2026 ASEE ANNUAL CONFERENCE June 21–24, 2026 | Charlotte Convention Center, NC
Host Sponsor UNC Charlotte
ASEE AMERICAN SOCIETY FOR ENGINEERING EDUCATION
Paper ID #51339

Controlling a Robot Arm with a Large Language Model (LLM) in an Educational Setting

Prof. Robert L. Avanzato, Pennsylvania State University, Abington

Robert Avanzato is an associate professor of engineering at the Penn State Abington campus where he teaches courses in electrical and computer engineering, computer science, robotics and computer vision. His research interests are robotics, computer vision and AI.

Jared Daniel, Pennsylvania State University, Abington

Jared Daniel is currently a Software Engineer at Lockheed Martin. During his time at Penn State University, where he studied software engineering, he contributed to this research paper on applying LLMs to robot arm control in an educational context. His academic experience included coursework in computer vision and deep learning. In addition to this work, he led three teams to finalist placements in Penn State's AI for Good Challenge.

WIP: Controlling a Robot Arm with a Large Language Model (LLM) in an Educational Setting

Abstract

The objective of this research project is to develop a controller for a low-cost robot arm using a multimodal Large Language Model (LLM) for educational purposes. Combining LLM visual object detection capabilities along with spatial understanding and code generation, a system was developed to control a low-cost robot arm using natural language commands (ex: “pick up the blue block”, “align the blocks in a horizontal line”). The goal is to establish an educational resource that can be used by educators to demonstrate the control of a robot arm with multimodal LLMs in the classroom. This will allow students to experiment with AI-centered engineering design.

1.0 Introduction

LLMs have had a profound impact in recent years in the areas of search, chatbots, document production and summarization, problem solving, art, multimedia generation, image analysis, coding support and beyond. More recently, commercially available, multimodal LLMs (MLLMs) support enhanced image analysis, object identification, segmentation and spatial reasoning [1, 2]. Researchers are exploring the integration of hardware, such as robotics, into an LLM solution [3]. There are existing systems which integrate LLM and robotics such as SayCan [4], Palm-E [5], RT-2 [6], Code as Policies [7], and Vima[8]. SayCan, Palm-E, RT-2 and Vima models do not generate intermediate robot code as is done in the system described in this paper. Code as Policies does not use a commercially available MLLM as is utilized in our system. Our design choice to use a commercially available, pretrained MLLM and code generation, is to support transparency, inspection and low cost for educational applications.

The benefit of integrating an MLLM into a robotics system is that the MLLM can handle the user interaction and language understanding and language ambiguity and multimodal input (voice, image, video) very effectively. The challenge then is to integrate the output of the commercially available MLLM into a robotics system, such as a robot arm. We have demonstrated a solution that includes natural language, image analysis and spatial reasoning and code generation to control a robot arm. It is expected that LLMs and generative AI will have a major impact on the way students learn engineering as well as how students design engineering systems in the future. The goal is to develop an educational resource with a hands-on component that will guide undergraduate students into understanding the role of MLLMs and specifically, how to integrate a multimodal LLM into a working system to control a robot arm. This experience will provide students with valuable knowledge in preparation for the workforce and research activities in a rapidly developing field. The technology described has been developed

with undergraduate student participation and will undergo testing with students in a robotics course.

The paper is divided into two main sections. In the first section, activities will be described to interactively analyze images of objects and perform operations such as bounding box calculations using the Gemini interface in a manual mode. The second section of the paper will describe a working system to autonomously analyze an image of objects and program a robot arm to execute a user command such as "move the blue block to the right of the yellow block."

