

## How LeRobot Hackathon Winners Built AI-Robots With NVIDIA Isaac GR00T



NVIDIA Omniverse  
49K subscribers

Subscribe

145



Share

Ask

Save

...

4,426 views Streamed live on Jul 31, 2025

Join us for a special livestream featuring two winning teams from the LeRobot Worldwide Hackathon: Spice Terminator and LeDetective DaVinci. See how these teams used GR00T N1.5 and the SO-101 robot to build AI-powered systems that understand natural language and perform complex real-world tasks. From robotic cleaning driven by vision-language-action models to AI agents solving mysteries like a detective, this session is full of innovation, creativity, and real-world applications of physical AI.

Check out the full calendar for all of our upcoming events → <https://nvda.ws/3JqaWnA>

Get Started → <https://nvda.ws/4cZAZO1>

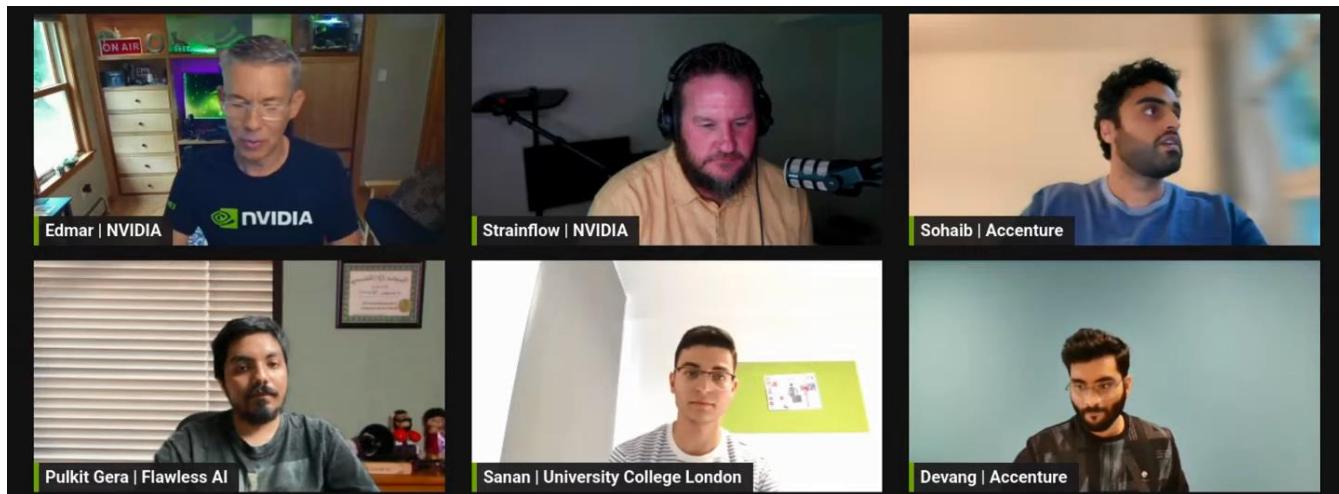
Explore OpenUSD → <https://nvda.ws/3CeozBQ>

Join the Community → <https://nvda.ws/3ZMfc6e>

AI-generated video summary · Quality and accuracy may vary. ⓘ



This NVIDIA Omniverse livestream showcases two LeRobot Hackathon winning teams. Teams Spice Terminator and LeDetective DaVinci demonstrate AI-powered robots performing complex tasks. They detail their innovative solutions using NVIDIA Isaac GR00T and the SO-101 robot arm, from initial concept to final demo.





# LeDetective DaVinci

Devang Darode  
Muhammad Sohaib Arshid  
Artur Habuda  
Shukrullo Nazirjanov  
Philipp Schmidt

## Most Common Barriers For Students, Hobbyist And Starters

- Getting hardware is... well, hard
- Lack of reliable data sets
- Training AI models for robots
- Lack of Community and Collaboration for an individual
- Programming robots isn't like coding software

