

Article

Towards Contactless Learning Activities during Pandemics Using Autonomous Service Robots

Anas Al Tarabsheh ^{1,2} , Maha Yaghi ¹ , AbdulRehman Younis ^{1,†}, Razib Sarker ^{1,†}, Sherif Moussa ^{1,†}, Yazeed Eldigair ^{1,†}, Hassan Hajjdiab ³, Ayman El-Baz ⁴  and Mohammed Ghazal ^{1,*} 

- ¹ Electrical, Computer and Biomedical Engineering Department, College of Engineering, Abu Dhabi University, Abu Dhabi 59911, United Arab Emirates; anas.altarabsheh@adu.ac.ae (A.A.); maha.yaghi@adu.ac.ae (M.Y.); 1059823@students.adu.ac.ae (A.Y.); 1065280@students.adu.ac.ae (R.S.); 1065669@students.adu.ac.ae (S.M.); 1074120@students.adu.ac.ae (Y.E.)
- ² Electrical Engineering Department, Faculty of Engineering, The Hashemite University, Zarqa 13115, Jordan
- ³ Computer Science Department, Concordia University, Montreal, QC H4B 1R6, Canada; hassan.hajjdiab@concordia.ca
- ⁴ Bioengineering Department, University of Louisville, Louisville, KY 40292, USA; aselba01@louisville.edu
- * Correspondence: mohammed.ghazal@adu.ac.ae
- † These authors contributed equally to this work.

Abstract: The COVID-19 pandemic has had a significant impact worldwide, impacting schools, undergraduate, and graduate university education. More than half a million lives have been lost due to COVID-19. Moving towards contactless learning activities has become a research area due to the rapid advancement of technology, particularly in artificial intelligence and robotics. This paper proposes an autonomous service robot for handling multiple teaching assistant duties in the educational field to move towards contactless learning activities during pandemics. We use SLAM to map and navigate the environment to proctor an exam. We also propose a human–robot voice interaction and an academic content personalization algorithm. Our results show that our robot can navigate the environment to proctor students avoiding any static or dynamic obstacles. Our cheating detection system obtained a testing accuracy of 86.85%. Our image-based exam paper scanning system can scan, extract, and process exams with high accuracy.

Keywords: robotics; service robots; autonomous navigation; simultaneous localization and mapping; natural language processing



Citation: Al Tarabsheh, A.; Yaghi, M.; Younis, A.; Sarker, R.; Moussa, S.; Eldigair, Y.; Hajjdiab, H.; El-Baz, A.; Ghazal, M. Towards Contactless Learning Activities during Pandemics Using Autonomous Service Robots. *Appl. Sci.* **2021**, *11*, 10449. <https://doi.org/10.3390/app112110449>

Academic Editors: Carlo Canali and Manuel Armada

Received: 31 August 2021

Accepted: 4 November 2021

Published: 7 November 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

With the recent advancements in computer vision and artificial intelligence, service robots are becoming increasingly involved in our daily lives. In the educational field, both students and instructors may benefit from the employment of robots. A robot may help students by acting as a tutor, and it can also help instructors by working as a teaching assistant [1]. One characteristic that makes a service robot promising as a teaching assistant is its ability to move and perform repetitive physical activities [2]. The primary duties of a teaching assistant include exam proctoring, communicating with the students, and delivering tutorials. However, with the widespread COVID-19 pandemic nowadays, people are more concerned about protecting their families, making them less willing to interact with others to avoid virus transmission.

With academic institutes adapting to the rules and regulations of social distancing due to the ongoing pandemic, online exams and their proctoring tools have been prevalent. These AI-proctored tests are a combination of AI and human proctors. A video of the student taking the exam is recorded using a webcam, and the AI flags any suspicious behaviors that the test-taker carries out. A human proctor then reviews the recording to verify whether the student was cheating during those flagged periods. Cheating events mainly include the student's face missing from the frame, multiple faces detected, and

an internet interruption. One of the limitations of such systems is that they only use the candidate's face to trigger flags. Some academic institutions have reverted to using eye-tracking equipment to combat cheating in exams further. The Eye Tribe is a device that can track eye movements and identify times at which the candidate's face was looking outside of the screen [3]. However, this solution is considered costly and difficult to implement on a large scale since all students will be required to have the tracking device set up in their testing area. Some advancements are also being made by using pose estimation techniques to detect cheating while taking online tests [4]. This allows for the detection of the head and the tilting of the head and the limbs of the test-taker. The different variations of the head poses can also indicate whether cheating has occurred or not.

Once an exam is over, the proctor would typically be required to collect the papers from the students to be handed over to the instructor for grading purposes. However, since papers are physical mediums, the coronavirus can be transmitted by simply touching or handling them. Therefore, scanning exam papers could be one of the solutions to avoid the exchange of papers. Several approaches in the literature propose scanning approaches. Scanning of documents using vision alone without dedicated hardware presents a few challenges to be solved. One of them is correctly detecting the borders or the corners of documents in the images, as described in [5]. Whether it is to fix the skew or properly extract the document page from an image, detecting the edges of a document is essential. Another problem is that when an image of a document is taken in an uncontrolled environment, it is inevitable that the paper will be skewed, which cannot be fixed with a simple crop. Ref. [6] describes a solution to this problem using Progressive Probabilistic Hough Transform. Ref. [7] outlines a "click-less" system where a video of the document to be scanned is recorded, and pages are then extracted and scanned without the need for any input from the user, saving time and effort.

A service robot can be used to display content based on the student's need to support students. Many companies use recommendation engines for many different types of content recommendations. For instance, they are used in the real world by services such as MovieLens for movie recommendations and social media sites such as Pinterest to recommend posts. As [8] explains, to recommend series and movies to a user, Netflix utilizes a complex recommender system (consisting of a variety of algorithms) that uses information such as what and how each member watches, un-played recommendations in a session, and where a video was found in Netflix itself. A recommender system could also be used in tag-based content discovery. Ref. [9] details how the similarity of users, measured from the tags they used, could be used with the relationships between the tags to recommend content for a user based on the semantic similarity of the tags of that content to that in the user's history. Ref. [10] shows that a recommender system can be used to recommend categories of news articles in real time based on the freshness of content using support vector machines (SVM), which outperforms the use of collaborative filtering. Ref. [11] introduces newer approaches to tag-based recommendations where the recommendation process incorporates tags, particularly higher-order singular value decomposition (HOSVD).

Several assistant robots have been proposed to aid in educational activities. Ref. [12] presents IDML tools that can provide teachers with a humanoid robot to assist them in engaging in classroom teaching activities. The results show that robots can enhance children's learning interests and have the potential to help elementary education. The authors of [13] developed a robot as a teaching assistant to offer teachers with the support they needed, and they used the robot teaching assistant to help teachers explain and illustrate two icebreaker games in particular. The findings show that education aided by a robot teaching assistant is more effective in enhancing student knowledge and assisting students in developing more favorable attitudes toward classroom learning activities. Robots can potentially be employed to proctor exams in the educational field. Ref. [14] describes a low-cost, self-contained desktop robot for proctoring exams in online/distance learning courses. The robot is connected to the student's computer through a USB port

and uses a webcam, as well as an array of sound sensors, to monitor the examination environment. The exam can be monitored in real time through the Internet by a live proctor, or the data can be recorded for later review. However, one of the drawbacks of such systems is that they are only applicable to online exams and cannot be utilized in paper exams or face-to-face classrooms.

To address the needs and limitations of other systems, we propose an autonomous AI and computer vision-based teaching assistant for face-to-face proctoring, contactless paper scanning in examinations, and voice-oriented course material provision to students. The robot initially maps a classroom to navigate via waypoints while avoiding obstacles. Using four cameras that are mounted and set up on the robot at right angles to each other, video footage of the exam session is recorded. The footage is fed into a trained neural network responsible for detecting suspicious activities during the recorded session. Our robot is also capable of interacting with students during the exam. The student can ask a question, and a response is returned that best fits the student depending on the learning style preference stored in their profile and the allowed answers during the exam. Answers to posed questions can be displayed as text or video on a tablet running any web browser. Once an exam session is complete, our robot uses the four cameras to scan four students' multi-page submissions simultaneously and rapidly. Students are asked to quickly flip their submission pages on camera, safely dispose of the paper, and leave the exam room.

The main contribution of our work is an end-to-end viable solution to contactless class activities during pandemics. To achieve this goal, we also propose: (a) an autonomous navigation system, (b) an accurate cheating detection pipeline, (c) a contactless computer vision-based exam paper scanning system, (d) a human–robot voice interaction system, and (e) an academic content personalization subsystem.