2.0 Gemini Visual Language Model (VLM)

Google's Gemini 1.5 LLM was released in February of 2024, and Gemini 2.5 was released in March of 2025 with enhanced performance. (NOTE: Gemini 3.0 was released in November of 2025). Gemini is a multimodal LLM and is considered a VLM (vision language model) given its performance with images [9,10,11]. These Gemini models represent a milestone in LLM performance and especially in image and spatial understanding. Due to the high performance, utility, low-cost and well-documented API, we have adopted this LLM tool into our project. (Other frontier VLMs can also be used to achieve similar educational outcomes.) Using the freely available, web-based AI Google Start App called Spatial Understanding [12], it is possible to upload an image and have the system draw bounding boxes around pertinent objects and return labels and bounding box information for those objects. Here is a sample image of red and blue squares that was uploaded to the starter app and the result is shown below with bounding boxes and labels (see Figure 1). Gemini can answer basic questions about size, color, shape, location and spatial relationships. There are cases when errors occur, and it is important to provide carefully constructed prompts. (It should be noted that we used Gemini 1.5 and 2.5 API in our initial software system described in this paper.)

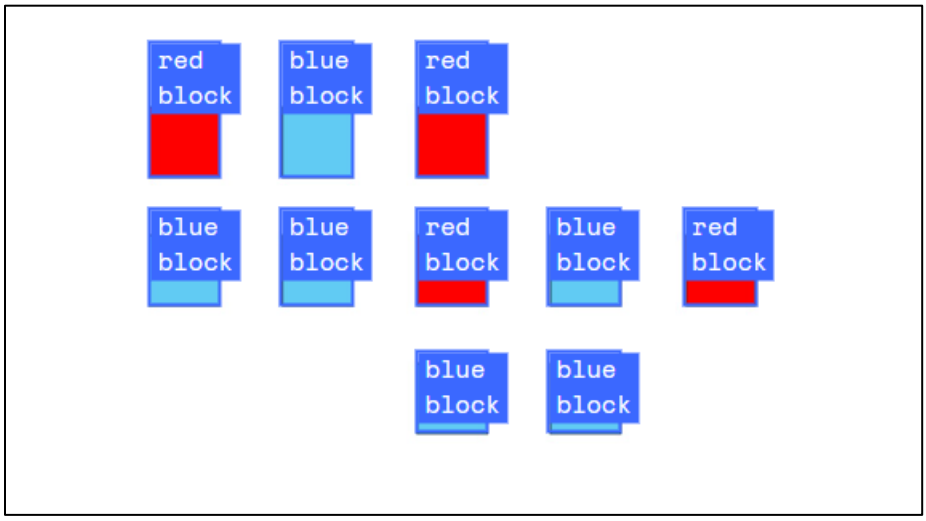

Figure 1: Colored squares with bounding boxes and labels

Here is a partial listing of the bounding box information (2 objects) generated by Gemini as a result of analyzing the above image of red and square blocks. There is bounding box data generated for each of the 10 objects and the information is stored in a data structure that will be

used as input data to future steps. The bounding box information will be critical when we integrate the Gemini LLM into the physical robot.

{"box_2d": [80, 162, 341, 240], "label": "red block"},

{"box_2d": [80, 302, 341, 382], "label": "blue block"},

Using the Starter app using the Google AI studio for Gemini, questions about the spatial relationships of the blocks in a sample image can be asked interactively in natural English and answered by Gemini. Here are some sample questions that can be answered successfully by the Gemini system in the Google AI studio using the image of colored blocks above. The significance of this exercise is to demonstrate to the students the spatial understanding capabilities of Gemini with respect to the objects in the image.

- How many red blocks are in the image? **Answer:** There are 4 red blocks.
- How many blue blocks are in the image? **Answer:** There are 6 blue blocks.
- How many horizontal rows are in the image? **Answer:** There are 3 horizontal rows.
- How many vertical columns are in the image? **Answer:** There are 5 vertical columns.
- If each block is 1 inch square and there is a 0.5-inch gap between the blocks, then what is the length of row 1? **Answer:** The length of row 1 is 4 inches.
- If rows 1 and 3 were combined into one row, how many blocks would be in row 1? **Answer**: If rows 1 and 3 were combined into one row, there would be 5 blocks in the new row 1.

