https://doi.org/10.65025/ICAIC25215m

Prashaansh : Automatic Question Paper Generator

Ashwini Matange
Department of Computer Engineering, Pimpri Chinchwad College of Engineering, Savitribai Phule Pune University, Pune, India.
ashwini.matange@pccoepune.org

Ashirwad Katkamwar
Department of Computer Engineering, Pimpri Chinchwad College of Engineering, Savitribai Phule Pune University, Pune, India.

Harshad Karale
Department of Computer Engineering, Pimpri Chinchwad College of Engineering, Savitribai Phule Pune University, Pune, India.

Aviraj Kale
Department of Computer Engineering, Pimpri Chinchwad College of Engineering, Savitribai Phule Pune University, Pune, India.

Nayan Keote
Department of Computer Engineering, Pimpri Chinchwad College of Engineering, Savitribai Phule Pune University, Pune, India.

Abstract

Prashnaansh is designed to streamline and optimize the creation of well-structured question papers. By leveraging predefined question banks and intelligent algorithms, it ensures balanced mark distribution, appropriate difficulty levels, and comprehensive syllabus coverage. The system adheres to various guidelines, ensuring compliance with standardized educational frameworks. Its user-friendly interface simplifies the process, significantly reducing the time and effort required for manual paper setting. By eliminating inconsistencies, enhancing efficiency, and ensuring fairness in assessments, Prashnaansh serves as an invaluable tool for educators and examination authorities.

1. Introduction

The process of question paper generation in educational institutions heavily depends on manual efforts, requiring educators to carefully create question papers that satisfy conditions such as balancing the difficulty level of the questions, ensuring appropriate weightage for each topic in the syllabus, covering the syllabus comprehensively, and aligning with the type of exam being conducted, among other considerations. This is a time-intensive task that requires the manual efforts of many educators for a single question paper, and this effort must be repeated for each subject. Despite these efforts, the process is still prone to human errors, inconsistencies, and biased question papers. Additionally, as the process involves multiple individuals, there is a constant threat of paper leaks. The dependency on individuals also increases the risk of mistakes, which can compromise the fairness and quality of the examination. To tackle these issues and with the increasing reliance on technology in education, there is a growing demand for automating the question paper generation process. Automation can reduce the workload on educators while maintaining the quality of question papers and ensuring standardized and unbiased question paper generation. Our proposed system for automated question paper generation utilizes advanced algorithms and techniques to generate unbiased and balanced question papers. It ensures comprehensive topic coverage, alignment with course outcomes, and transforms the manual process into an efficient and reliable one. By minimizing human intervention, the system aims to make the entire process more reliable and fairer. The system is easy to use, anyone can generate a balanced question paper even if that person does not have any background knowledge of the particular subject of which the question paper is getting generated. To deal with these pressing challenges we would like to propose

Prashnaansh: An Intelligent Question Paper Generator, an unrivaled and user-centric platform designed, implemented and delivered to automate the generation of question papers. Prashnaansh features an organized question bank and uses intelligent logic of question selection, and so ensuring proportional weightage, varying difficulty and balance with complete coverage of the entire syllabus. The most significant change is that Prashnaansh can dynamically adjust to initial academic defined assumptions, course outcomes and institutional quality standards, moving beyond the limitations imposed by conventional systems. As well as decreasing human dependence, by way of embedding quality assurances in hierarchy of the generation logic and defining process, Prashnaansh is able to simplify the workflow for educators, whilst also improving and assuring the security of the examination process. Importantly, the user-friendly interface allows users - including those with no specialization knowledge - to generate quality confirmed, structured and examination-ready question papers with rela- tively little input and effort.

2. Literature Survey

The process of question paper generation process has traditionally been a manual task. This task used to require the educator to carefully select the questions while maintaining a balance in difficulty levels, syllabus coverage, and mark distribution.

