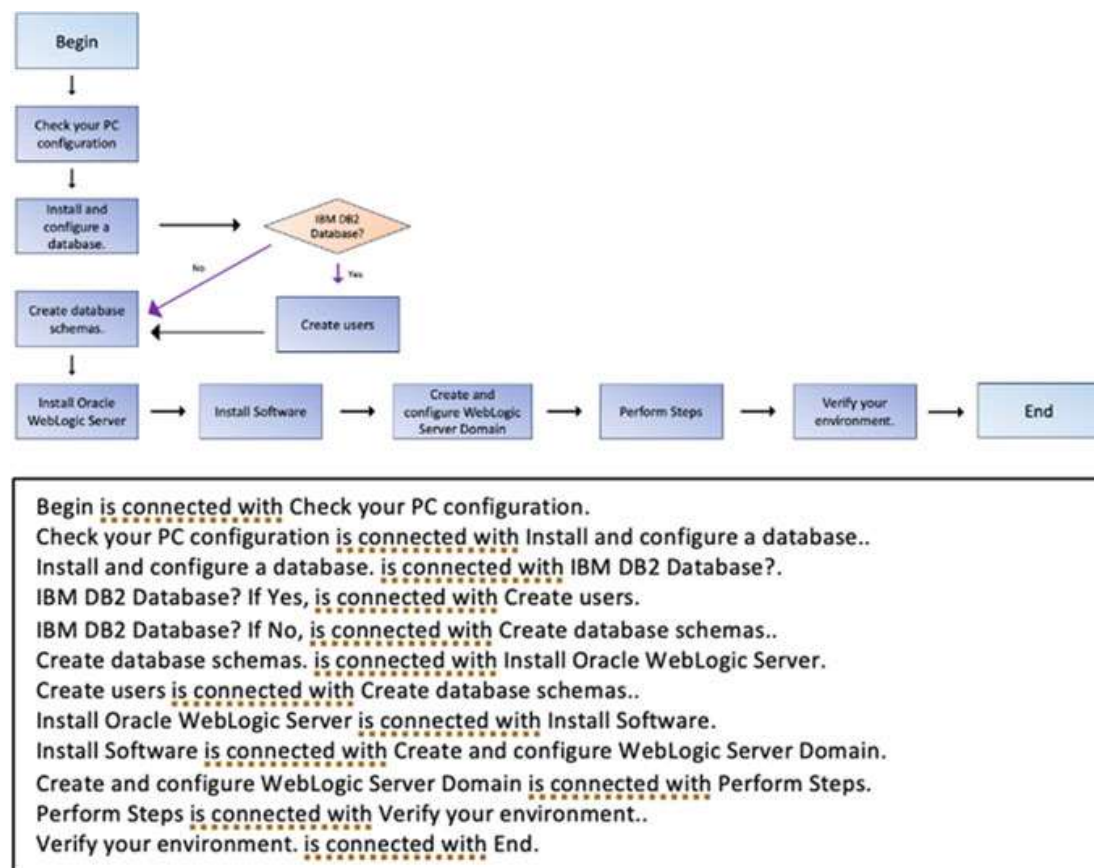


Fig. 4 Example of the diagram analysis generation algorithm results

## 5 Evaluation

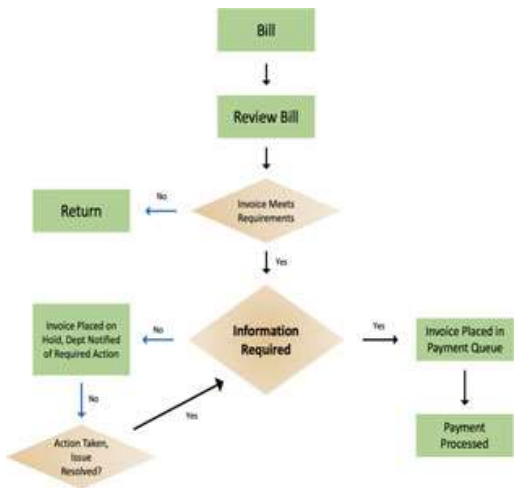

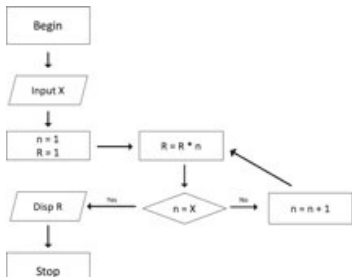
This chapter deals with the evaluation of the diagram annotation techniques described in Sect. 4. The diagram annotation algorithm is the core part of this research, and we want to evaluate the accuracy of the algorithm.

Bhushan and Lee [15] discuss function similar to the diagram commentary function proposed in this paper. While Bhushan and Lee [15] developed a diagram commentary technology using machine learning, we utilized an original algorithm in the Python language. Our algorithm was evaluated by five university students. In this evaluation, the diagram commentary generated by our algorithm and the commentary simply generated using the Image Caption API that performs image-to-text were listened to for each of the 10 diagrams, and the degree of adequacy and satisfaction were evaluated out of 10. We also tested the system using the diagrams' commentary answer data to determine its accuracy. This allowed us to measure the algorithm's performance and user response quantitatively and qualitatively. The results of the test are shown in Table 1, and some of the data we used to run the test is shown in Table 2.