In the next example Gemini was asked to provide steps to achieve the reassembly of the blocks based on the image provided.

- What are the steps to combine rows 1 and 3 into one row at location row 1

The full response from Gemini is provided in the figure below. Again, the significance here is to have students observe and verify the task planning performed autonomously by the MLLM. This information is valuable when designing prompts for future actions.

Okay, let's describe the steps to combine rows 1 and 3 into a single row at the location of the original row 1.

Initial State: Blocks are in their original positions.

Goal: Move all the blocks from row 3 into row 1, forming a single row in the space currently occupied by row 1.

Steps:

1. **Block 1:** The first block in row 1 is already in place. No move needed.
2. **Block 2:** The second block in row 1 is already in place. No move needed.
3. **Block 3:** The third block in row 1 is already in place. No move needed.
4. **Block 4:** Select the first block from row 3. Move it horizontally to the right of the last block in the current row 1, leaving a gap that matches the original gap between the blocks.
5. **Block 5:** Select the second block from row 3. Move it horizontally to the right of the last block in the current row 1, leaving a gap that matches the original gap between the blocks.

Final Result: All the blocks from row 3 are now moved into row 1, forming a single row of 5 blocks in the space previously occupied by row 1.

Note:

- The gaps are approximate, matching the existing spacing of blocks in the original image.
- The blocks maintain their original colors.

Figure 2: Steps to rearrange blocks

The step-by-step solutions (subtasks) from the Gemini LLM combined with the availability of bounding boxes for each object, enables Gemini processing to construct Python code that can direct a robot arm to achieve the sequence of steps to achieve the goal. All of this functionality is accessible with the Gemini API. The Spatial Reasoning App, on the other hand, is a useful, interactive, web-based tool to gain insight into the capabilities and experiment with prompt creation.

3.0 VLM Agentic Interface to Dobot Robot Arm

A software system interfacing the Gemini MLLM, a webcam (Logitech C922), and a low-cost, 4 DOF robot (Dobot Magician) was designed and tested. An image of the experimental setup is depicted below in Figure 3. The camera is positioned 12 to 14 inches above the surface. The software was running in Python on a standard Windows PC. The workspace for the robot was established at 5 inches by 8 inches, as indicated by the white rectangle. Although Gemini reports some spatial interpretation capabilities in 3D, all operations were limited to the 2D plane for our development and testing. The robot arm has a suction cup end-effector and a vacuum pump (not shown in image).

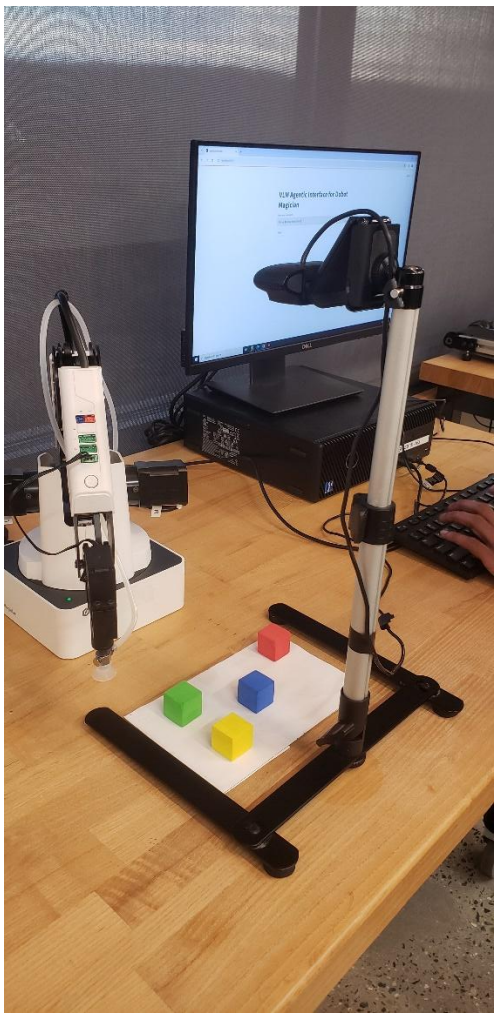

Figure 3: Experimental Setup

Below is a high-level description of the 4 major agents (modules) which operate on the images captured by the webcam. For each agent there is an LLM prompt or set of prompts to retrieve information from the LLM based on the image. The text prompts, which are embedded in the Python script and can be modified, are sent to Gemini using the API. The Python code then processes the result of the prompt and moves on to the next agent.