2. Materials and Methods

2.1. Proposed System Overview

Our proposed system consists of five integrated subsystems. In Section 2.3, we present the central subsystem, the Human–Robot Voice Interaction system, which is based on voice and synthesized text-to-speech technology. The user can talk to the robot and give commands to control the other subsystems. For example, during an exam, a robot or multiple robots are placed in the classroom as illustrated in Figure 1. The robot can then proctor the exam, in which case it will communicate with the navigation subsystem, presented in Section 2.4, which consists of a robot base, Lidar, and 3D sensor. This subsystem allows the robot to autonomously navigate the classroom, find the best paths, map its surroundings, and avoid obstacles. While proctoring the exam, two other subsystems are activated, course material retrieval and recommendation and cheating detection. The instructor, ahead of time, can program the robot to answer specific questions during the exam. The student can then talk to the robot and ask for help. Our robot then uses a camera to detect faces, identify the user and then provide personalized content based on similar queries if the information is programmed to be revealed during the exam, as explained in Section 2.5. The cheating detection subsystem, presented in Section 2.6, is also activated. It consists of 4 cameras mounted on top of the robot, recording a 360-degree view around it. The robot uses the navigation subsystem to navigate the classroom without the need of a teacher, thus maintaining social distancing between students.

Once the exam is over, our proposed Image-Based Exam Paper Scanning subsystem, described in Section 2.7, scans students' exam papers in a contactless way by recording a video of the student flipping through their exam papers in front of the camera. The video is then uploaded to the cloud along with the proctoring video. Uploaded videos are then processed using our computing server, using an AI Vision model to find instances of cheating and extract each exam page and student ID to be graded. Reports are then generated for the examiners to review manually.

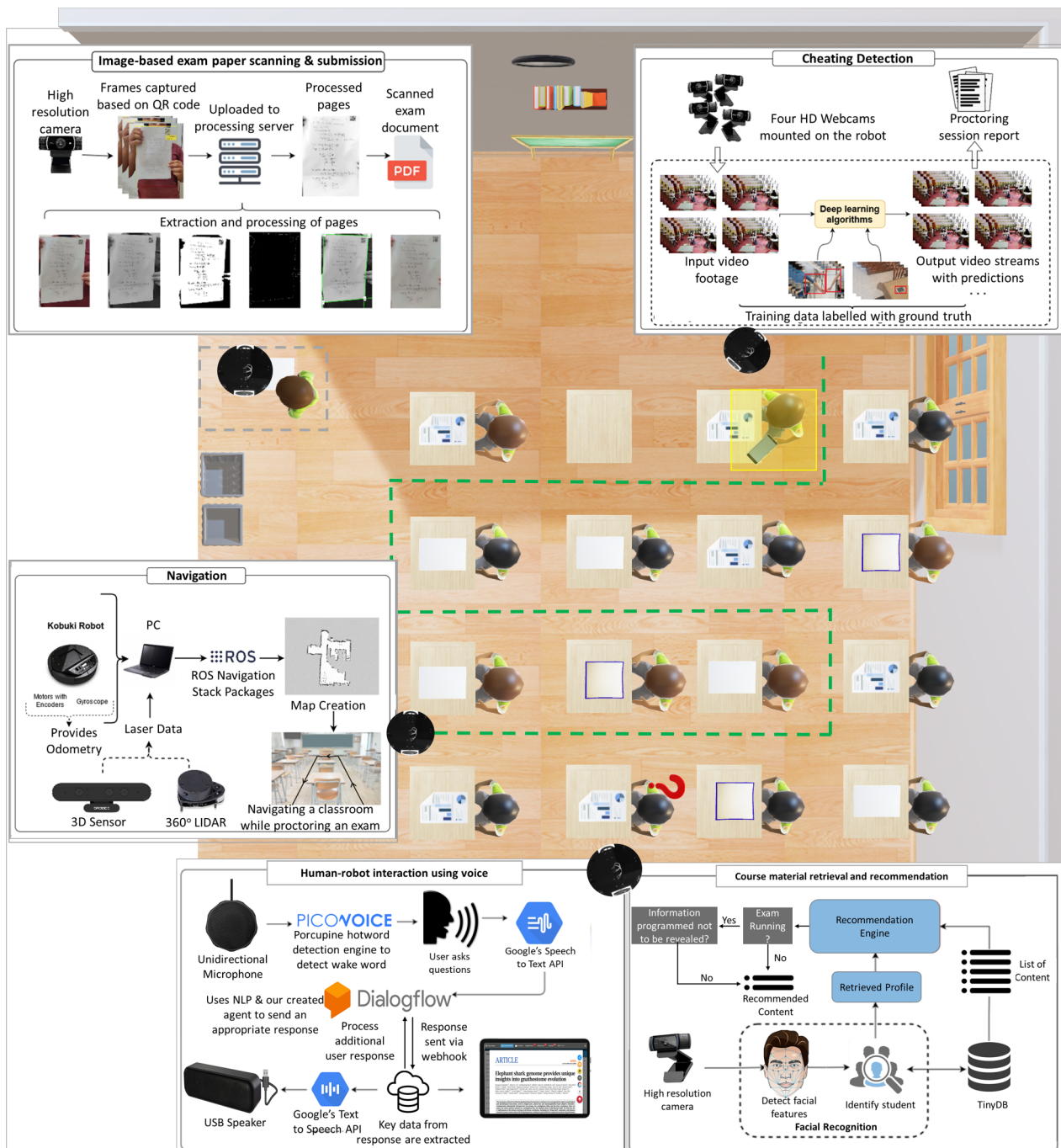


Figure 1. Overall system overview.

2.2. Robot Body

Our autonomous service robot, Reacher (Robot Teacher), consists of a Kobuki robot base, two motors with encoders, and a built-in gyroscope to provide accurate odometry (ROS Components, Valencia, Spain). The robot is equipped with a tablet (ASUS, Taipei City, Taiwan), battery (YOWOO Power, China), 3D sensor (Orbbec, Troy, MI, USA), 360 planar lidar sensor (SLAMTEC, Shanghai, China), and a laptop supported by two tall poles (HP, Palo Alto, CA, USA). It also has a bumper sensor located in the front for obstacle avoidance. The robot design and sensor positions are shown in Figure 2.



Figure 2. Robot design.

2.3. Human–Robot Voice Interaction

We use the porcupine wake word engine [15] to create the wake word “Hey Reacher”. Porcupine is an engine that is both accurate and lightweight. It allows developers to create voice-activated apps that are always listening while maintaining low power consumption. Since our robot is battery operated, a higher CPU usage directly increases power consumption and can slow down other operations such as collision avoidance and navigation speed.

After triggering the wake word, the user is then required to ask their question to the robot, and the raw audio signal is recorded and converted to text using Google’s Speech-to-Text API (Google, CA, USA) [16]. The converted text is then transmitted to Google’s DialogflowCX (Google, CA, USA) [17] for natural language processing and capturing the intents of the phrases made by the user. We create a DialogflowCX chatbot using a collection of pages that maintain the state of the conversation as shown in Figure 3, where each route can be handled with fulfillment as illustrated in Figure 4. Responses are then uploaded to our back-end server for key parameters extraction and fed to the recommendation engine to generate the appropriate response and display it on the user interface. The communication between the back-end server and the user interface is done through WebSockets [18] for increased efficiency. The user interface, depicted in Figure 5, starts by displaying a welcome message and then waits to receive a command from the user. Responses are then converted to text, input to Dialogflow, converted back to audio and sent to the WebSocket. The user would visually be able to view all generated responses through a tablet, running on a browser where the audio would also be played.

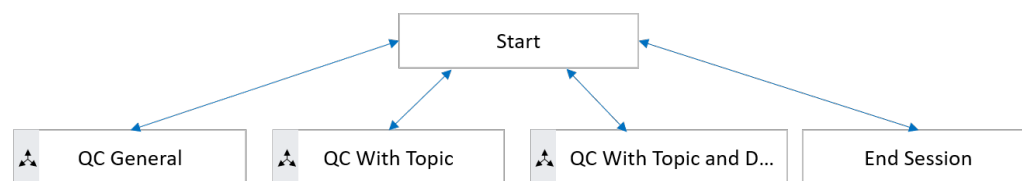


Figure 3. Start flow of the conversation.