**Table 1** Average score of the usability evaluation for diagram interpretation

Method	User test		System test
	Satisfaction	Adequacy	Accuracy
Image caption	3.1	2.5	0.9
The algorithm proposed in this paper	8.3	9.0	9.8

**Table 2** Results obtained by applying [15]'s dataset to the algorithm proposed in this paper

 <pre> graph TD     Bill[Bill] --&gt; ReviewBill[Review Bill]     ReviewBill --&gt; InvoiceMeets{Invoice Meets Requirements}     InvoiceMeets -- No --&gt; Return[Return]     InvoiceMeets -- Yes --&gt; InformationRequired{Information Required}     InformationRequired -- Yes --&gt; InvoicePlacedQueue[Invoice Placed in Payment Queue]     InvoicePlacedQueue --&gt; PaymentProcessed[Payment Processed]     InformationRequired -- No --&gt; InvoicePlacedHold[Invoice Placed on Hold, Dept Notified of Required Action]     InvoicePlacedHold --&gt; ActionTaken{Action Taken, Issue Resolved?}     ActionTaken -- Yes --&gt; InformationRequired     ActionTaken -- No --&gt; Return </pre>	<p>Bill is connected with Review Bill</p> <p>Review Bill is connected with Invoice Meets Requirements</p> <p>Invoice Meets Requirements. If No, is connected with Information Required</p> <p>Invoice Meets Requirements. If Yes, is connected with Return</p> <p>Information Required. If Yes, is connected with Invoice Placed in Payment Queue</p> <p>Information Required. If No, is connected with Invoice Placed on Hold, Dept Notified of Required Action</p> <p>Invoice Placed on Hold, Dept Notified of Required Action. If No, is connected with Action Taken, Issue Resolved?</p> <p>Invoice Placed in Payment Queue is connected with Payment Processed</p> <p>Action Taken, Issue Resolved? If Yes, is connected with Information Required</p>
 <pre> graph LR     Begin[Begin] --&gt; AlarmRings[Alarm Rings]     AlarmRings --&gt; ReadyToFace{Ready to face the world?}     ReadyToFace -- No --&gt; HitSNOOZE[Hit SNOOZE]     HitSNOOZE --&gt; Ignore[Ignore]     Ignore --&gt; AlarmRings     ReadyToFace -- Yes --&gt; GetUp[Get Up]     GetUp --&gt; End[End] </pre>	<p>Begin is connected with Alarm Rings</p> <p>Alarm Rings is connected with Ready to face the world?</p> <p>Ignore is connected with Alarm Rings</p> <p>Ready to face the world? If No, is connected with Hit SNOOZE</p> <p>Ready to face the world? If Yes, is connected with Get Up</p> <p>Hit SNOOZE is connected with Ignore</p> <p>Get Up is connected with End</p>
 <pre> graph TD     Begin[Begin] --&gt; InputX[/Input X/]     InputX --&gt; Init["n = 1 R = 1"]     Init --&gt; Calc["R = R * n"]     Calc --&gt; Decision{"n = X"}     Decision -- No --&gt; Calc     Decision -- Yes --&gt; DispR[/Disp R/]     DispR --&gt; Stop[Stop] </pre>	<p>Begin is connected with Input X</p> <p>Input X is connected with <math>n = 1</math> <math>R = 1</math></p> <p><math>n = 1</math> <math>R = 1</math> is connected with <math>R = R * n</math></p> <p><math>R = R * n</math> is connected with <math>n = X</math></p> <p><math>n = X</math> If No, is connected with <math>n = n + 1</math></p> <p><math>n = X</math> If Yes, is connected with Disp R</p> <p><math>n = n + 1</math> is connected with <math>R = R * n</math></p> <p>Disp R is connected with Stop</p>



The results of these evaluations show that both satisfaction and adequacy scores are higher when using the algorithm proposed in this paper than when simply using the image caption API. The system test results also show that our algorithm scored significantly higher. These results strongly suggest that our algorithm can provide a better educational experience for learners, allowing them to better comprehend a variety of visual materials and increasing the efficiency and effectiveness of online education.

## 6 Conclusion

In this paper, we propose a mobile application that can explain visuals in online lectures and playback lectures for visually impaired students. The visual materials in the lecture include text, images, and diagrams, and an automatic diagram commentary generation algorithm is proposed. In this way, students with visual impairments who have difficulty in accessing visual materials, including diagrams, independently can be interested in the lecture content and understand the content more effectively.

The application was developed to meet the special needs of contactless education. It follows the growing trend of online education around the world due to the COVID-19 pandemic and provides an environment for visually impaired students to optimize their learning experience and engage with their education. In particular, by accurately narrating visuals from lecture videos and outputting them to voice, visually impaired students can enjoy visual information and build an independent learning environment at the same time.