Bounding Box Agent:

- Extracts workspace and block positions in a specific format.
- Parses the bounding box results into a structured format.
- Key LLM prompts:

        - "Return bounding boxes for just the flat, white, rectangular paper in picture in the following format as: [ymin, xmin, ymax, xmax]"
        - "Return bounding boxes for all the blocks in the following format as a list [ymin, xmin, ymax, xmax] (block_color)"

- **Spatial Analysis Agent:**
    - Analyzes spatial relationships between blocks based on the scene and user command.
    - Extracts details like object size, orientation, and spatial relationships.
    - Key LLM prompt:
        - "Given the user's command for a robot arm: {user_command} Describe the spatial relationship between the all the relevant objects in the scene captured in the image. I do not need the location/bounding boxes. I need just the approximate size (mm), shape(mm), orientation and spatial relationship of each of the relevant objects. Include anything else that would be useful to know if I will be coming up with a plan to perform the user's command."

- **Action Plan Agent**:
    - Performs coordinate system transformation between image coordinates and robot system coordinates.
    - Generates a step-by-step plan for the Dobot Magician robot arm to execute the user's command.
    - Key LLM prompts
        - "Based on the scene provided below, generate a list of detailed and specific steps to perform this action using a Dobot Magician robot arm: {user command included here}. Here are the location/bounding boxes of the objects (use these actual values to determine locations, etc.; don't use example locations. {bounding box data from step 4 supplied here}. Here is the spatial analysis of the objects: {data from spatial analysis agent included here}"
- **Code Generation Agent:**
    - Produces executable Python code tailored for the Dobot Magician.

- Sample Dobot code is provided to the LLM such as: move arm from position A to B, turn on/off vacuum pump, etc.
- Saves and runs the script using a subprocess
- Key LLM Prompts:
  - "Here is a list of steps I want to perform on a Dobot Magician robot arm: (output of action plan agent included here). Now write a Python program to execute these steps. Only return Python code, and no other text (use comments for other info). I've also attached some example code.''', {example robot code included here}"

It was determined empirically that sequencing the steps in this manner yielded better results than an approach which would use a single, larger prompt. We also noticed improvements overall in performance moving from Gemini 1.5 to 2.5 and we expect more performance gains with Gemini 3.0.

Interested readers may request sample code and instructions on integration into a classroom activity from the authors.

4.0 Results

Listed below are the intermediate results of the software interface responding to the command "pick up the blue object". The processed image with bounding boxes displayed is shown in Figure 4. The output of the action plan agent is shown below. Notice this action plan agent output, generated by the Gemini LLM, includes robot arm motion and operation of the vacuum gripper.