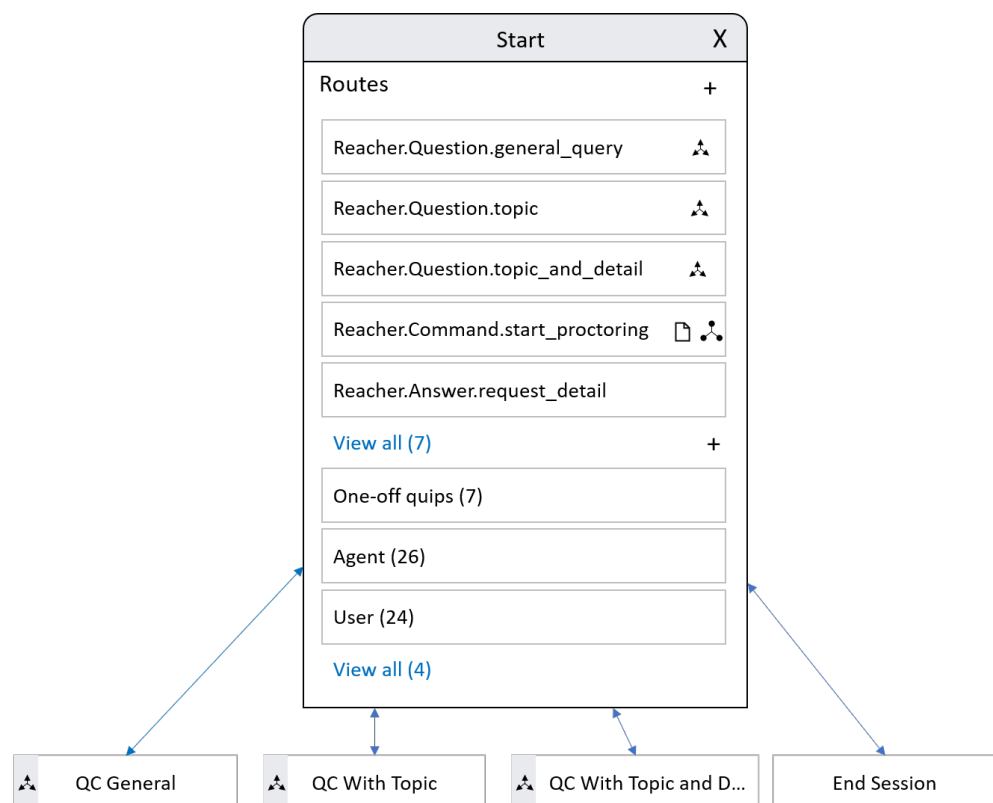


Figure 4. Start page which contains the possible routes to other pages.

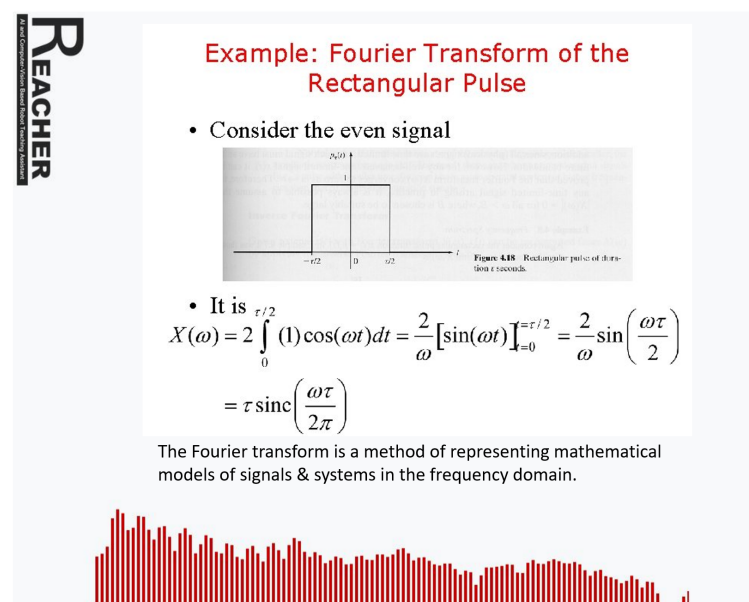


Figure 5. User interface where the upper part displays the main content retrieved, and the lower part shows the audio spectrum.

2.4. Autonomous Navigation

Our proposed robot uses the ROS [19] framework to navigate autonomously in the environment. The Kobuki robot base and the 360 LIDAR sensor provide the odometry and 2D laser data, respectively, fed into a PC running ROS. The 3D sensor mounted on the robot provides additional height information, and laser scan data and is combined with the LIDAR data to provide a more accurate map. The height information detected by the 3D sensor is used to detect the legs of the tables and chairs typically found in a

classroom. The odometry data, sensor streams for the lidar and 3D sensor, and a static map of the environment are then input into the navigation algorithm, which outputs the corresponding velocity commands to the motor driver. As shown in Figure 2, the static transformations of the lidar and 3D sensor are located at positions (0 m, 0 m, 0.3 m) and (0.03 m, 0.22 m, 0.4 m), respectively. They are then broadcasted to the navigation stack.

To generate a static map of the environment, we manually drive the robot around using a joystick. We then use the Gmapping [20] SLAM algorithm from the ROS Navigation stack to generate a map. The accuracy of the map is highly dependent on the accuracy of the localization. For localization, we use the visual odometry data from the lidar sensor, the odometry from the motor encoders and gyroscope. Although the position and pose of the robot are primarily calculated from the odometry data, this form of dead reckoning has a small error that gets amplified over time. Therefore, we use the AdaptiveMonte Carlo localization [21] approach to compensate for such errors by using the lidar's laser scan data and using particle filters to track a robot's pose against a known map. Figure 6 shows multiple red arrows where each represents a particle filter that corresponds to an estimated position of the robot. Initially, these arrows are spread out around the map. Still, as the robot is driven around, we compare the environment, and these particles' variation becomes smaller, thus improving the estimation over time.

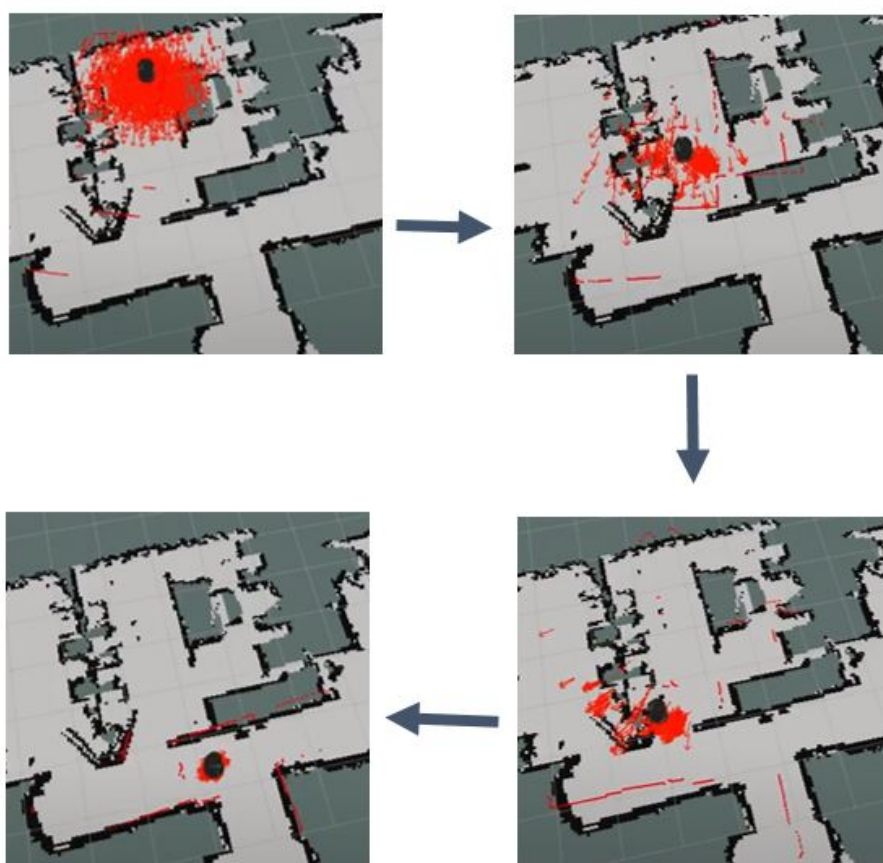


Figure 6. Particle filters are initially spread out but become more concentrated as more data is gathered from the laser scans while constantly being compared to the map, leading to better pose estimation.

Once the goal points are set for the robot, we configure the global planner to use Dijkstra's algorithm [22] to compute the path to the goal pose. The local planner, on the other hand, uses the laser scan data from the 3D sensor. As opposed to creating the map and during localization where the 360° lidar sensor is used, the local planner uses the 3D sensor to compute the local cost map as the 3D sensor provides more height information

so that the legs of the tables and chair of the classroom are more easily detected. Like the global planner, the local cost map inflates the objects of the laser scan sensor readings and is constructed live during the navigation itself. We utilize the implementation of the Trajectory Rollout and Dynamic Window algorithm [23]. This algorithm checks for objects by sampling and simulating multiple paths/trajectories and choosing the one with the minimum cost within a fixed window size that would avoid these objects and be closest to the global path as much as possible.

During navigation, the robot can face several obstacles, static or dynamic, which may affect its movement. The robot would execute a conservative reset during the first recovery behavior, whereby obstacles specified by a user's defined region (the window size) are cleared from the local cost map. If the obstacles were not cleared and no path can be found, the robot rotates in place and checks whether previous obstacles have been cleared. If the robot is still blocked, an aggressive reset of the entire cost map is mapped by removing all obstacles. If it fails to find a path, it does one last in-place rotation after clearing the whole cost map. If all of these recovery actions fail, the robot will abort trying to reach the goal.

