Code Generation with Vision Language Models for Robot Arms Application

Members: Tran Quang Minh, Luu Trong Hieu, Nguyen Cong Khanh, Nguyen Quang Trung

Department of Artificial Intelligence
FPT University - VietDynamic JSC

Internship Presentation - Fall 2025

Methodology

01 Overview

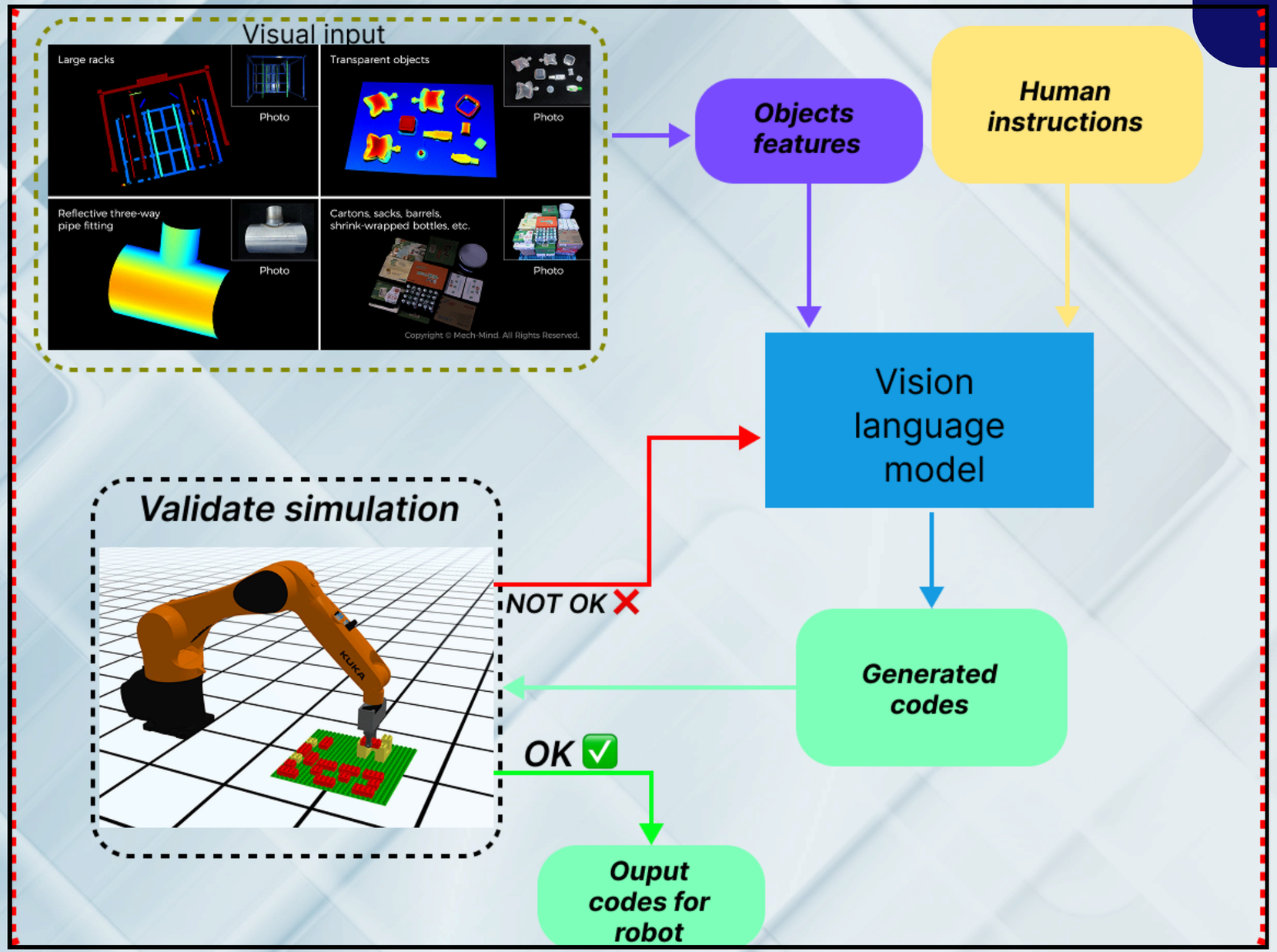
02 Systems Design

03 App

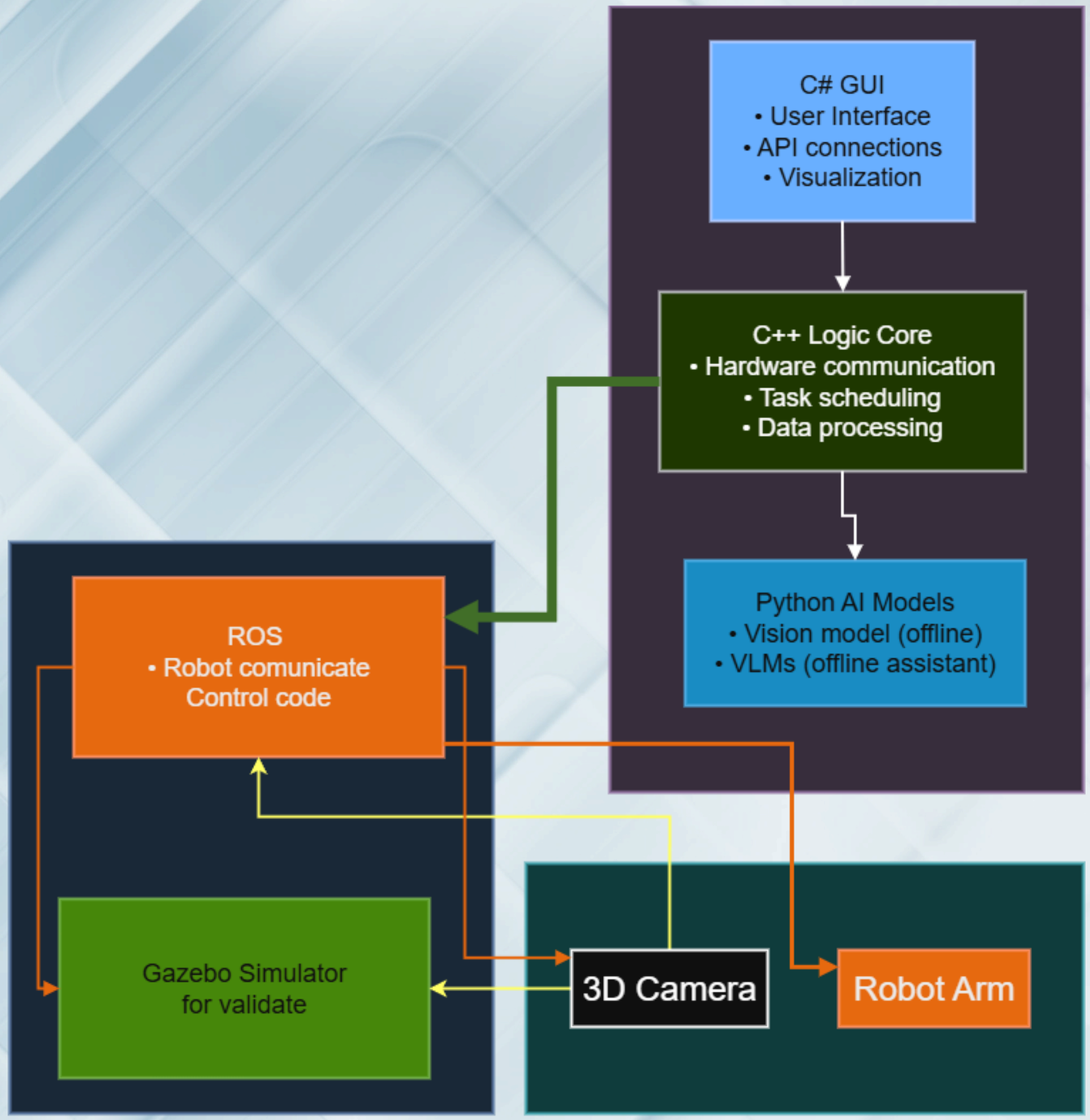
04 Vision Language Models

05 Vision Models

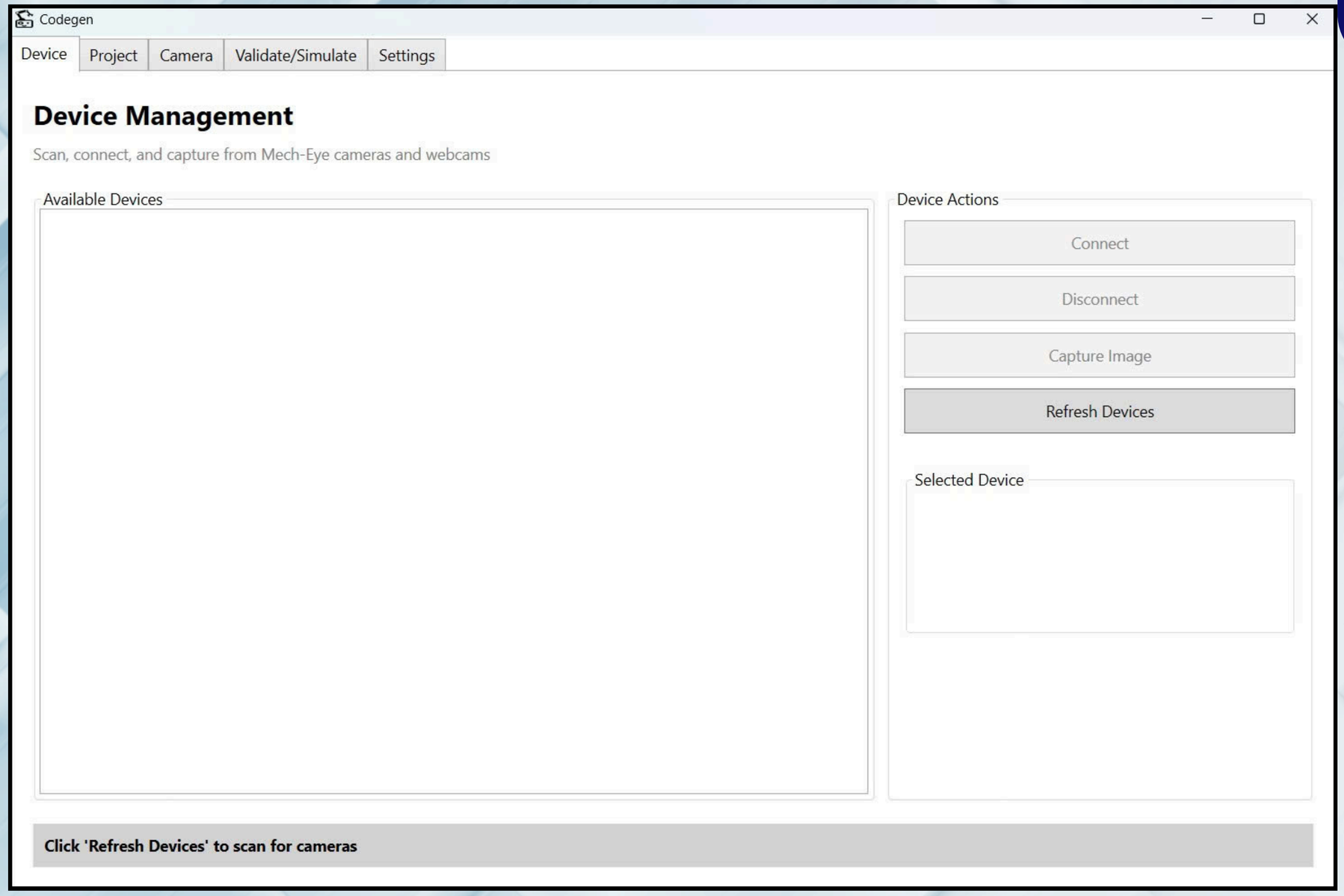
Overview


Visual input
Large racks
Photo
Transparent objects
Photo
Reflective three-way pipe fitting
Photo
Cartons, sacks, barrels, shrink-wrapped bottles, etc.
Photo
Copyright © Mech-Mind. All Rights Reserved.
Objects features
Human instructions
Vision language model
Validate simulation
NOT OK
Generated codes
OK
Ouput codes for robot

