Incorporating AI in Construction to Automate Excavator Operations

Arai Leyva, *Citrus College*
Rakesh Mahto, Electrical & Computer Engineer, California State University, Fullerton

Background

- Artificial Intelligence has been increasingly integrated into the construction industry, with the goal of enhancing productivity and safety.
- Despite the advancements made, the practical use of AI in autonomous excavator operations is not extensively explored, mainly because of the complexity involved in their tasks.
- Conventional excavators often rely on manual controls and lack interactive features, which may limit their efficiency in modern construction tasks.
- To explore the possibilities of implementing advancements in automation into real excavators and overcome existing limitations, we adopted the approach of modifying a toy excavator.
- By using a single-board computer, sensors, and computer vision to modify the toy excavator, we want to enhance its functionality and provide a foundation for future improvements on excavators.

Figure 1: Scaled 1:14 Remote Controlled Toy Excavator

Figure 2: Raspberry Pi 4 attached with RP LiDAR

Methods

Global Materials:

- Remote Controlled Toy Excavator 1:14 Scale
- Raspberry Pi 4 4GB with 16GB microSD Card
- Raspberry Pi Camera Module with Case
- Slamtec RP LiDAR A1M8 2D 360˚ 12 Meter Scanning Radius
- Electrical Components (encoders, potentiometers, wiring cables)

Toy Excavator:

- Disassemble remote control and identify the connections associated with the manual joysticks and buttons.
- Next, desolder the existing manual joysticks and replace them with digital potentiometers, while simultaneously swapping out the manual buttons with transistors to enable control through the Raspberry Pi.

Raspberry Pi 4:

- Use the Raspberry Pi Imager to download and flash Raspberry Pi OS Full version onto the microSD card.
- Once the microSD card is inserted into the Pi's card slot, connect peripherals to set up our Pi and update/install any applications, libraries, or packages we may need.
- Connect the digital potentiometers and transistors to the Raspberry PI, using the manufacturer's pin configuration data sheet as a reference. Utilize jumper wires to establish the required connections, including SPI communication lines(clock, data in, and data out) and the essential power supply lines.

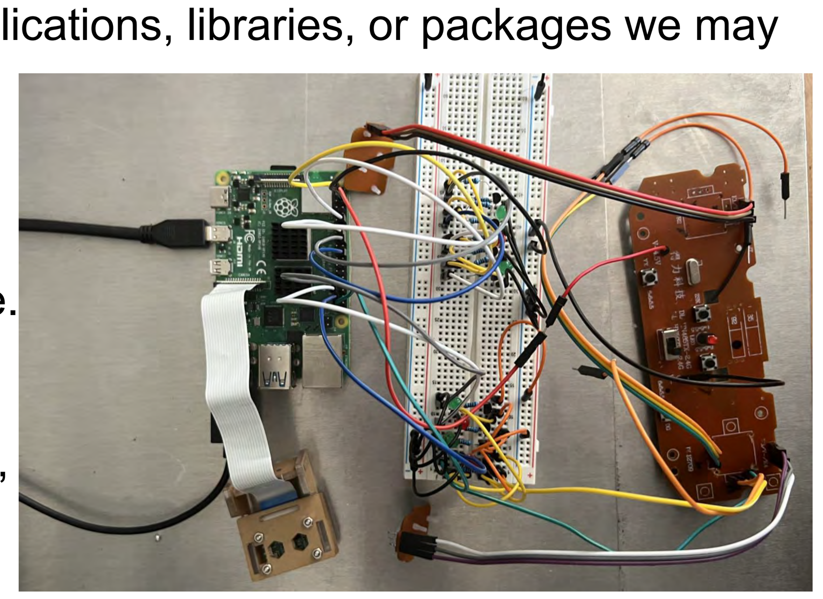

Figure 3: Disassembled remote control connected to potentiometers through breadboard into the Raspberry Pi.

Methods continued

Computer Vision:

- Utilized Google Colab for model training due to access to powerful GPUs provided by them. Obtained soil image dataset using a snippet of code from Roboflow for training a soil detection model.
- Prepared required directories and trained the YOLOv8 model using a concise one-line code with the assistance of the GPUs from Colab.
- To make the model compatible with the Raspberry Pi, it was converted to a suitable format, ensuring integration with the environment. Once we have the required dependencies and libraries (Python, OpenCV, YOLOv8) setup OpenCV on the Pi to process the live feed from the camera module.