2.5. Content Retrieval and Recommendation

We build a database of various topics from multiple courses. Each topic consists of three levels of complexities and definitions, details, equations, examples, images, videos, and text to be presented to the user based on the question. We then retrieve the content based on the parameters received from Dialogflow and the recommendation engine. The robot is programmed to open a camera and scan for new faces once the user begins interacting. We detect the face, extract the features, and compare them to the stored features in our database using our face detection algorithm. We then calculate the distance between the faces and return the user's ID with the lowest distance from the detected face if it is below a certain threshold. In case the distances are all below the threshold, the detected face will be stored as a new face with a random ID along with the extracted features. Figure 7 illustrates the process of the facial recognition algorithm.

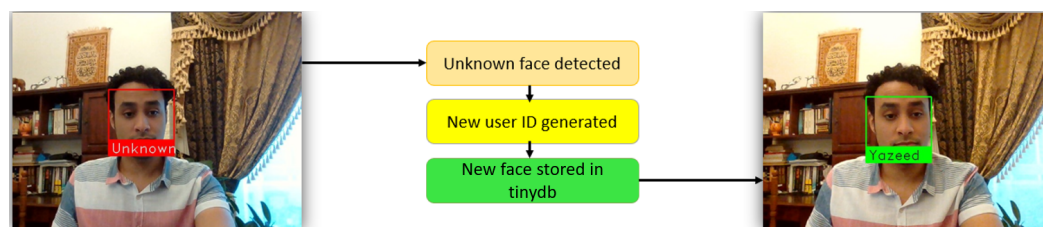


Figure 7. Face recognition algorithm flow.

For personalizing the content based on the user's level, we pass the user's ID to our recommendation model. If the user has an existing profile, the model uses their previous choice, if any, or recommends the complexity level based on other user and topic data to predict this user's preferred level of complexity for that topic. However, a general recommendation is made if no user profile exists, and the model gets retrained and updated. The system can also be used during exams while the robot is proctoring the exam. The instructor, ahead of time, can set the questions allowed during the exam. The robot will then reply to the student by saying, "I can't tell you that during the test. You should know this" if asked about something it is not programmed to reveal.

2.6. Cheating Detection

Our robot's design consists of a system responsible for flagging suspicious activities performed by students during an exam session while the robot is navigating the classroom. Four wide-angle cameras are mounted on the robot to provide a 360-degree view of the environment. We train a spatial-temporal network on 13 different cheating events to detect instances of cheating and classify them accordingly. Detected events include gazing, head movement, talking, using the phone, and the presence of unauthorized material as depicted

in Figure 8. Our AI model consists of a Convolutional Neural Network (CNN) and multiple Long Short-Term Memory (LSTM) networks. We first convert the input videos to feature vectors using our Convolutional Neural Network, which consists of 22 layers. Using the BI-LSTM architecture, the model takes the sequences and processes them through several layers to classify the sequences of vectors. To reduce overfitting, we add a 50% dropout layer and shuffle the data after every epoch.

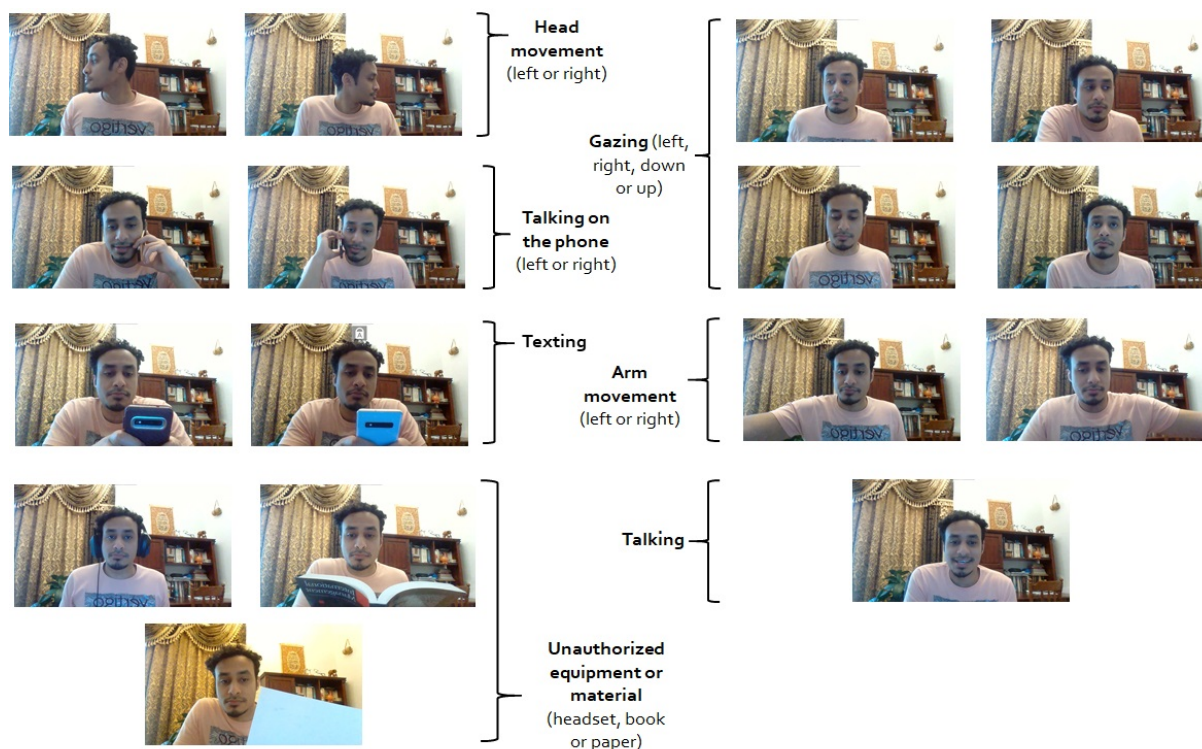


Figure 8. Cheating events detected by the system.

2.7. Image-Based Exam Paper Scanning

After the exam is done, the user asks the robot to start scanning the exam papers. The robot turns on the front-facing camera and records a video for 10 s. Meanwhile, the user holds up the exam papers and flips through them quickly. Each paper should be visible to the camera for around half a second. Exam papers should have an ID box and the page number encoded in a QR code. Once the video is recorded, it is uploaded to our remote server for storage and processing. We then sequentially process each frame of the video using our QR detection algorithm. We apply multiple filters on the image; median filter, canny edge detector, and then morphological operations to blur out the text and search for the QR code as shown in Figure 9. Once the QR code is detected and localized, we crop the image to capture the overall ROI. We use the size of an A4 paper and the size and location of the QR code to calculate the page pixel height using Equation (1), which defines the ROI that includes the page as depicted in Figure 10.

$$\frac{QR_{PixelHeight}}{QR_{ActualHeight}} * Page_{ActualHeight} = Page_{PixelHeight} \quad (1)$$



Figure 9. QR detection algorithm pipeline.

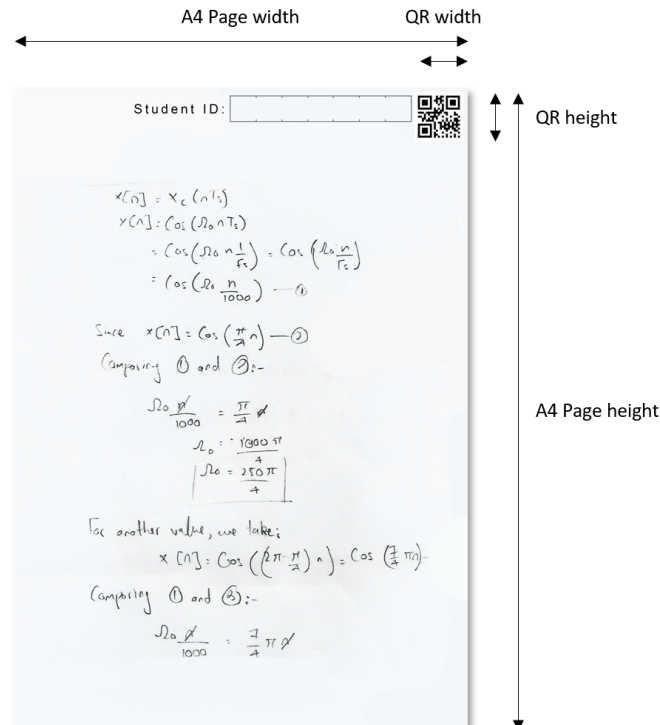


Figure 10. Simple crop calculation.