Systems Design



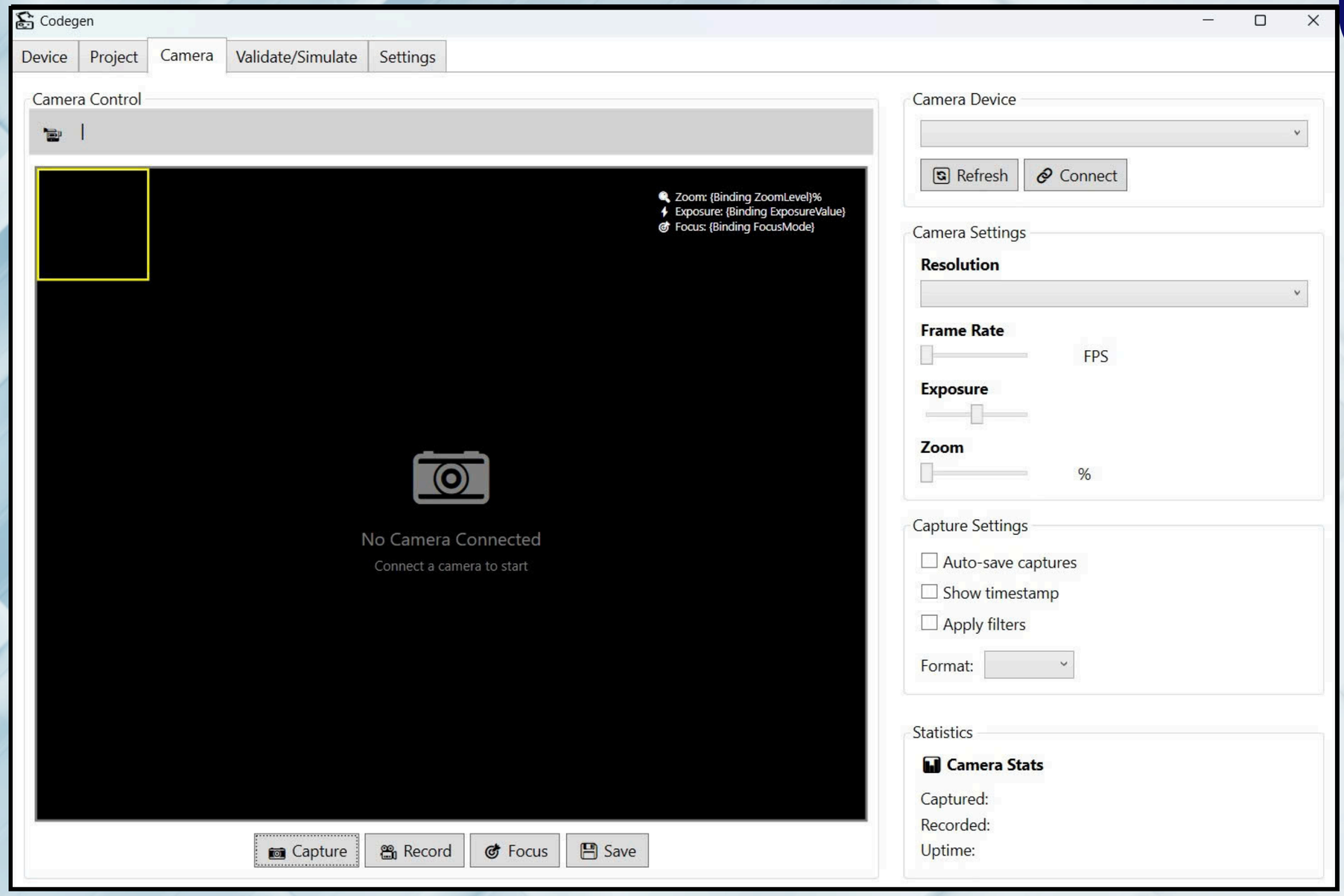
App demo



Device Management:

- Automatic Mech-Eye camera detection and connection
- Simultaneous scanning and management of multiple camera devices
- Simple connect /disconnect interface for hardware control