RP LiDAR:

- Increase Raspberry Pi Swap memory to allow it to use the swap files as additional memory when RAM is exhausted.
- Download the RP LiDAR SDK on GitHub which gives access to libraries and dependencies. Along with that we download a ROS package to evaluate the data provided by the LiDAR.
- Error compiling a workspace on the SDK due to manufacturer not keeping the kit up to date with newer version of operating systems and IDE's.

```
%cd {HOME}/drive/MyDrive/datasets

!pip install roboflow --quiet

from roboflow import Roboflow
rf = Roboflow(api_key="[redacted]")
project = rf.workspace("taufans-qnmrz").project("soil-detection-2uaco")
dataset = project.version(5).download("yolov8")
```

```
%cd {HOME}/drive/MyDrive

!yolo task=detect mode=train model=yolov8s.pt data={dataset.location}/data.yaml epochs=25 imgsz=800 plots=True
```

Figure 4: In Google Colab. (Top) Snippet of code from Roboflow to extract the dataset into the notebook. (Bottom) Incisive one-line code to train the model.

Results

- Trained model using YOLOv8 and Google Collab to detect soil.
- Reconfigured the buttons on remote control to be controlled by the potentiometers proving the Raspberry Pi connection was effective.

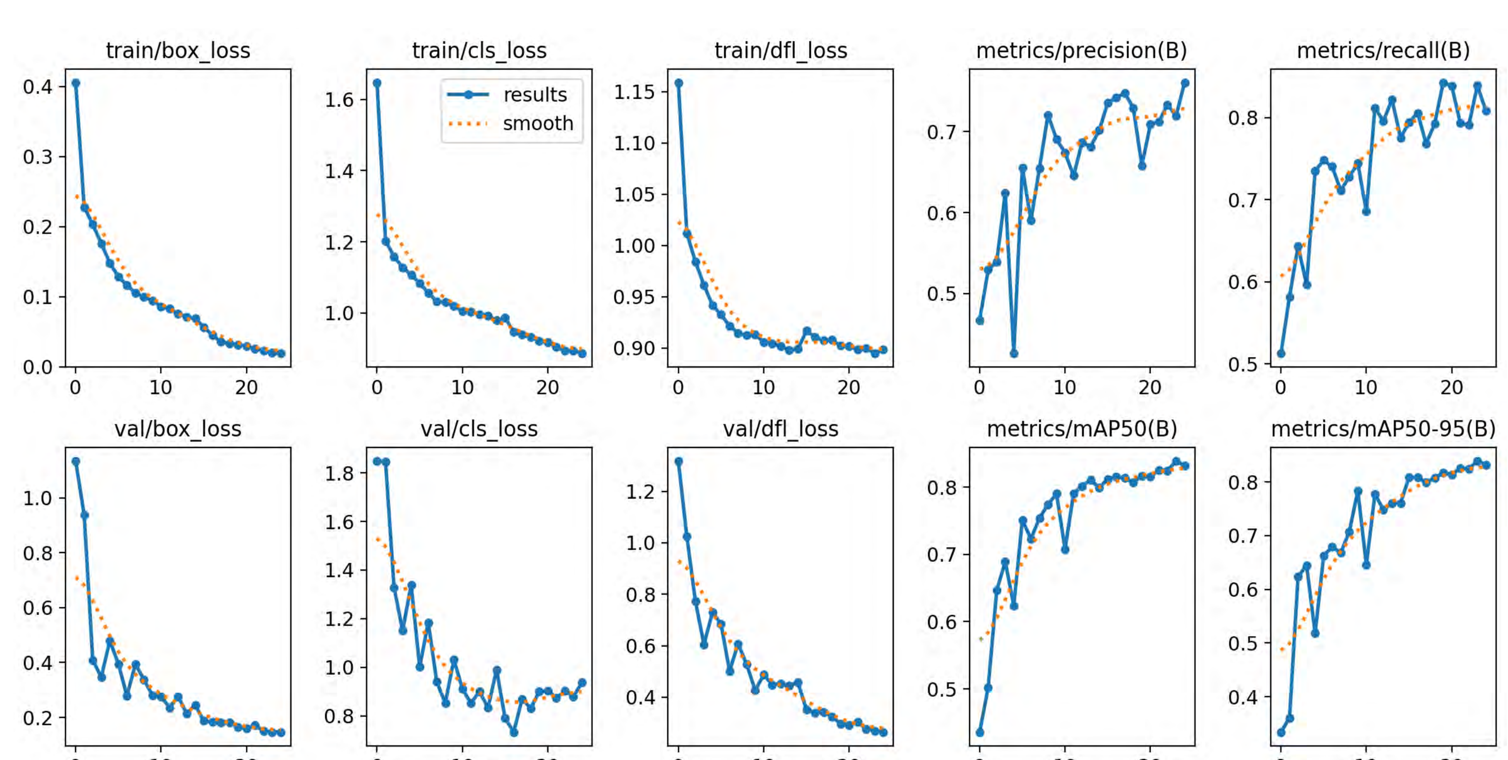

Figure 5: Plotted points and trend of model becoming more accurate and minimizing the loss for every epoch it runs.

Conclusion

- To conclude, the soil detection training model was successful and can be deployed on Roboflow and combine it with OpenCV to perform real time detection.
- Ineffectual in getting the LiDAR to work with the Pi.
- The adaptation of a Raspberry Pi for controlling the toy excavator lays a foundation for utilizing AI in construction.
- The platform 's compatibility with AI and autonomous tasks offers exciting opportunities for improving the decision-making and performance of the excavator.

Figure 6: Raspberry Pi controlling toy excavator.

Future Work

- Further training can be done on the model to improve its accuracy, potentially add additional classes apart from soil.
- Real world testing will validate the system's performance in unique scenarios, which will provide insight for industry adoption
- Examining the ways in which workers and autonomous excavators interact, we can enhance workflows and boost overall productivity.