We then apply the median filter and the Hough transform on the frame, which now only contains the ROI to detect the lines in the image. Lines that are not approximately horizontal or vertical are filtered out. The rest of the lines are sorted into horizontal and vertical lines. Frames with less than two vertical and two horizontal lines detected are

discarded. We find the points of intersection between the remaining lines to detect the rectangle representing the ID location. Using the bounded lines and the aspect ratio of the A4 paper, the perspective fix is applied to the frame. The process is illustrated in Figure 11.

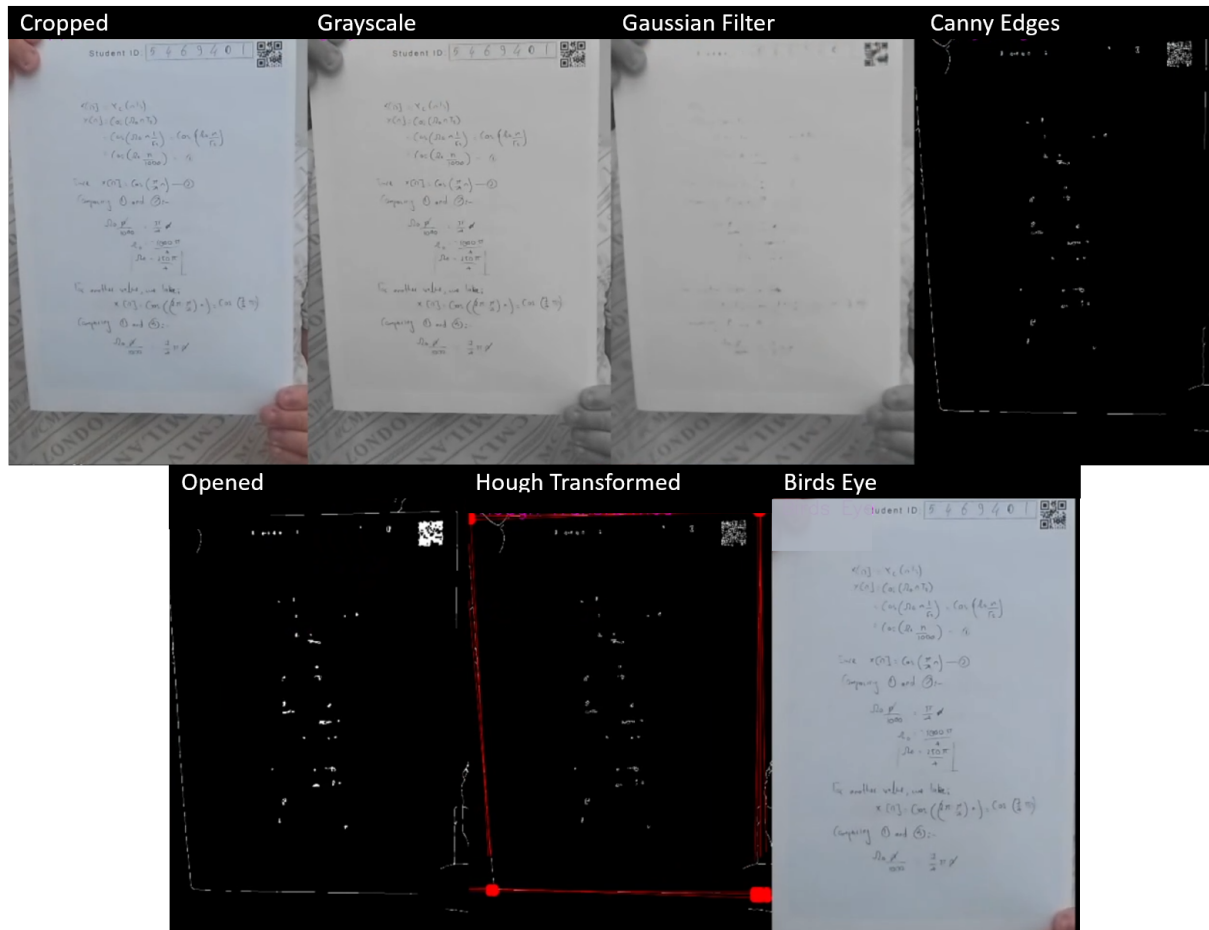


Figure 11. ID detection algorithm pipeline.

We then calculate the sharpness of the image by taking the variance of the Laplacian and extract the ID box. The frame, the page number decoded, and the sharpness are then stored along with the ID. Figure 12 describes the full-frame processing pipeline applied to each frame.

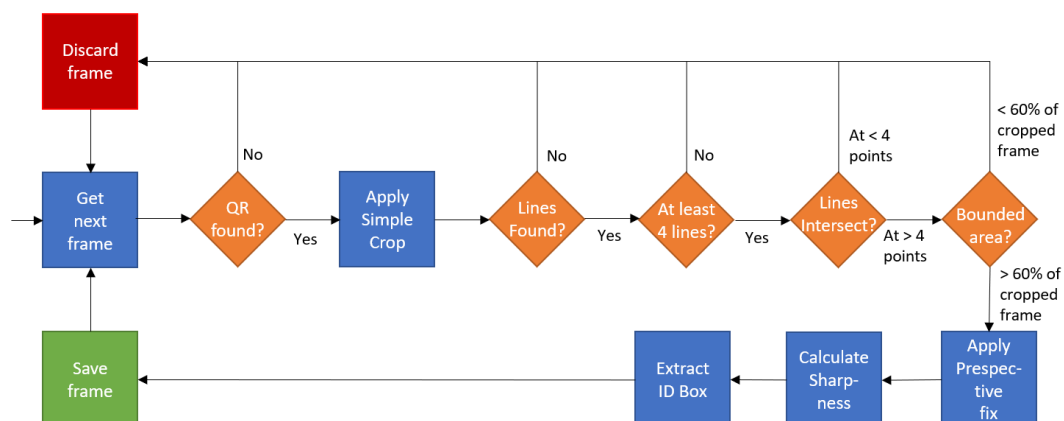


Figure 12. Frame processing pipeline.

Once all documents' frames are scanned, they are binarized, and all features except text are masked. We use an AI model trained using Transfer learning on Resnet [23,24] on the MNIST dataset [22] to detect the handwritten ID of the student. The system then writes the selected frames for each page into a PDF, labels the PDF with the detected student ID, and sends it to the involved parties. The video file is also moved for archiving.

3. Discussion

3.1. Human–Robot Voice Interaction Results

To test our Human–Robot Interaction subsystem, we tested the wake word detection, the intent accuracy, and the webhook response and audio clarity. In each test, we measured the detection accuracy in a different tone. Our system can detect the wake word “Hey Reacher” at least 90% of the time regardless of the user’s tone and accent. We also tested whether the right intent gets triggered based on the audio command provided by the user and concluded that at least 80% of the intents were classified correctly. Some words such as “Fourier” were misclassified. To achieve better results, more training phrases need to be provided. Our proposed robot is able to display the correct content on the user interface along with clear and proper audio. The user interface views at each stage are depicted in Figure 13.

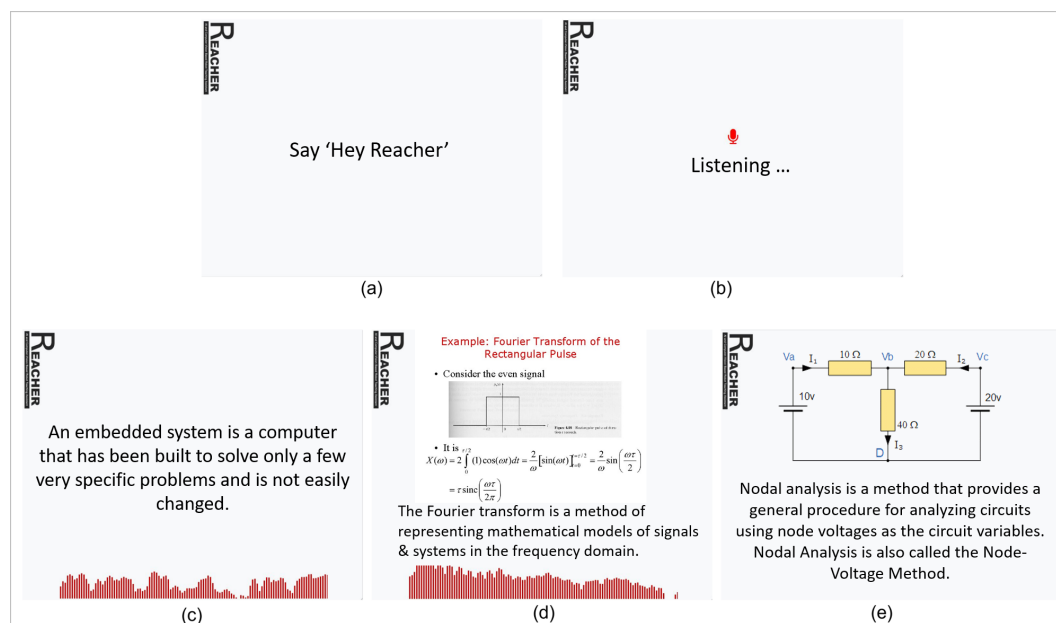


Figure 13. User interface views. (a) Shows the main view, (b) shows the view when audio is being recorded, (c–e) show the different views displayed on receiving definition, example, and images commands.