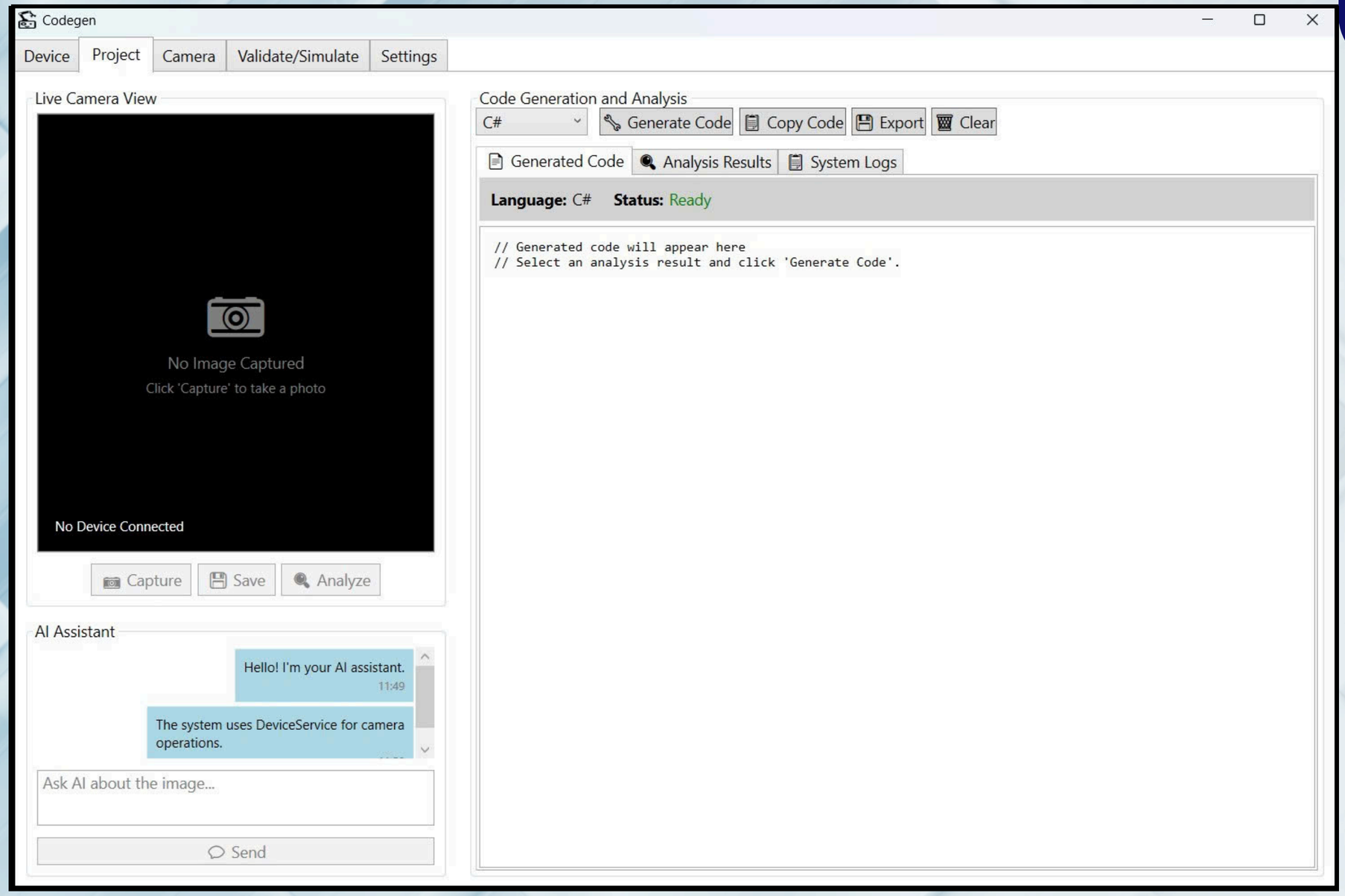
App demo



Camera Control

- Advanced parameter adjustment: Zoom, Exposure, Focus
- Customizable resolution and frame rate settings
- Capture settings with auto-save and timestamp features