- Traditional Question Paper Setting Previously, question paper creation was dependent on educators manually curating questions from the question bank, previous examinations, and knowledge based on experience. Studies tell us the challenges faced in ensuring fairness, avoiding repetition, and maintaining syllabus coverage. Manual methods causes mistake and require significant time and effort (Smith et al., 2018).
- Automated generation of question papers using rule-based systems. Early attempts at automating question paper generation used rule-based systems, at that time questions were categorized based on predefined templates. Research by Gierl et al. (2012) introduced a framework for automated test assembly (ATA), this ensured a balanced distribution of difficulty levels. However, rule-based systems didn't have the adaptability, as they were heavily dependent on humans predefined constraints.
- Adaptive Learning and Intelligent Question Selection Currently the research is focusing on adaptive learning techniques, where AI dynamically selects questions based on a student's allover academic performance history. Studies by Wauters et al. (2010) and Shute and Rahimi (2021) discuss how intelligent tutoring systems analyze students previous responses to generate personalized question paper, ensuring balanced difficulty level and improved the overall outcomes.

3. System Architecture

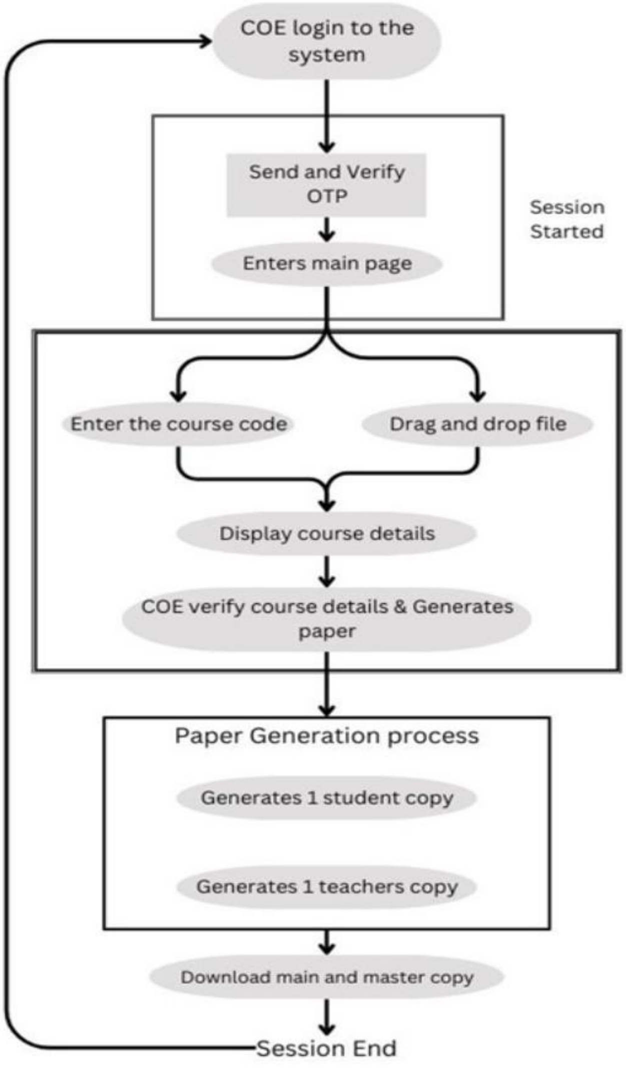


Fig. 1. System Architecture

4. Core Components of the System

The Prashnaansh system is designed with multiple core modules that work together to ensure the efficient, accurate, and unbiased generation of question papers. These components used predefined question banks, intelligent algorithms, and easy to use user interface to streamline the process. The key components of the system are as follows:

Question Bank Management: The system maintains a well-organized Question bank that stores a wide range of questions categorized based on different criteria like:

- Unit: Questions from all the units ensure the complete syllabus coverage.
- Difficulty Level: Questions are categorized as easy, medium, and hard to create well balanced question pa- pers.

- Marks Allocation: Predefined marks to each question maintain proper weight distribution in question paper.
- Question Type: There are two main types of questions theory and numerical.
- Bloom's Taxonomy Level: Classifies questions based on cognitive skills such as Remember, Understand, Apply, Analyze, Evaluate, and Create. This ensures a balanced assessment.
- Course Outcomes (CO): Links each question to predefined learning outcomes, ensuring that assessments align with the expected skills for the course.