3.2. Autonomous Navigation Results

We tested our robot in several environments similar to a classroom, and it was able to map the environment and navigate through it without colliding with any obstacle. Our robot was able to map the overall shape of the room along with the static objects using the Lidar sensor as shown in Figure 14.

Since a typical classroom consists of tables, chairs, and other static and live obstacles, we tested our proposed navigation and obstacle avoidance system in such an environment. With goal points set on the planner, our robot successfully traversed through them without colliding with any obstacles. It was able to arrive at the final destination with accuracy within 50 cm² as illustrated in Figure 15. Once a new obstacle is present in the environment, the local planner generates a local path around the obstacles and to the goal point in less than 30 s. Figure 16 shows the testing results of the collision avoidance algorithm.

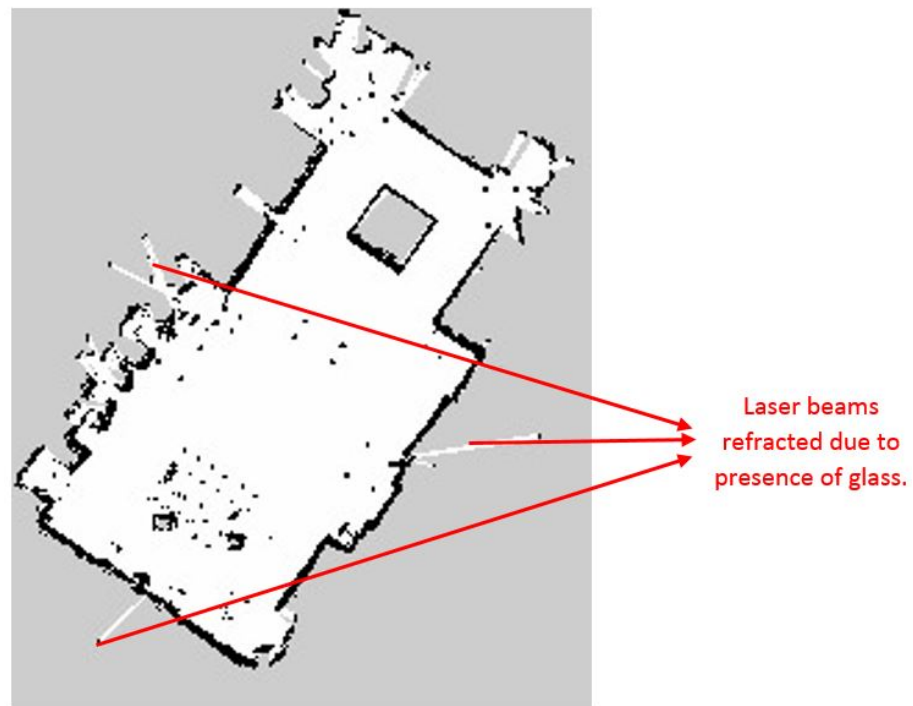


Figure 14. Map generated using the Lidar sensor.

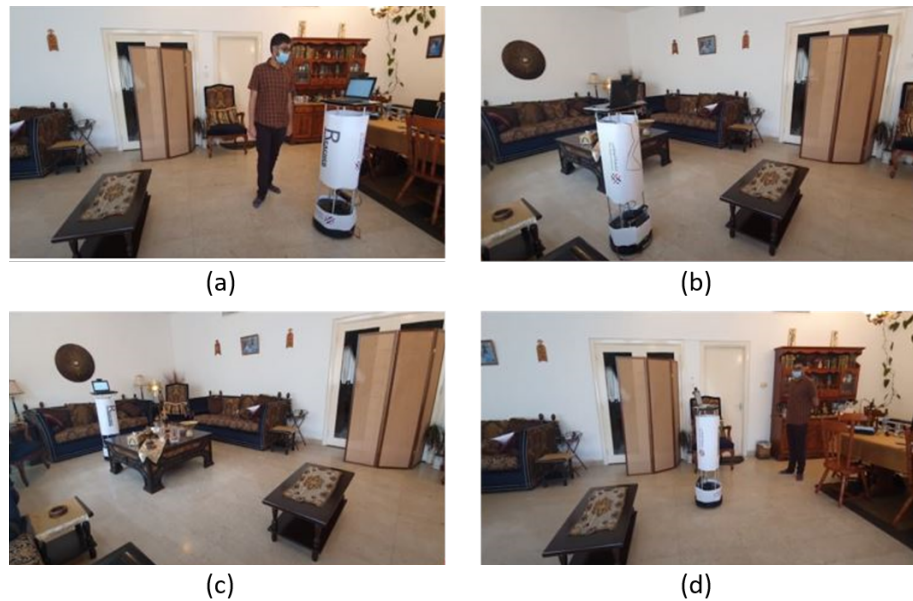


Figure 15. Robot navigating around a room with predefined goal points. (a) Shows the initial position of the robot, (b) illustrates the robot reaching the first goal point, (c) shows the robot when it arrived at the second goal point, and (d) depicts the robot after it arrived at the final goal point.

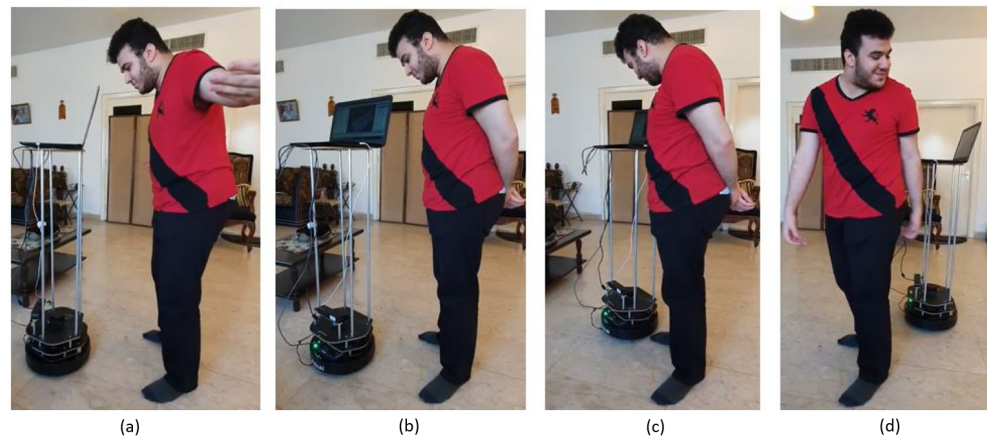


Figure 16. Collision avoidance algorithm testing results. (a) Shows the robot when detecting an obstacle by the front sensor, (b–d) show the robot avoiding the obstacle.

3.3. Content Retrieval and Recommendation Results

We tested our recommendation system for new users and users with existing profiles. The system generated a new ID for the user and made a general recommendation for the complexity level based on the most common level of complexity preferred by existing users. On the other hand, in cases where a profile of the user exists, the system was able to successfully recommend a complexity of the appropriate level based on the user's previous complexity ratings of other topics. An example of the actual content retrieved for this user is depicted in Figure 17, where the content is displayed on the screen.

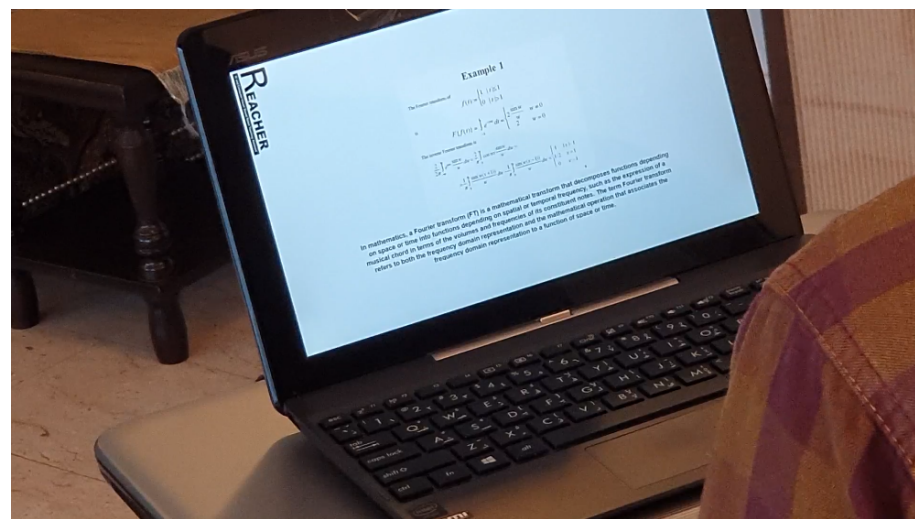


Figure 17. Content retrieval result for an existing user.