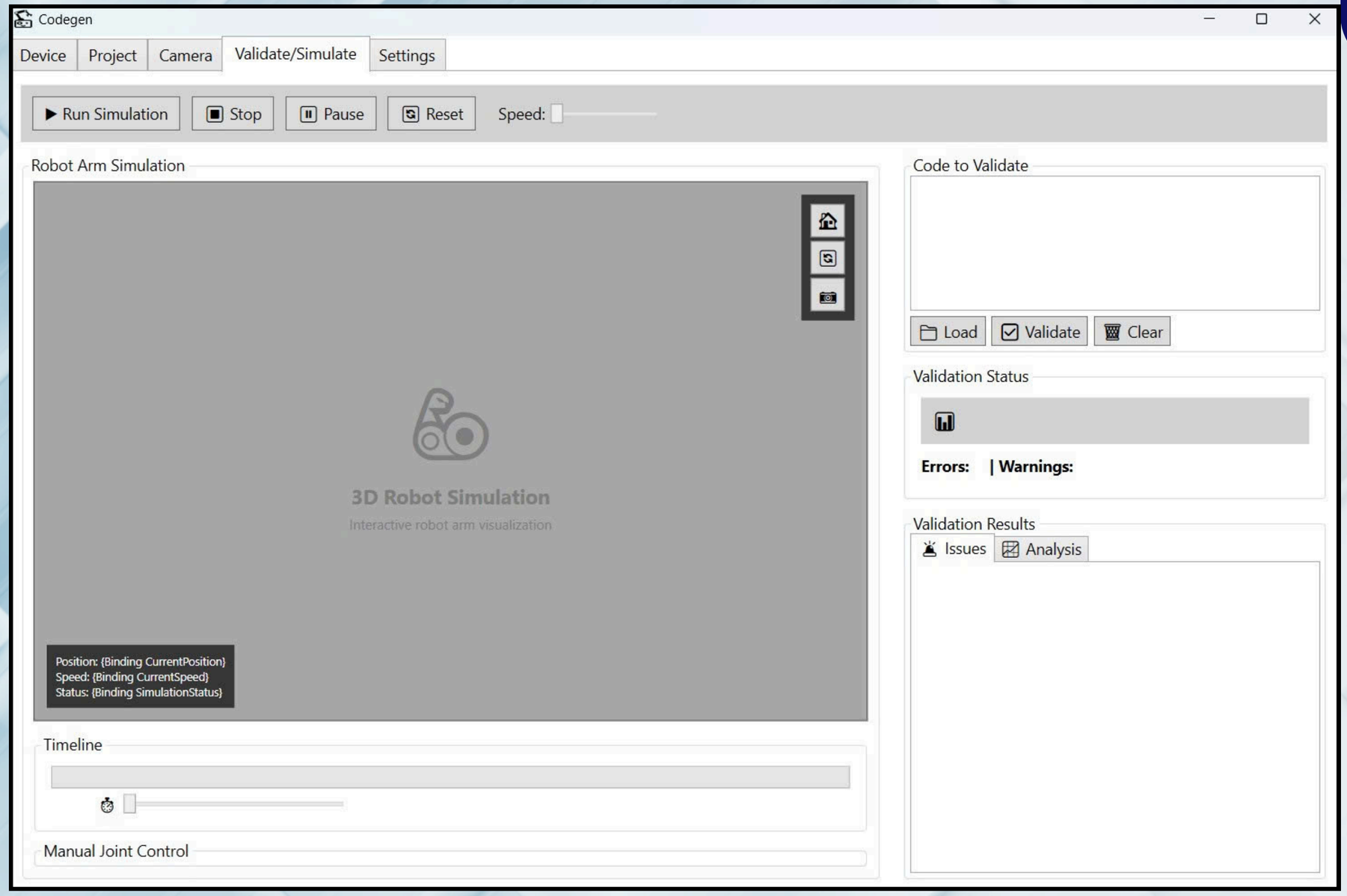
App demo



Main Workspace

- Live camera view with real-time preview
- Integrated AI Assistant for code generation support
- Code analysis and generation panel
- Multi-language programming support (C#, Python, etc.)

App demo



Simulation & Validation

- 3D robot arm simulation environment
- Code validation with error/warning detection
- Manual joint control and timeline management
- Real-time position and speed monitoring

Systems Design

01 GUI Module

02 Logic Core Module

03 VLM and Vision Integration Module

04 Simulate and Validate Module

GUI Module

- **Core Function**: Handles user interaction.
- **Key Features:**
  - Task input via natural language.
  - Parameter configuration & high-level command interpretation.
  - Real-time monitoring (live camera feeds, point cloud visualization).
  - Code generation progress tracking.
- **System Configuration & Control:**
  - VLM model selection & configuration.
  - Camera calibration & image/point cloud capture.
  - Hardware settings management.

Logic Core Module

- **Core Function**: Central coordination hub managing all inter-module communication and data flow synchronization
- **Key Features:**
  - Ensures seamless coordination between all system components
  - Handles hardware communication protocols
  - Processes user inputs and system data
  - Maintains smooth data flow across modules

VLM and Vision Integration Module

- **Core Function**: Multimodal processing center combining visual data and language instructions for code generation
- **Key Features:**
  - Integrates vision models for object detection and scene understanding
  - Processes multimodal inputs (images + text instructions)
  - Generates executable code through selected VLM models
  - Manages model inference and output formatting

Simulate and Validate Module

- **Core Function**: Virtual testing environment for code validation and performance assessment
- **Key Features:**
  - Executes generated code in simulated robotics environment
  - Validates code functionality and task completion
  - Assesses performance metrics (accuracy, efficiency, robustness)
  - Provides feedback for system improvement and code refinement

Vision Language Models (VLMs)

Model Selection & Rationale:

- Base Model: MobileVLM V2
- Key Advantages:
  - Lightweight and efficient architecture
  - Optimized for limited computational resources
  - Enhanced multimodal input handling
  - Superior performance-speed balance (as shown in Fig. 1)

Customization Strategy

- Fine-tuning Approach:
  - Specialized on custom robotic task datasets
  - Hyperparameter optimization for code generation
  - Context-aware adaptation for robotic applications
- Integration:
  - Real-time code generation capability
  - Dynamic adaptation to environmental changes

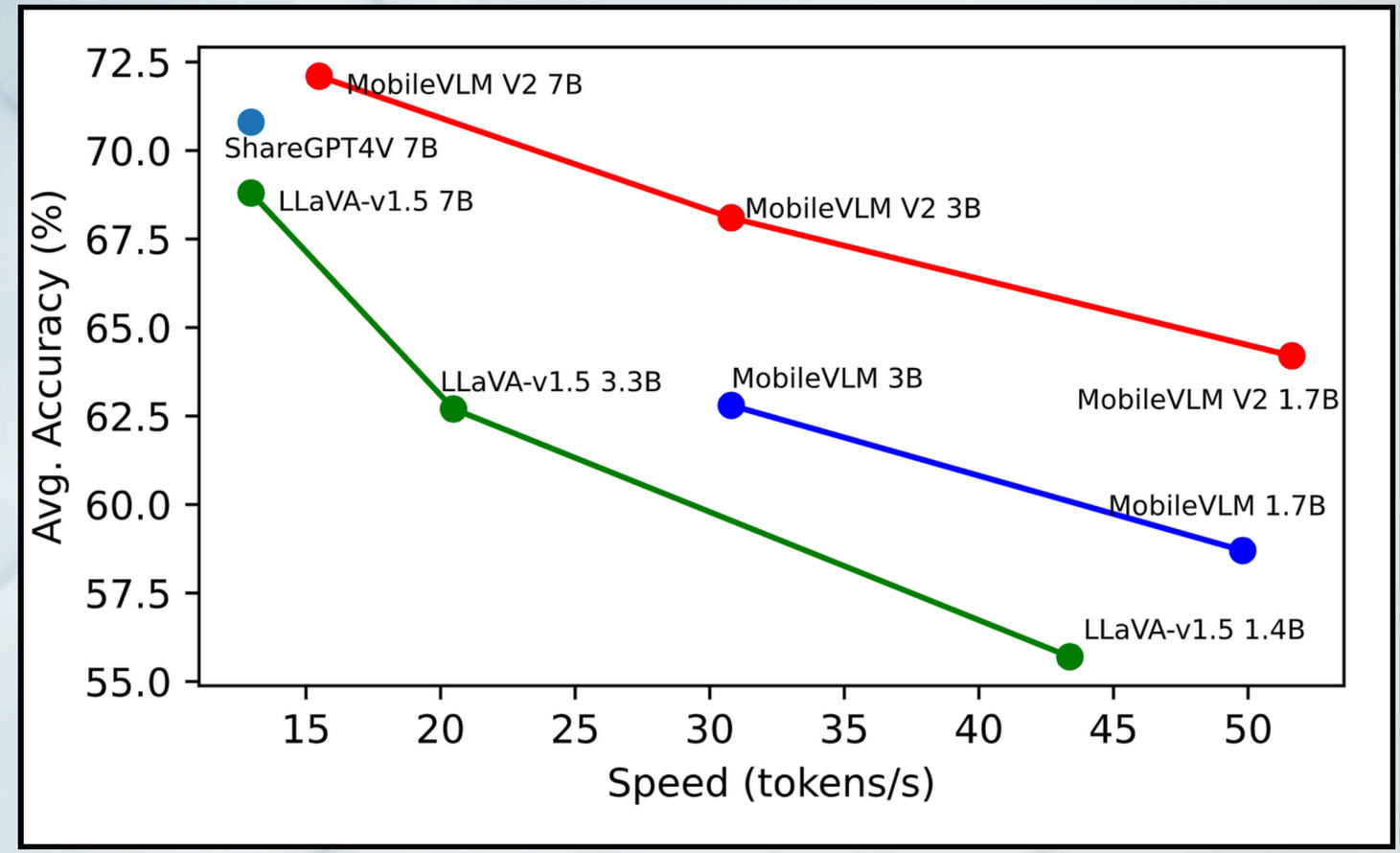


Fig. 1: SOTA VLMs Comparison – Avg. Accuracy vs. Speed (NVIDIA Jetson Orin, llama.cpp). MobileVLM V2 Achieves New SOTA with Faster Inference.