5. Implementation Details

Algorithm Description

This system's algorithm is developed as a complex multiple process aimed at efficiency, fairness, and adaptability. The advanced techniques of question selection, fitness evaluation, and document generation integrate it into an end-to-end solution to its end product for the question paper

Key Components

i. **System Initialization:** The backend of the system is built using the Flask framework. During initialization, essential environment variables such as the JWT secret key and file upload paths are loaded for secure configuration. Middleware components are also set up to handle reverse proxies and enable Cross-Origin Resource Sharing (CORS), ensuring the system is compatible with standard web protocols.

i. **Authentication and Access Control:** To safeguard the system, all API endpoints are secured using JSON Web Tokens (JWT). This mechanism authenticates users and authorizes access, allowing only verified faculty members to utilize the system's functionalities, thereby preventing unauthorized interventions.

ii. **File Upload and Validation:** Faculty members are required to upload an Excel- based question bank and may optionally upload a Word document template for formatting purposes. The system validates the uploaded files in terms of format (e.g.,.xlsx, .xls) and structure, ensuring consistency with the expected schema before further processing.

iii. **Data Extraction and Preprocessing:** Upon successful validation, the system extracts key parameters from the uploaded Excel file, including unit- wise marks distribution, overall marks allocation, and the theory-to-numerical question ratio. The question bank is parsed and categorized according to unit, difficulty level (easy, medium, hard), and question type (theoretical or numerical).

iv. **Question Selection Algorithm:** The core logic iteratively selects questions for each unit to fulfill predefined constraints such as total marks and balance across difficulty levels. The selection targets a distribution of:

- Easy: 40% ± 5%
- Medium: 40% ± 5%
- Hard: 20% (or the remaining balance) Additionally, the algorithm adheres to the specified theory-to-numerical ratio. If a valid set of questions is not found, a backtracking mechanism is employed to retry the selection process up to a defined limit of attempts.

v. **Fitness Evaluation:** Each generated question paper is evaluated using a fitness function that considers:

- Alignment with the total target marks
- Correct distribution of difficulty levels
- Adherence to the theory-to-numerical ratio
- Papers are ranked based on their fitness scores, and the top three are retained. One of these is then randomly selected to ensure variability while maintaining quality.

vi. **Document Generation:** Two distinct Word documents are generated:

- Faculty Copy: Contains metadata such as Bloom's taxonomy levels, Course Outcomes (COs), question difficulty levels, and mark allocations.
- Student Copy: A clean version with only the finalized questions, suitable for direct use in examinations.

The documents are formatted with institution-specific placeholders, unit headers, and standardized layout elements.

vii. **Output Packaging:** The generated documents are compressed into a ZIP file. This file is then provided as a downloadable response, facilitating easy access and storage.

viii. **Error Handling and Logging:** The system implements robust error handling, returning descriptive error messages for issues related to file format, authentication, or question selection failures. Additionally, all API requests and responses are logged systematically for auditing and debugging purposes.

Enhanced Features: The proposed system, Prashnaansh, incorporates several enhanced features aimed at streamlining the question paper generation process while ensuring flexibility, reliability, and academic rigor. The following outlines the core enhancements and the sequential operations executed during the generation process.

- One-Click Operation: The system is designed for maximum ease of use, allowing faculty members to generate complete, syllabus- aligned question papers within seconds using a single click, eliminating the need for extensive configuration.
- Customizable Patterns: The system accommodates institutional diversity by allowing users to define unique patterns, including varying total marks, section-specific weightages, and exam- specific configurations, making it adaptable across different academic boards and universities.
- Data Integrity Assurance: All generated questions are strictly selected from the uploaded Excel-based question bank. No external or unverified content is introduced, thereby maintaining the academic authenticity and relevance of the paper.