- Researching collaborative tasks involving multiple autonomous excavators holds the potential to significantly enhance construction efficiency and project outcomes.
- Ensuring the industry's acceptance of autonomous excavators necessitates addressing safety concerns and establishing robust regulatory frameworks to mitigate potential risks.
- Integrate the RP LiDAR on Raspberry Pi.

References

Skalski, P. (2023, June 9). Train yolov8 on a custom dataset. Roboflow Blog. https://blog.roboflow.com/how-to-train-yolov8-on-a-custom-dataset/

Overseas help center - slamware - slamtec wiki. (n.d.). https://wiki.slamtec.com/display/SD/Overseas+Help+Center

Soil Detection Object Detection dataset and pre-trained model by Taufans. Roboflow. (n.d.). https://universe.roboflow.com/taufans-qnmrz/soil-detection-2uaco

Acknowledgements

Project supported by Project RAISE, U.S. Department of Education HSI-STEM award number P031C210118.

Alternate Text

Arai Leyva

Cal State Fullerton

‘Incorporating AI in Construction to Automate Excavator Operations’

Background:

- Artificial Intelligence has been increasingly integrated into the construction industry, with the goal of enhancing productivity and safety.
- Despite the advancements made, the practical use of AI in autonomous excavator operations is not extensively explored, mainly because of the complexity involved in their tasks.
- Conventional excavators often rely on manual controls and lack interactive features, which may limit their efficiency in modern construction tasks.
- To explore the possibilities of implementing advancements in automation into real excavators and overcome existing limitations, we adopted the approach of modifying a toy excavator.
- By using a single-board computer, sensors, and computer vision to modify the toy excavator, we want to enhance its functionality and provide a foundation for future improvements on excavators.

Figure 1: Scaled 1:14 Remote Controlled Toy Excavator

Figure 2: Raspberry Pi 4 attached with RP LiDAR

Methods:

Global Materials:

Remote Controlled Toy Excavator 1:14 Scale

Raspberry Pi 4 4GB with 16GB microSD Card

Raspberry Pi Camera Module with Case

Slamtec RP LiDAR A1M8 2D 360° 12 Meter Scanning Radius

Electrical Components (encoders, potentiometers, wiring cables)

Toy Excavator:

Disassemble remote control and identify the connections associated with the manual joysticks and buttons.