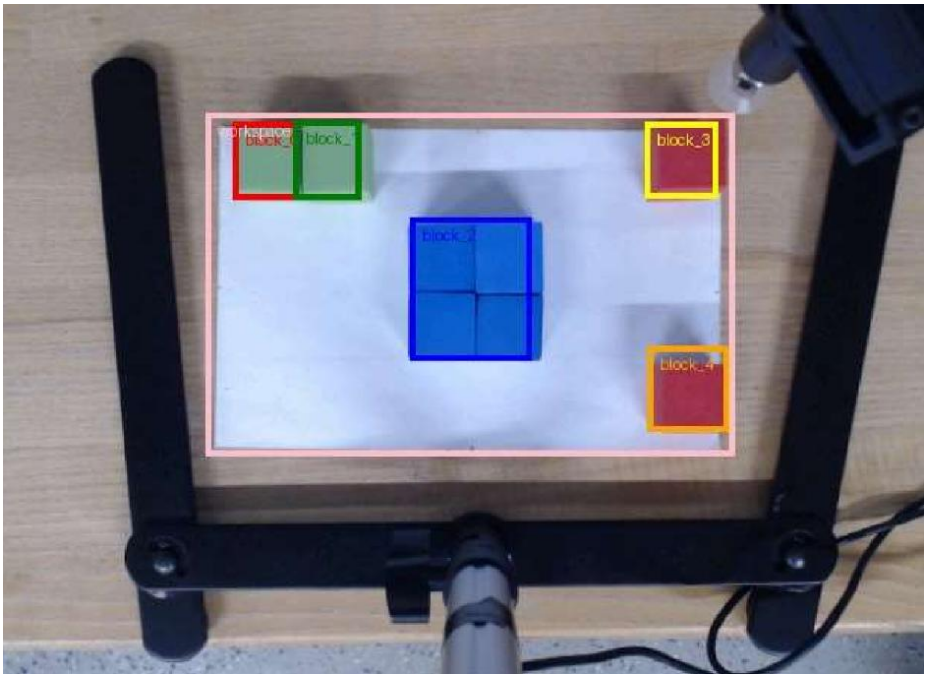

Figure 4: Input Image for Demonstration

```
**Step-by-Step Procedure for Picking up the Blue Block with Dobot Magician using Suc

1. **Move to Pre-Pick Position:**
    * **Command:** `moveTo(250, 0, -30, 0)`  (Move above the blue block, leaving som
    * **Rationale:** This positions the suction cup directly above the block at a sa

2. **Descend to Pick-up Height:**
    * **Command:** `moveTo(250, 0, -50, 0)` (Descend to the surface height of the bl
    * **Rationale:**  This lowers the arm to engage the suction cup with the block.

3. **Activate Suction:**
    * **Command:** `activateSuction()`  (This command assumes the Dobot Magician API
    * **Rationale:** This engages the suction cup to firmly grip the blue block.

4. **Ascend with Block:**
    * **Command:** `moveTo(250, 0, -30, 0)` (Raise the arm slightly to clear the sur
    * **Rationale:** This lifts the blue block clear of the surface, ensuring it's n

5. **Move to Safe Drop-off Position (Optional):**
    * **Command:** `moveTo(280, 50, -30, 0)` (Move to a designated position to hold
    * **Rationale:**  Moving the block to a secondary position offers a safe place b


 6. **Deactivate Suction:**
     * **Command:** `deactivateSuction()`
     * **Rationale:**  This releases the suction cup from the block. This should be d

 7. **Return to Home Position (Optional):**
     * **Command:** `moveTo(250,0,0,0)` (Return the arm to a home position.)
```

The coding agent was then able to automatically generate code for the Dobot based on the action steps and then execute the code to successfully pick up the blue block. All agent output for this demo is available from the authors. In the event a user needed to adapt this LLM system architecture to another robot arm platform, it would only be necessary to provide sample code of basic operations from the robot documentation. No application-specific or custom robot arm code development is required.

5.0 Conclusion and Future Directions

It is hoped that this LLM controlled robot arm prototype can be used as an educational resource to introduce engineering and computer science students to “LLM-centric” or “AI-centric” design strategies. The Gemini multimodal LLM provided: 1) a natural language interface, 2) object detection and spatial reasoning, 3) step-by-step planning (action plan), and 4) automated code generation for the robot arm. This allowed a system to be able to control a low-cost robot arm

(Dobot Magician) to perform specific tasks. We have limited our scope currently to manipulating blocks in a 2D environment for simplicity.

One of the limitations of our project is the relatively small robot workspace which limits the sizes of objects to be manipulated and the distances for objects to be moved. Also, there were occasions when the calibration step of the robot arm failed and the calibration step had to be repeated. The system worked successfully in a wide range of conventional indoor lighting and no special lighting was required.