Operational Workflow

The system follows a well-defined pipeline of operations, ensuring secure, efficient, and reliable generation of question papers:

- Authentication and Access Control: Each request to the system is authenticated using a JWT token. The token is validated at the API level, and any request with a missing or invalid token is immediately rejected to prevent unauthorized access.
- File Handling and Validation: Upon authentication, the system accepts an Excel file containing the question bank and an optional Word template. Uploaded files undergo validation checks to confirm their format and structure. The system then parses the question bank and extracts associated meta- data, including marks distribution, question categories, and formatting information.
- Question Paper Generation: The generation algorithm begins by initializing a unit- wise marks allocation map. For each unit in the syllabus:
  - Questions are filtered based on marks requirements and difficulty constraints.
  - Randomized selection is applied to maintain question variability while ensuring adherence to total marks and unit weightages.
- Fitness-Based Evaluation: Once a draft paper is generated, it undergoes a fitness evaluation against preset criteria such as:
  - Correct total marks
  - Difficulty level distribution (Easy: 40% ± 5%, Medium: 40% ± 5%, Hard: 20%)
  - Theory-to-numerical ratio

    If the generated paper meets all criteria, the system proceeds to the next step. If not, the algorithm re- tries selection within a capped number of attempts and ultimately selects the best-fit paper based on the fitness score.
- Document Generation: Upon finalizing a question paper, the system dynamically populates a Word template:
  - Metadata such as marks, Bloom's taxonomy levels, and Course Outcomes (COs) are injected for the faculty copy.
  - The student copy includes only cleanly formatted questions.

All tables, unit headers, and sections are formatted for clarity and institutional compliance.

- Output Generation and Cleanup: The final output consists of both document versions compressed into a ZIP file. The system then provides this ZIP file as a downloadable asset to the user. Temporary files are securely deleted post-processing to optimize server storage and maintain user data privacy.

Security

It is one of the prime goals of this software to make the system safe from possible threats by keeping it safe. We have developed various precautionary techniques for preventing malicious activities against the software as well as unapproved access to it. Such measures make sure that a user has a secure and reliable environment. This ensures confidentiality as well as integrity of data.

- **Authentication:** In our software, a two-step verification process has been implemented to ensure that only authorized individuals can access the system. The first step requires users to provide login credentials that are specific to their account. Once the credentials are verified, an OTP (One-Time Password) is sent to the registered email address of the user. In this step, the user inputs the OTP within the time limit provided. Access is allowed only if the OTP is verified successfully. The strong authentication process minimizes unauthorized access and ensures that only the authorized users are allowed to continue.

- **Token System:** After authenticating the user, the system creates a JWT for each user. This token is the key to giving access to main features of the software. For example, users can only generate papers if they have a valid token. Moreover, every token has an expiration mechanism to enhance security. Tokens automatically get invalidated once used or after a certain time period, minimizing the risk of token misuse. This method ensures safe management of sessions and mitigates the occurrence of unethical practices, such as token theft or unauthorized usage.
- **Disabled keys and devtools:** For piracy and unauthorized access to sensitive content, some keyboard functionalities and browser tools have been disabled. Key combinations, for instance, Ctrl

+ C (copy), have been disabled to ensure users cannot copy content. Moreover, no parts of the data can be chosen or marked to protect it even more. In addition, the Developer tools and inspect mode are also disabled, thus nothing else except authorized code or data is allowed to access or alter. And the feature for recording screen or screenshotting any data is totally forbidden to avoid leakage of information. All these measures ensure sensitive information, be it question papers or user data, is kept secure and not pirated or leaked.

- **IP restriction and logs:** We have included a section on IP restriction and logs. There, we described how authorized IPs are used for request logging through MongoDB in concise words.

It will integrate advanced authentication methods, token-based access control, and restriction on unauthorized actions for maximum protection. Addressing all key areas of vulnerability and implementation of robust preventive measures ensures that the system remains intact and establishes trust among the users. Security is not just an add-on feature but is deeply embedded into every design aspect of the software.

Logs

Logs are one of the most prominent features of our software, tracking how the system and the activity of the user has been utilized. This feature adds transparency to the system, so it's easy to detect problems and fix them more efficiently.

The following information for every request that gets made to the system is safely logged and stored on the MongoDB server:

- IP Address: The source IP address from where the request originated.
- Information of User: Information about who requested the page.
- Device details: Information related to the user's device, including his or her browser and operating system.
- Reason for Access: Reason behind the user's request, such as to log in or print a paper.
- Error Details: Errors when trying to complete an action.