3.4. Cheating Detection Results

To validate our cheating detection algorithm, we trained our proposed network on 70% of the data, and 10% of the collected data was used for validation. Figure 18 below shows the training process of the LSTM network. The validation accuracy reached 93.1%, and the validation loss was approaching zero after a total of 30 epochs.

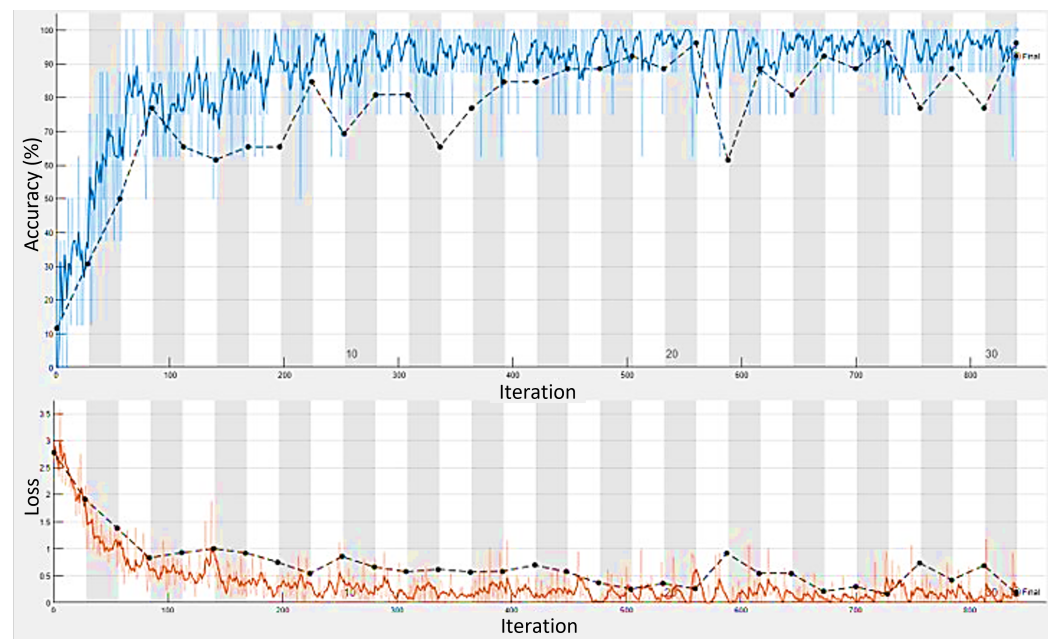


Figure 18. Proposed network with a mini-batch size of 8 and an initial learning rate of 0.0001.

The algorithm was tested under different conditions. Testing accuracy of 86.84% is obtained when the subject and background of the environment remain the same as the one trained on. However, when tested on a random person from the dataset with a different environment, the testing accuracy obtained was 53.33%, as depicted in Figure 19.

Clip	Ground_truth_label	Predicted_event	Predicted_correctly
1	ArmMovement_Left	ArmMovement_Left	True
2	ArmMovement_Left	ArmMovement_Left	True
3	ArmMovement_Right	Gaze_Down	False
4	ArmMovement_Right	Gaze_Down	False
5	Gaze_Above	ArmMovement_Right	False
6	Gaze_Above	Gaze_Down	False
7	Gaze_Down	Gaze_Down	True
8	Gaze_Down	Gaze_Down	True
9	Gaze_Left	Gaze_Down	False
10	Gaze_Left	Gaze_Left	True
11	Gaze_Right	Gaze_Right	True
12	Gaze_Right	Gaze_Down	False
13	HeadMovement_Left	Texting	False
14	HeadMovement_Left	HeadMovement_Left	True
15	HeadMovement_Right	HeadMovement_Right	True

Testing Accuracy is = 53.33%

Figure 19. Cheating detection algorithm results in a different environment.

3.5. Image-Based Exam Paper Scanning Results

We test our Image-Based Exam Paper Scanning system on multiple exams and conclude it can detect and handle multiple files. The system was able to convert the various documents present in the videos to multiple PDF files. Each page from each file was successfully extracted and is clear, readable, and properly cropped. A sample output of these results can be seen in Figure 20. The system achieved a success rate of above 80%, which could be improved by fine-tuning. Regarding the system's ability to detect handwritten digits, the system was able to correctly extract the student ID box from the page, with minor mistakes as illustrated in Figure 21.

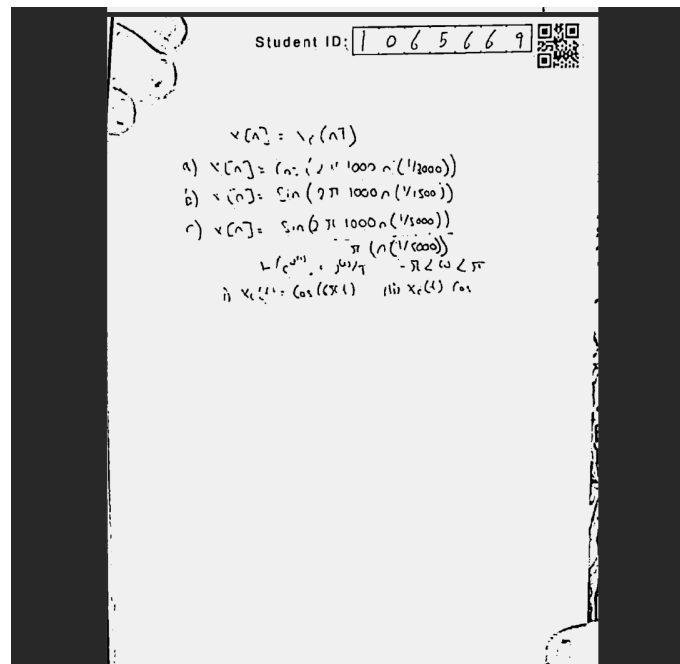


Figure 20. A sample of the pages scanned by the image scanning system.

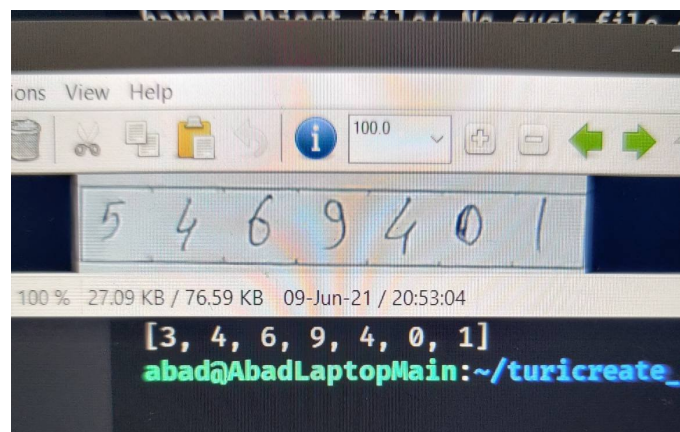


Figure 21. Digit classification result.

3.6. End-to-End Integration Testing and Validation

We performed end-to-end integration testing by placing a robot in an environment similar to a classroom. Our robot navigated the environment and was able to traverse all goal points without colliding with any obstacles. It was able to respond to the user and display correct content with clear and proper audio. Multiple cheating events were detected and classified with an accuracy of 86.85%. After the exam was done, the image-based exam paper scanning subsystem was able to handle multiple files and scan and

process them. Proctoring videos, generated reports, and scanned exam papers were then sent to the examiner.

4. Analysis and Discussion

A year and a half after schools and universities moved to distance learning due to COVID-19, students returned for in-person classes in universities and schools with certain limitations and regulatory rules in effect. Locally, the regulated return to face-to-face instructions contributed to reducing new cases from over 4000 to below 100.

We design our service robot to meet schools' needs for frequently administering proctored paper-based exams without a major investment in computers or requesting that students bring their laptops. To understand the constraints driving our design, we present the regulatory rules in place. Students and teachers are to observe strict social distancing of 2 m in all directions at all times. Moreover, there cannot be any collection of exam papers. Disinfection is to be carried out between exams. While some schools moved their examinations to their well-equipped computer labs, others do not have such capacity. The number of exams and students far exceeds the number of available computers. Typically, the teacher, while proctoring the exam, moves around the classroom to better spot cheating. However, it becomes harder to navigate the room when its capacity is stretched to the maximum to accommodate social distancing rules. Our robot use case becomes more apparent in light of these guidelines and limitations. Our robot autonomously navigates the room on behalf of the proctor, recording what it sees for post-exam inspection aided with AI flagging of suspicious cheating behaviors. Although cheating behavior can be subtle and complex, certain cheating behaviors are detectable. Many universities and schools have started to use AI-powered cheating detection software and services for online exams. Some schools use automated proctoring services, such as Respondus Monitor, which relies primarily on face detection to raise flags to be investigated by the teacher. Our robot is designed and trained to flag more suspicious behavior than ones based on face detection alone. Cheating detection using AI is possible and, while not standardized, has been shown to help [25,26].