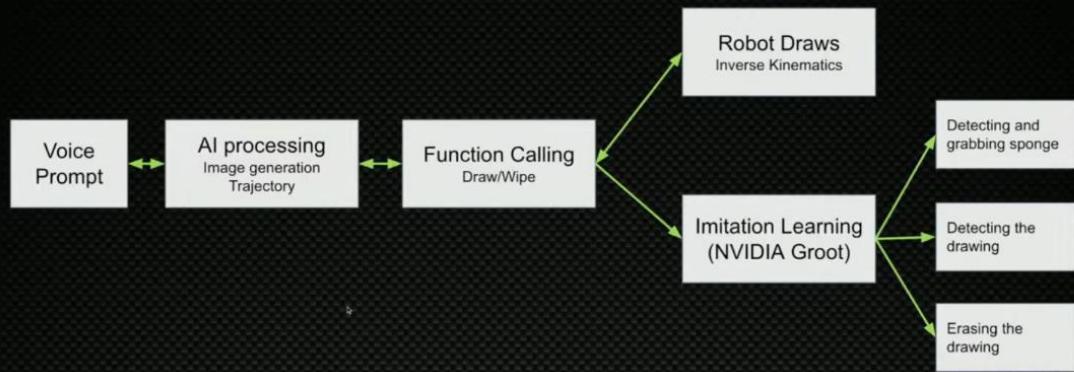
Today we will see how these barrier are being strategically brought down

## LeRobot Worldwide Hackathon

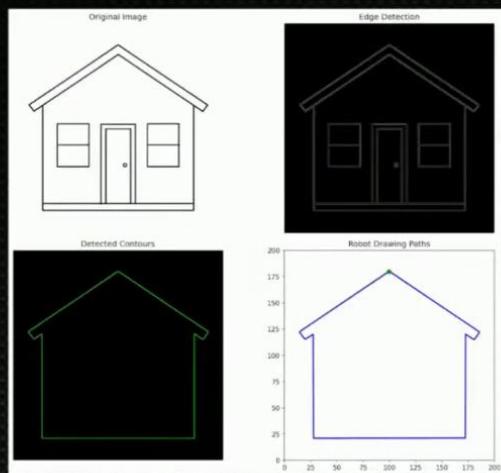


- Two days, 3300 participants, 100+ locations across globe
- Form team with strangers
- Generate creative ideas
- Design solutions, implement, and debug
- Present working demo and pitch the idea to the panel

# Project Idea



## Drawing Trajectory



```
{
  "paths": [
    {
      "points": [
        {
          "x": 99.2,
          "y": 165.4
        },
        {
          "x": 35.7,
          "y": 110.0
        },
        {
          "x": 35.9,
          "y": 107.8
        },
        {
          "x": 46.9,
          "y": 108.0
        }
      ]
    }
  ]
}
```

## HuggingFace LeRobot S0-101 ARM

- Open source
- Affordable
- Minimal barrier to entry
- Suitable For Rapid prototyping
- Community (Hugging Face ecosystem)



# Phosphobot

- Open Source
- Control your robots to record datasets in minutes
- Train Action models such as ACT, π0 or gr00t-n1.5 with one click
- Compatible with the SO-100, SO-101 and others
- Dev-Friendly API
- Fully compatible with LeRobot and HuggingFace



The screenshot shows the Phosphobot dashboard with a dark theme. On the left is a sidebar with navigation links: Dashboard, Control & Record (selected), Browse Datasets, Calibration, AI & Training, Admin Configuration, API Documentation, Camera Overview, and Network Management. The main area has three main sections: 'Control and Record' (with a status message 'Status: Disconnected'), 'AI Training and Control' (with buttons for 'Train an AI Model' and 'Go to AI Control'), and 'Advanced Settings'. Each section has a 'Learn More' button.

The screenshot shows the 'Robot Control' interface. It features a 3D model of a robot with movement arrows labeled 'Front Arrow', 'Back Arrow', 'Left Arrow', 'Right Arrow', 'Space Bar', and 'F Key'. Above the robot is a control bar with buttons for 'Start recording robot movements', 'REC', 'STOP', 'DISCARD', 'REPLAY', and a camera icon. Below the robot is a keyboard control panel with keys for ARROWUP, ARROWDOWN, ARROWLEFT, ARROWRIGHT, F, V, and a numeric keypad. At the bottom, there's a status bar with 'Select robot to move' and a timer '0:04'.

Safari File Edit View History Bookmarks Window Help

localhost phosphobot Documentation Sign in Sign up

## Browse Datasets

Navigation

- Dashboard
- Control & Record
- Control Robot
- Browse Datasets**
- Calibration

AI & Training

- AI Training
- AI Control

Advanced Settings

- Admin Configuration
- API Documentation
- Camera Overview
- Network Management

Name:  [+ Add dataset from hub](#)

Safari File Edit View History Bookmarks Window Help

localhost phosphobot Documentation Sign in Sign up

## Calibration

Navigation

- Dashboard
- Control & Record
- Control Robot
- Browse Datasets
- Calibration**

AI & Training

- AI Training
- AI Control

Advanced Settings

- Admin Configuration
- API Documentation
- Camera Overview
- Network Management

Calibration is only required if you're not a physical customer.  
Unless you got from `robotics.phosphobot.g`, are already calibrated.