In addition to logging, we provide log management features to further enhance user experience. We have search functionalities where administrators can look for certain entries in the logs or apply filters to narrow down results. Examples of filters based on logs are:

- Status: Success or error logs.
- Task Type: Category of login attempts, paper generation, or any other system activity.

These features help administrators gain insights into usage by patterns that may help respond to anomalies or problems in the system. Our software maintains detailed logs, which enables successful log management and therefore a safe environment for everyone.

VI. Results & Evaluation

A. A Performance Analysis

[1] The Prashnaansh system was evaluated based on key parameters such as efficiency, accuracy, customization, and optimization. The following results were observed after multiple tests and faculty feedback

Efficiency – Time and Effort Reduction Traditional manual question paper generation is a time-intensive process, requiring 2–5 hours per paper. With Prashnaansh, the same task is completed in less than a minute, reducing time and effort by 80–90

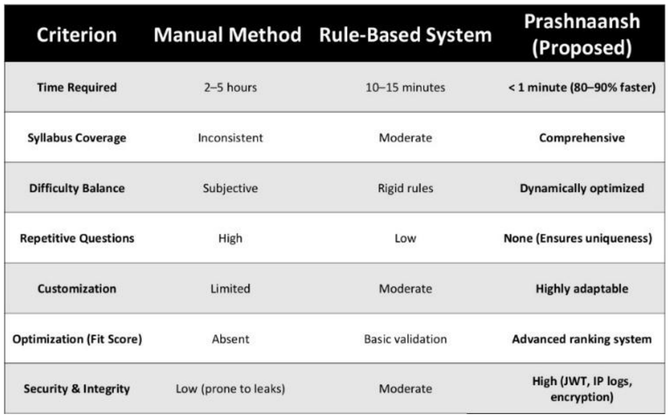
Accuracy – Balanced and Fair Paper Generation The system ensures minimal human intervention, eliminating issues like bias, inconsistency, and errors. The generated papers were verified for compliance with syllabus distribution, difficulty levels, and mark allocation. The system maintained a ±5

Customization – Adaptability to Various Educational Standards Prashnaansh is designed to be flexible and supports different educational boards, subjects, and levels. Users can configure question distribution, difficulty settings, and paper formats to align with institutional requirements. This adaptability allows it to cater to schools, colleges, and universities with diverse assessment structures.

Optimization – Ensuring High-Quality Papers The system integrates a fitness evaluation mechanism, ranking question papers based on adherence to syllabus, difficulty balance, and structural coherence. The Fit Score ensures that only well-structured, high-quality question papers are selected for final output.

B. Comparative Evaluation

A comparative analysis was conducted between manual, rule-based, and Prashnaansh methods:



Criterion	Manual Method	Rule-Based System	Prashnaansh (Proposed)
Time Required	2–5 hours	10–15 minutes	< 1 minute (80–90% faster)
Syllabus Coverage	Inconsistent	Moderate	Comprehensive
Difficulty Balance	Subjective	Rigid rules	Dynamically optimized
Repetitive Questions	High	Low	None (Ensures uniqueness)
Customization	Limited	Moderate	Highly adaptable
Optimization (Fit Score)	Absent	Basic validation	Advanced ranking system
Security & Integrity	Low (prone to leaks)	Moderate	High (JWT, IP logs, encryption)

Fig. 2. Comparative Evaluation

C. User Feedback and Adoption

A survey was conducted among 50 faculty members who tested Prashnaansh in a real-world college examination setting. The key observations were:

92 % of users found the system easy to use and required minimal training.

89 % reported a significant reduction in workload due to automation.

95 % acknowledged that the generated papers were well-balanced and fair.

87 % preferred Prashnaansh over manual methods because of its efficiency and accuracy.

V. Conclusion

The evaluation confirms that Prashnaansh effectively reduces time and effort by 80–90 %, ensures high accuracy, supports customization across educational institutions, and optimizes question selection through a Fit Score mechanism. The system's automation, security features, and adaptability make it a powerful tool for educators and examination authorities, transforming traditional paper-setting into a fast, reliable, and standardized process.