In response to a shortage of computers for testing, some schools asked students to bring their laptops to take the tests. There is hardly enough access to power outlets to facilitate their use. Moreover, the increase in network traffic pressures in some school networks inevitably has students running into technical problems, leading to an increased rate of make-up exam requests citing technical difficulties. Using our robot proctors, students are not required to bring their laptops to campus to take exams. Our robot allows schools to fall back to the convenient use of paper while addressing measures put in place to keep students safe and curb the spread of COVID-19. We facilitate the fast scanning and submission of exam papers on the students' way out instead of the traditional paper collection. Students may need clarifications in the exam, and our policies dictate that teachers are to provide them. Our roving robot uses AI to respond and provide clarifications during the exam, reducing the need for the proctor to approach students and, in doing so, violate social distancing rules. While it is known that the virus can live on the surfaces of the robot for hours, students do not touch the robot, and fog-based disinfection after exams keep the robot surfaces virus-free.

The real and established need to run low-cost yet large-scale and paper-based yet contactless testing drives the integration of the different components into the proposed end-to-end robot. While our work demonstrated the potential and benefit, it also highlights the problems and paves the way for the research community to explore the issues further.

5. Conclusions

In this paper, we proposed an autonomous voice-controlled service robot capable of navigating classroom environments, interacting with students, proctoring exams, and scanning exam papers rapidly. The user starts by using the wake word "Hey Reacher" to initiate the human-robot voice interaction. The robot uses Google's Dialogflow CX and

natural language processing to respond to student questions. The robot uses SLAM to map the environment and an adaptive Monte Carlo localization and dynamic roll-out window algorithm to navigate the environment. We also proposed a cheating detection deep learning network with an accuracy of 86.84% to detect and classify cheating events during exam proctoring. When an exam is over, the robot scans multiple exam papers simultaneously and organizes them before uploading to the cloud. Instructors receive detailed reports about suspected cheating events and the scanned exam papers that are separated into multiple files. Our work demonstrates the potential of using service robots to support classroom activities especially during social distancing advisories.

Author Contributions: Funding acquisition, A.A.-T., H.H., A.E.-B. and M.G.; Investigation, M.Y., A.Y., R.S., S.M., Y.E. and M.G.; Methodology, M.Y., A.Y., R.S., S.M., Y.E. and M.G.; Project administration, A.A.-T., M.Y., H.H., A.E.-B. and M.G.; Software, M.Y., A.Y., R.S., S.M., Y.E. and M.G.; Supervision, A.A.-T., M.Y., H.H., A.E.-B. and M.G.; Validation, M.Y., A.Y., R.S., S.M., Y.E. and M.G.; Writing—original draft, M.Y., A.Y., R.S., S.M., Y.E. and M.G.; Writing—review and editing, A.A.-T., M.Y., H.H., A.E.-B. and M.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Office of Research and Sponsored Programs in Abu Dhabi University with grant number 19300581.

Acknowledgments: This work is supported by the Office of Research and Sponsored Programs of Abu Dhabi University.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Sharkey, A. Should we welcome robot teachers? *Ethics Inf. Technol.* **2016**, *18*, 283–297. [CrossRef]
2. Cooney, M.; Leister, W. Using the Engagement Profile to Design an Engaging Robotic Teaching Assistant for Students. *Robotics* **2019**, *8*, 21. [CrossRef]
3. The Eye Tribe. Available online: <https://theeyetribe.com/theeyetribe.com/about/index.html> (accessed on 8 August 2021).
4. Chuang, C.; Craig, S.; Femiani, J. Detecting probable cheating during online assessments based on time delay and head pose. *High. Educ. Res. Dev.* **2017**, *36*, 1123–1137. [CrossRef]
5. Shamqoli, M.; Khosravi, H. Border detection of document images scanned from large books. In Proceedings of the 2013 8th Iranian Conference on Machine Vision and Image Processing (MVIP), Zanjan, Iran, 10–12 September 2013; doi:10.1109/iranianmvip.2013.6779955. [CrossRef]
6. Boudraa, O.; Hidouci, W.; Michelucci, D. An improved skew angle detection and correction technique for historical scanned documents using morphological skeleton and progressive probabilistic hough transform. In Proceedings of the 2017 5th International Conference on Electrical Engineering-Boumerdes (ICEE-B), Boumerdes, Algeria, 29–31 October 2017; doi:10.1109/icee-b.2017.8192043. [CrossRef]
7. Tariq, W.; Khan, N. Click-Free, Video-Based Document Capture—Methodology and Evaluation. In Proceedings of the 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), Kyoto, Japan, 9–15 November 2017; pp. 21–26. [CrossRef]
8. Gomez-Urbe, C.; Hunt, N. The Netflix Recommender System. *Acm Trans. Manag. Inf. Syst.* **2016**, *6*, 1–19. [CrossRef]
9. Zanardi, V.; Capra, L. Social ranking. In Proceedings of the 2008 ACM Conference on Recommender Systems—RecSys '08, Lausanne, Switzerland, 23–25 October 2008; doi:10.1145/1454008.1454018. [CrossRef]
10. Fortuna, B.; Fortuna, C.; Mladenić, D. Real-Time News Recommender System. In *Machine Learning and Knowledge Discovery in Databases*; Springer: Berlin/Heidelberg, Germany, 2010; pp. 583–586. [CrossRef]
11. Bogers, T. Tag-Based Recommendation. In *Social Information Access*; Springer: Cham, Switzerland, 2018; pp. 441–479. [CrossRef]
12. Chin, K.; Wu, C.; Hong, Z. A Humanoid Robot as a Teaching Assistant for Primary Education. In Proceedings of the 2011 Fifth International Conference on Genetic and Evolutionary Computing, Kitakyushu, Japan, 29 August–1 September 2011; pp. 21–24. [CrossRef]
13. Sun, Z.; Li, Z.; Nishimori, T. Development and Assessment of Robot Teaching Assistant in Facilitating Learning. In Proceedings of the 2017 International Conference of Educational Innovation through Technology (EITT), Osaka, Japan, 7–9 December 2017; pp. 165–169. [CrossRef]
14. Rosen, W.A.; Carr, M.E. An autonomous articulating desktop robot for proctoring remote online examinations. In Proceedings of the 2013 IEEE Frontiers in Education Conference (FIE), Oklahoma City, OK, USA, 23–26 October 2013; pp. 1935–1939. [CrossRef]
15. Home | p5.js, P5js.org. 2021. Available online: <https://p5js.org/> (accessed on 8 August 2021).
16. Speech-to-Text: Automatic Speech Recognition | Google Cloud, Google Cloud. 2021. Available online: <https://cloud.google.com/speech-to-text> (accessed on 13 August 2021).

17. Dialogflow CX documentation | Google Cloud, Google Cloud. 2021. Available online: <https://cloud.google.com/dialogflow/cx/docs> (accessed on 15 August 2021).
18. Getting Started. 2021. Available Online: <https://websockets.readthedocs.io/en/stable/intro/index.html> (accessed on 4 August 2021).
19. Rozario, A. FAQ: How Was GMAT Held Using AI Proctors? Can It Detect Cheating? TheQuint. 2021. Available online: <https://www.thequint.com/news/education/what-is-an-artificial-intelligence-proctored-test-can-it-detect-cheating-faq> (accessed on 8 August 2021).
20. Base_LOCAL_PLANNER-ROS Wiki. 2021. Available Online: http://wiki.ros.org/base_local_planner (accessed on 20 August 2021).
21. Pech-Pacheco, J.L.; Cristobal, G.; Chamorro-Martinez, J.; Fernandez-Valdivia, J. Diatom autofocusing in brightfield microscopy: A comparative study. In Proceedings of the 15th International Conference on Pattern Recognition. ICPR-2000, Barcelona, Spain, 3–7 September 2000; Volume 3, pp. 314–317. [CrossRef]
22. MNIST Handwritten Digit Database, Yann LeCun, Corinna Cortes and Chris Burges. 2021. Available online: <http://yann.lecun.com/exdb/mnist> (accessed on 8 August 2021).
23. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778. [CrossRef]
24. Donahue, J.; Jia, Y.; Vinyals, O.; Hoffman, J.; Zhang, N.; Tzeng, E.; Darrell, T. DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition. *arXiv* **2013**, arXiv:1310.1531.
25. Nigam, A.; Pasricha, R.; Singh, T.; Churi, P. A Systematic Review on AI-based Proctoring Systems: Past, Present and Future. *Educ. Inf. Technol.* **2021**, *26*, 6421–6445. [CrossRef] [PubMed]
26. Holden, O.; Norris, M.; Kuhlmeier, V. Academic Integrity in Online Assessment: A Research Review. *Front. Educ.* **2021**, *6*, 258. [CrossRef]