The technology resulting from this paper is not limited to university classes. It can also be applied to a wide range of educational content, such as educational broadcasts and television programs. This will contribute to bridging the divide in education and helping all students have a better learning experience. Our research is a step toward a better future in education, as we hope to contribute to a new educational paradigm and expand educational opportunities for visually impaired students.

**Acknowledgements** This research was supported by the Ministry of Science and ICT (MSIT), Korea, under the ICT Challenge and Advanced Network of HRD (ICAN) program (IITP-2023-RS-2022-00156299) supervised by the Institute of Information & Communications Technology Planning & Evaluation (IITP). This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIT) (No.2022R1F1A1063408). This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIT) (No. 2022H1D8A3037394).

## References

1. Akram F, Ul Haq MA, Malik HA, Mahmood N (2021) Effectiveness of online teaching during COVID-19. In: 2021 international conference on innovation and intelligence for informatics, computing, and technologies (3ICT), Zallaq, Bahrain, pp 568–573. <https://doi.org/10.1109/3ICT53449.2021.9582144>
2. Abrahamsson S, Dávila López M (2021) Comparison of online learning designs during the COVID-19 pandemic within bioinformatics courses in higher education. *Bioinformatics* 37:I9–I15. <https://doi.org/10.1093/bioinformatics/btab304>
3. Nyarko E, Agyemang EF, Arku D (2023) COVID-19 and online teaching in higher education: a discrete choice experiment. *Model Assisted Stat Appl* 18(2):115–123
4. Park J (2020) The reality and problems of non-face-to-face instruction according to the COVID-19 situation from the perspective of college students with disabilities. *Special Educ Res* 19(3):31–53. <https://doi.org/10.18541/ser.2020.08.19.3.31>
5. Hickson A et al (2022) Accessing and delivering online education in the time of COVID-19: challenges for visually impaired people in Malaysia. *Horizon J Hum Soc Sci Res* 4(1):63–71
6. Kumar D, Ramasamy R, Stefanich GP (2001) Science for students with visual impairments: teaching suggestions and policy implications for secondary educators. *Electron J Res Sci Math Educ*
7. Jones MG, Minogue J, Oppewal T, Cook MP, Broadwell B (2006) Visualizing without vision at the microscale: students with visual impairments explore cells with touch. *J Sci Educ Technol* 15(5):345–351
8. Kolis M, Kolis BH (2016) Thinking diagrams: processing and connecting experiences, facts, and ideas. Rowman & Littlefield
9. Larkin JH, Simon HA (1987) Why a diagram is (sometimes) worth ten thousand words. *Cogn Sci* 11(1):65–100. <https://doi.org/10.1111/j.1551-6708.1987.tb00863.x>
10. Kembhavi A et al (2016) A diagram is worth a dozen images. In: Computer vision–ECCV 2016: 14th European conference, Amsterdam, The Netherlands, 11–14 Oct 2016, proceedings, Part IV 14. Springer International Publishing
11. Sahin M, Yorek N (2009) Teaching science to visually impaired students: a small-scale qualitative study. *Online Submission* 6(4):19–26
12. Perkins School for the Blind. An overview of Braille devices. <https://www.perkins.org/resource/overview-braille-devices/>
13. Fichten CS et al (2009) Accessibility of e-learning and computer and information technologies for students with visual impairments in postsecondary education. *J Vis Impairment Blindness* 103(9):543–557
14. Patel C, Patel A, Patel D (2012) Optical character recognition by open source OCR tool tesseract: a case study. *Int J Comput Appl* 55(10):50–56
15. Bhushan S, Lee M (2022) Block diagram-to-text: understanding block diagram images by generating natural language descriptors. In: Findings of the Association for Computational Linguistics: ACL-IJCNLP 2022
16. [Accepted] Eun N, Lee J (2024) Design of automatic online lecture video commentator for visually impaired students supporting diagram commentary. In: Information systems for intelligent systems: proceedings of ISBM 2023. Springer Nature Singapore, Singapore. Available: <https://url.kr/hokjlt>
17. Dart [Online]. Available: <https://dart.dev/>. Accessed: Oct 2023
18. Flutter [Online]. Available: <https://flutter.dev/>. Accessed: Oct 2023
19. Flask [Online]. Available: <https://flask.palletsprojects.com/en/2.3.x/>. Accessed: Oct 2023
20. Python [Online]. Available: <https://www.python.org/>. Accessed: Oct 2023
21. PySceneDetect API [Online]. Available: <https://pyscenedetect.readthedocs.io/en/latest/ref-erence/python-api/>. Accessed: Oct 2023

22. Microsoft Azure cognitive computer vision API [Online]. Available: <https://azure.microsoft.com/ko-kr/>. Accessed: Oct 2023
23. Google Cloud Text-to-Speech API [Online]. Available: <https://cloud.google.com/text-to-speech>. Accessed: Oct 2023