References

[1] Hsieh, I.-H.; Cheng, H.-C.; Ke, H.-H.; Chen, H.-C.; Wang, W.-J. ”A CNN-Based Wearable Assistive System for Visually Impaired People Walking Outdoors.” Applied Sciences 2021, 11, 10026. DOI: 10.3390/app112110026

[2] Likynas, D., Zvironas, A., Gudauskis, M., Budrionis, A., Daniusis, P., Sliesoraityte, I. ”Research Advances of Indoor Navigation for Blind People: A Brief Review of Technological Instrumentation.” IEEE Instrumentation Measurement Magazine, 2020, pp. 22-29. DOI: 10.1109/MIM.2020.9126068

[3] Chang, W.-J.; Chen, L.-B.; Hsu, C.-H.; Chen, J.-H.; Yang, T.-C.; Lin, C.-P. ”MedGlasses: A Wearable Smart-Glasses-Based Drug Pill Recognition System Using Deep Learning for Visually Impaired Chronic Patients.” IEEE Access, 2020. DOI: 10.1109/ACCESS.2020.2967400

[4] Husin, M. H., Lim, Y. K., School of Computer Sciences, Universiti Sains Malaysia, USM, Minden, Malaysia. ”InWalker: smart white cane for the blind.” Disability and Rehabilitation: Assistive Technology, 2019, pp. 1-7. DOI: 10.1080/17483107.2019.1615999

[5] Mainkar, V. V., Tamhankar, H. R., Bagayatkar, T. U., Jadhav, R. G., Shetye, S. K., Tendolkar, R. S. ”Raspberry Pi-Based Intelligent Reader for Visually Impaired Persons.” Proceedings of the Second International Conference on Innovative Mechanisms for Industry Applications (ICIMIA 2020), IEEE, 2020. ISBN: 978-1-7281-4167-1

[6] Ayachi, R., Afif, M., Said, Y., Abdelali, A. B. ”Traffic Sign Recognition Based on Scaled Convolutional Neural Network for Advanced Driver Assistance System.” 2020 IEEE 4th International Conference on Image Processing, Applications, and Systems (IPAS), IEEE, 2020. DOI: 10.1109/IPAS50080.2020.9334944

[7] Arifando, R., Eto, S., Wada, C. ”Improved YOLOv5-Based Lightweight Object Detection Algorithm for People with Visual Impairment to Detect Buses.” Applied Sciences, vol. 13, no. 5802, 2023. DOI: 10.3390/app13095802

[8] M. A. Khan, P. Paul, M. Rashid, M. Hossain, and M. A. R. Ahad, “An AI-based visual aid with integrated reading assistant for the completely blind,” IEEE Trans. Human-Mach. Syst., vol. 50, no. 6, Dec. 2020, doi: 10.1109/THMS.2020.3027534

[9] B. Kuriakose, R. Shrestha, and F. E. Sandnes, “Tools and technologies for blind and visually impaired navigation support: A review,” IETE Tech. Rev., 2020, doi: 10.1080/02564602.2020.1819893

[10] M. Gowda C.P., R. Hajare, and P. S. S. Pavan, “Cognitive IoT system for visually impaired: Machine learning approach,” Mater. Today Proc., 2021, doi: 10.1016/j.matpr.2021.03.666

[11] Y. Lei, S. L. Phung, A. Bouzerdoum, H. T. Le, and K. Luu, “Pedestrian lane detection for assistive navigation of vision-impaired people: Survey and experimental evaluation,” IEEE Access, Sep. 2022, doi: 10.1109/ACCESS.2022.3208128

[12] R. B. Islam, S. Akhter, F. Iqbal, M. S. U. Rahman, and R. Khan, “Deep learning based object detection and surrounding environment description for visually impaired people,” Heliyon, 2023, doi: 10.1016/j.heliyon.2023.e16924

[13] M. A. Adnan, N. Alam, F. Ahammed, I. A. Prince, M. F. Rahman, and M. T. Ahamed, “Raspberry Pi-powered IoT device for blind navigation using MobileNetV3-SSD image processing,” in Proc. Int. Conf. Inf. Commun. Technol. Sustain. Develop. (ICICT4SD), Sep. 2023, doi: 10.1109/ICICT4SD59951.2023.10303636