### Prepare Your Robot

**Safety Warning**  
Make sure you can safely catch your robot. Calibration disables torque.

**Start Calibration**

Safari File Edit View History Bookmarks Window Help

localhost phosphobot Documentation phosphobot API GitHub

# Welcome to phosphobot

An absurdly simple way to train AI models for real-world robots, built for ML engineers.

Phospho is how ML engineers make real robots intelligent. We provide the hardware, libraries, and remote control capabilities so developers can collect data, train AI models and deploy applications to real robots in minutes instead of months.

### Highlights

- Control your robots to record datasets in minutes with a keyboard, a gamepad, a leader arm, and more
- Train Action models such as ACT, s0 or gr001-vt5 with one click
- Compatible with the SG-100, SG-101, Unitec Go2, Agiles Pirot...
- Dev-friendly API
- Fully compatible with Leftebot and HuggingFace
- Runs on macOS, Linux and Windows
- Meta Quest app for teleoperation
- Supports most cameras (classic, depth, stereo)
- Open Source: Extend it with your own robots and cameras

### Installation

In a terminal, run the following command:

```
git clone https://github.com/phospho-project/phosphobot.git
cd phosphobot
curl -fSL https://raw.githubusercontent.com/phospho-project/phosphobot/main/install.sh | bash
```

Then, fire up the the server:

```
phosphobot run
```

## Data Example (LeRobot Data Visualizer)

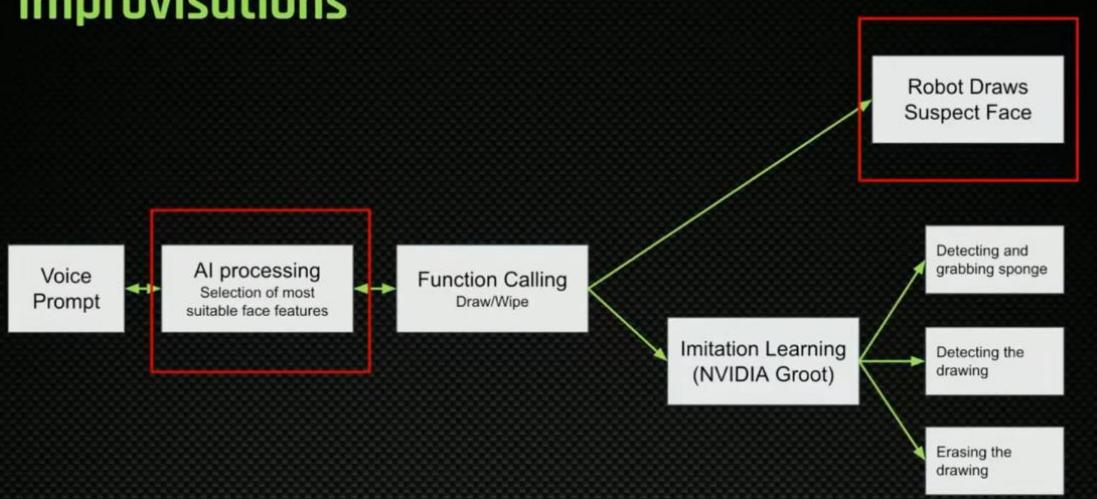


### NVIDIA GROOT-N1-2B (Imitation Learning)

- Open Source Robot Foundation Model
  - GROOT N1 can easily generalize across common tasks
  - Few episodes of data are enough to train a reasonably good imitation model
  - ~120 samples over two days for training
  - Trained for 10 epochs

Succeeded	arturasah-gr00t-wipe3...			gr00t	<pre>epochs: 10 data_dir: data/ batch_size: 49 output_dir: outputs/ learning_rate: 0.0001 train_test_split: 1 path_to_gr00t_repo: ..</pre>
Succeeded	arturasah-gr00t-wipe2...			gr00t	<pre>epochs: 10 data_dir: data/ batch_size: 49 output_dir: outputs/ learning_rate: 0.0001 train_test_split: 1 path_to_gr00t_repo: ..</pre>