Next, desolder the existing manual joysticks and replace them with digital potentiometers, while simultaneously swapping out the manual buttons with transistors to enable control through the Raspberry Pi.

Raspberry Pi 4:

Use the Raspberry Pi Imager to download and flash Raspberry Pi OS Full version onto the microSD card.

Once the microSD card is inserted into the Pi's card slot, connect peripherals to set up our Pi and update/install any applications, libraries, or packages we may need.

Connect the digital potentiometers and transistors to the Raspberry PI, using the manufacturer's pin configuration data sheet as a reference. Utilize jumper wires to establish the required connections, including SPI communication lines (clock, data in, and data out) and the essential power supply lines.

Figure 3: Disassembled remote control connected to potentiometers through breadboard into the Raspberry Pi.

Methods Continued:

Computer Vision:

- Utilized Google Colab for model training due to access to powerful GPUs provided by them. Obtained soil image data set using a snippet of code from Roboflow for training a soil detection model.
- Prepared required directories and trained theYOLOv8model using a concise one-line code with the assistance of the GPUs from Colab.
- To make the model compatible with the Raspberry Pi, it was converted to a suitable format, ensuring integration with the environment. Once we have the required dependencies and libraries (Python, OpenCV, YOLOv8) setup OpenCV on the Pi to process the live feed from the camera module.

RP LiDAR:

- Increase Raspberry Pi Swap memory to allow it to use the swap files as additional memory when RAM is exhausted.
- Download the RP LiDAR SDK on GitHub which gives access to libraries and dependencies. Along with that we download a ROS package to evaluate the data provided by the LiDAR.
- Error compiling a workspace on the SDK due to manufacturer not keeping the kit up to date with newer version of operating systems and IDE's.

Figure 4: In Google Colab. (Top) Snippet of code from Roboflow to extract the dataset into the notebook. (Bottom) Incisive one-line code to train the model.

Results:

- Trained model using YOLOv8 and Google Collab to detect soil.
- Reconfigured the buttons on remote control to be controlled by the potentiometers proving the Raspberry Pi connection was effective.

Figure 5: Plotted points and trend of model becoming more accurate and minimizing the loss for every epoch it runs.

Conclusion:

- To conclude, the soil detection training model was successful and can be deployed on Roboflow and combine it with OpenCV to perform real time detection.
- Ineffectual in getting the LiDAR to work with the Pi.

- The adaptation of a Raspberry Pi for controlling the toy excavator lays a foundation for utilizing AI in construction.
- The platform 's compatibility with AI and autonomous tasks offers exciting opportunities for improving the decision-making and performance of the excavator.

Figure 6: Raspberry Pi controlling toy excavator.

Future Work:

Further training can be done on the model to improve its accuracy, potentially add additional classes apart from soil.

- Real world testing will validate the system's performance in unique scenarios, which will provide insight for industry adoption
- Examining the ways in which workers and autonomous excavators interact, we can enhance workflows and boost overall productivity.
- Researching collaborative tasks involving multiple autonomous excavators holds the potential to significantly enhance construction efficiency and project outcomes.
- Ensuring the industry's acceptance of autonomous excavators necessitates addressing safety concerns and establishing robust regulatory frameworks to mitigate potential risks.
- Integrate the RP LiDAR on Raspberry Pi.

References:

Skalski, P. (2023, June 9). Train yolov8 on a customdataset. Roboflow Blog. https://blog.roboflow.com/how-to-train-yolov8-on-a-custom-dataset/

Overseas help center - slamware - slamtec wiki. (n.d.). https://wiki.slamtec.com/display/SD/Overseas+Help+Center

Soil Detection Object Detection dataset and pre-trained model by Taufans. Roboflow. (n.d.). https://universe.roboflow.com/taufans-qnmrz/soil-detection-2uaco

Acknowledgements:

Project supported by Project RAISE, U.S. Department of Education HSI-STEM award number P031C210118.