[14] M. N. Rithuan, A. S. Md Zain, and N. M. Nawawi, "Indoor navigation system for vision impairment people through visible light communications," in Proc. 1st Int. Symp. Eng. Technol. (ISETech), IOP Conf. Ser.: Mater. Sci. Eng., vol. 767, 2020, doi: 10.1088/1757-899X/767/1/012057

[15] J. Kilian, A. Neugebauer, L. Scherffig, and S. Wahl, "The unfolding space glove: A wearable spatio-visual to haptic sensory substitution device for blind people," Sensors, vol. 22, no. 5, 2022, doi: 10.3390/s22051859

[16] Y. Said, M. Atri, M. A. Albahar, A. Ben Atitallah, and Y. A. Alsariera, "Obstacle detection system for navigation assistance of visually impaired people based on deep learning techniques," Sensors, vol. 23, 2023, doi: 10.3390/s23115262

[17] J.-Y. Lin, C.-C. Yao, C.-L. Chiang, and M.-C. Chen, "Smart glasses application system for visually impaired people based on deep learning," in Proc. IEEE Int. Conf. Comput. Sci. Inf. Eng., 2020, doi: 10.1109/ICCSIE49998.2020.000

[18] A. R. See, B. G. Sasing, and W. D. Advincula, "A smartphone-based mobility assistant using depth imaging for visually impaired and blind," Appl. Sci., vol. 12, 2802, 2022, doi: 10.3390/app12062802

[19] E. Syahrudin, E. Utami, and A. D. Hartanto, "Enhanced Yolov8 with OpenCV for blind-friendly object detection and distance estimation," J. Rekayasa Sist. Teknol. Inf., vol. 8, no. 2, pp. 199–207, 2024, doi: 10.29207/resti.v8i2.5529

[20] M. D. Messaoudi, B.-A. J. Menelas, and H. Mcheick, "Review of Navigation Assistive Tools and Technologies for the Visually Impaired" Sensors, vol. 22, no. 20, 7888, 2022, doi: 10.3390/s22207888

[21] Y. Said, M. Atri, M. A. Albahar, A. Ben Atitallah, and Y. A. Alsariera, "Obstacle detection system for navigation assistance of visually impaired people based on deep learning techniques," Sensors, vol. 23, no. 11, 5262, 2023, doi: 10.3390/s23115262

[22] A. Lavric, C. Beguni, E. Zadobrischi, A.-M. Cailean, and S.-A. Avatamanitei, "A comprehensive survey on emerging assistive technologies for visually impaired persons: Lighting the path with visible light communications and artificial intelligence innovations," Sensors, vol. 24, no. 15, 4834, 2024, doi: 10.3390/s24154834

[23] S.-U. Maya-Martinez, A.-J. Arguelles-Cruz, Z.-J. Guzman-Zavaleta, and M.-d.-J. Ramirez-Cadena, "Pedestrian detection model based on Tiny-Yolov3 architecture for wearable devices to visually impaired assistance," Front. Robot. AI, Mar. 2023, doi: 10.3389/frobt.2023.1052509

[24] J. Li, L. Xie, Z. Chen, L. Shi, R. Chen, Y. Ren, L. Wang, and X. Lu, "An AIoT-based assistance system for visually impaired people," Electronics, vol. 12, no. 18, 3760, 2023, doi: 10.3390/electronics12183760

[25] N. Lakhani, H. Lakhotiya, and N. Mulla, "Be My Eyes: An Aid for the Visually Impaired," in Proc. 2022 IEEE 3rd Global Conf. Adv. Technol. (GCAT), Oct. 7-9, 2022, doi: 10.1109/GCAT55367.2022.9972160

[26] K. P. Ajitha Gladis, R. Srinivasan, T. Sugashini, and S. P. Ananda Raj, "Smart-YOLO Glass: Real-time video-based obstacle detection using paddling/paddling SAB YOLO network," in Alexandria Engineering Journal, vol. 77, 2023, doi: 10.1016/j