# Improvisations

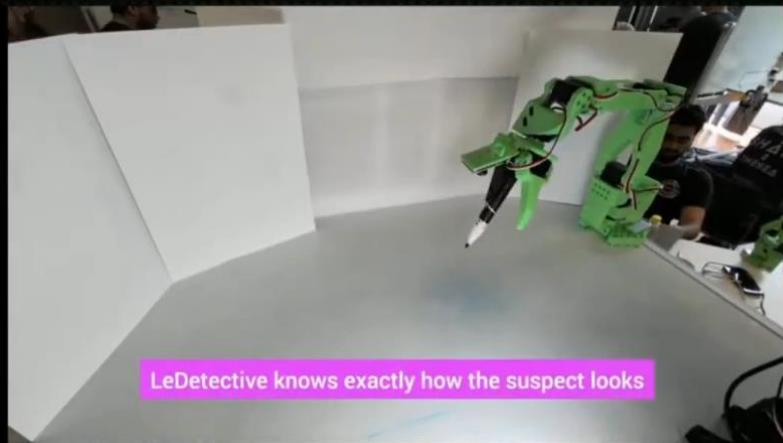


## And The Result...?

[LeDetective DaVinci]

I couldn't see him well but he was running really fast. It looked as he knew exactly where to go after stealing my wallet as if he was very familiar with the streets in München. He was also big and tall (maybe bigfoot idk)

## And The Result...?



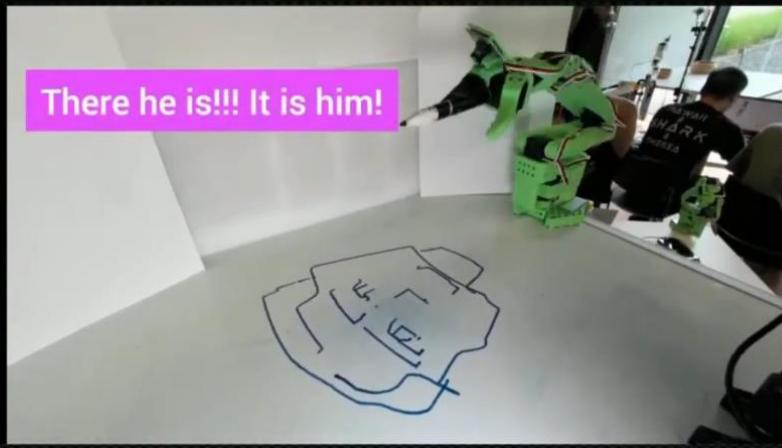
LeDetective knows exactly how the suspect looks

## And The Result...?

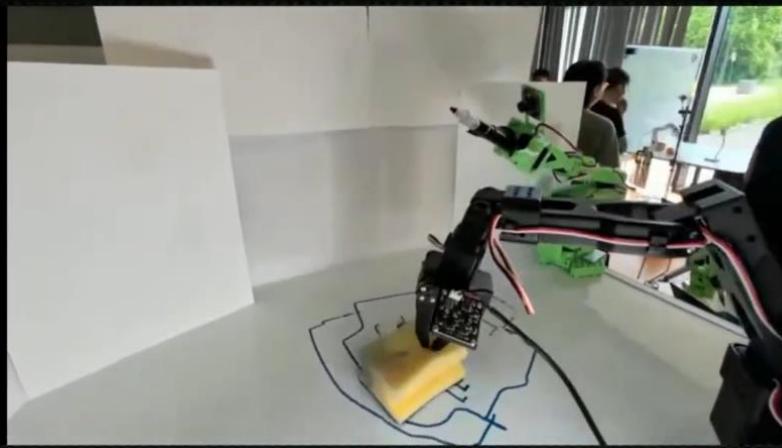


He is always alert, he has big eyes

**And The Result...?**



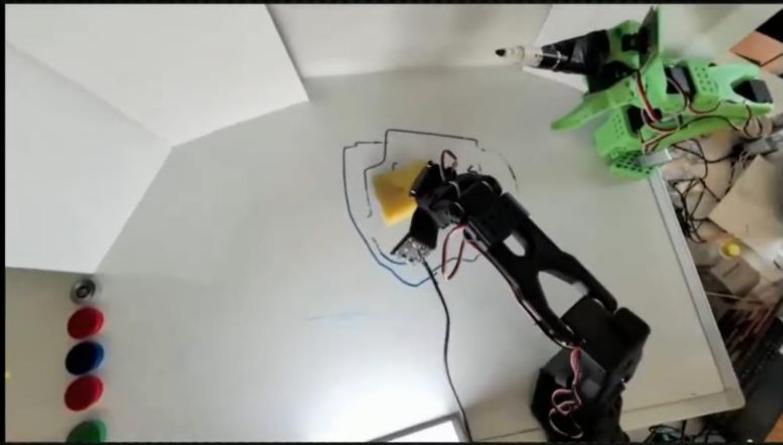
**And The Result...?**



**And The Result...?**



**And The Result...?**



**The team**