The learning outcomes of this technology include: 1) LLM and VLM interfacing. Students may use alternate MLLMs and compare performance; 2) prompt engineering. Students will be encouraged to analyze and modify prompts to improve capabilities; 3) code generation. The code generation is trained via in-context learning from the robot sample code which can be modified as needed; 4) robot arm technology. Students learn to set up, operate and integrate a robot arm with a vision system, and 5) API programming.

Our innovation is that we have developed a working prototype and developed educational resources to implement a commercially available MLLM control of a robot arm with vision using low-cost equipment that is suitable for use in an undergraduate engineering program. All resources (including materials, code, setup, demos and instructor guide) will be available to readers upon request. Utilizing the power of a pre-trained, multimodal LLM with vision capabilities will potentially allow students and researchers to design and implement advanced robotics applications that would otherwise be out of reach or require advanced coursework. This technology can impact areas such as intelligent manufacturing, 3D construction, assembly/disassembly, art, circuit fabrication, home robotics, healthcare, etc.

This is a work-in-progress, and while we have successfully implemented a select number of test cases with our working prototype, we hope to expand the demonstrations and capabilities and share these results online. One near future effort is to integrate and evaluate Gemini 3.0. Preliminary results with Gemini 3.0 look promising. We also plan to interface other robot arm platforms in the future, incorporate voice input and experiment with 3D manipulation of blocks. This resource described in this paper is planned to be integrated into a robotics course at our institution in the spring and fall of 2026 for further evaluation and assessment. The laboratory is equipped with 7 stations, with each station comprising a robot arm, webcam, camera stand and workstation PC with internet access. At that time, we will collect additional data on performance, student feedback, usage and educational outcomes.

6.0 References

[1] Zhang, Jingyi, et al., Vision-Language Models for Vision Tasks: A Survey; IEEE transactions on pattern analysis and machine intelligence, 08/2024, Volume 46, Issue 8.

[2] Lu, J., Batra, D., Parikh, D., & Lee, S. (2019). A survey of vision-language models. arXiv preprint arXiv:1912.06863.

[3] Jeong, H., Lee, H., Kim, C., & Shin, S. (2024). A Survey of Robot Intelligence with Large Language Models. *Applied Sciences*, *14*(19), 8868. https://doi.org/10.3390/app14198868

[4] Brian Ichter, et al., "Do As I Can, Not As I Say: Grounding Language in Robotic Affordances," in Proceedings of the 6th Conference on Robot Learning, PMLR, vol. 205, pp. 287–318, Dec. 2023.

[5] Danny Driess, et al., "PaLM-E: An Embodied Multimodal Language Model," in *Proceedings of the International Conference on Machine Learning (ICML)*, 2023.

[6] Brianna Zitkovich, et al., "RT-2: Vision-Language-Action Models Transfer Web Knowledge to Robotic Control," in *Proceedings of the 7th Conference on Robot Learning*, PMLR, vol. 229, pp. 2165–2183, Nov. 2023.

[7] J. Liang, et al,, "Code as Policies: Language Model Programs for Embodied Control," in *Proceedings of the 2023 IEEE International Conference on Robotics and Automation (ICRA)*, London, UK, May–June 2023, pp. 9493–9500.

[8] Y. Jiang, A. Gupta, Z. Zhang, G. Wang, Y. Dou, Y. Chen, L. Fei-Fei, A. Anandkumar, Y. Zhu, and L. Fan, "VIMA: General Robot Manipulation with Multimodal Prompts," *arXiv preprint arXiv:2210.03094*, 2022.

[9] Google. (2023). Introducing Gemini: The Next Chapter in Google AI. https://gemini.google.com/

[10] Vision with Gemini API: https://ai.google.dev/gemini-api/docs/vision?lang=python

[11]Gemini Spatial Example: https://gemini-spatial-example.grantcuster.com/

[12] Gemini Spatial Understanding Starter App: https://aistudio.google.com/app/starter-apps/spatial