MobileVLM V2 Technical Advantages

Vision Language Models (VLMs)

Architectural Improvements:

- Enhanced model architecture over previous version
- Advanced training techniques
- Increased robustness to varied input types
- Better scalability (as demonstrated in Fig. 2)

Performance Validation:

- State-of-the-art results on multiple benchmarks
- Faster inference speed on edge devices (NVIDIA Jetson Orin)
- Proven scalability - performance improves with model size
- Maintained efficiency despite performance gains

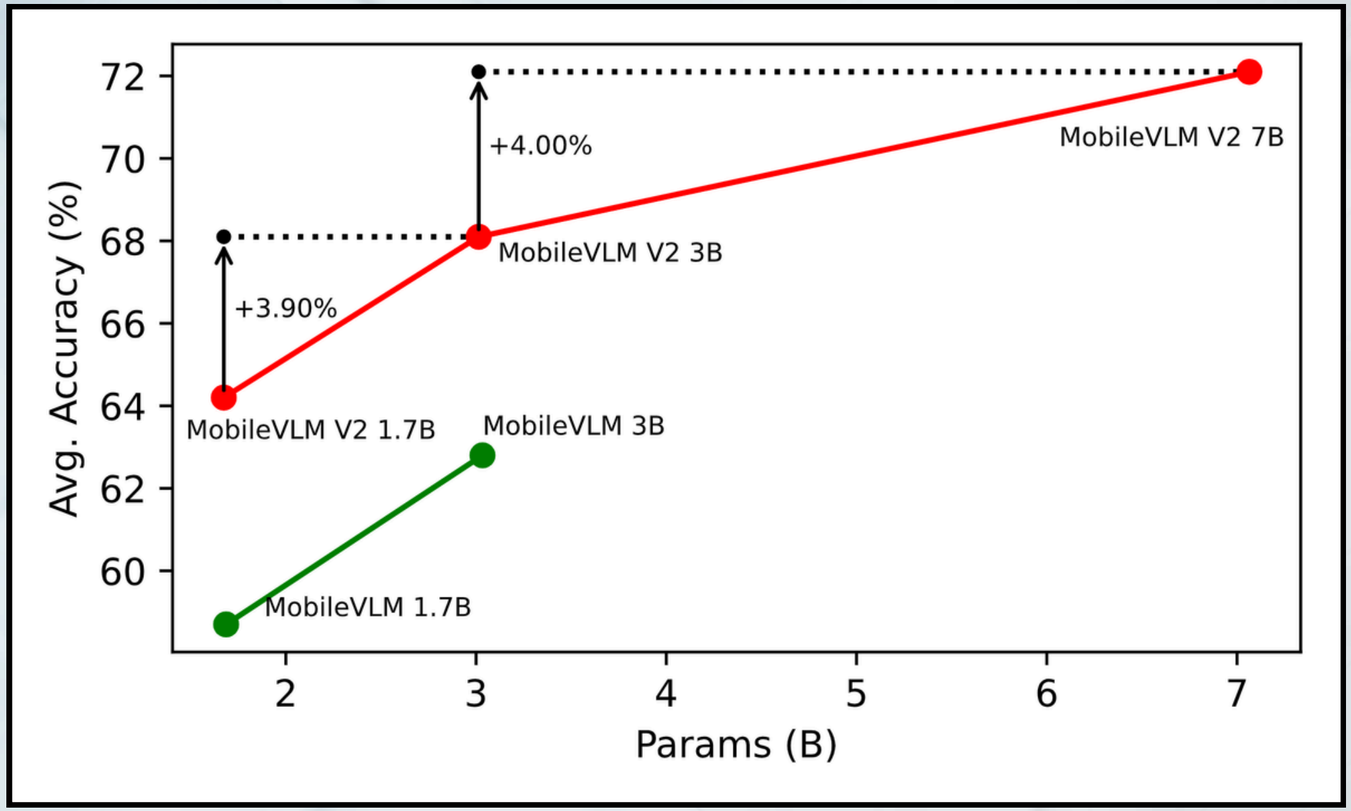


Fig. 2: MobileVLM V2 Scaling – Avg. Accuracy vs. Params (B). Performance Improves with Size, Validating Architecture and Training.

Vision Models

Initial Plan & Hardware Setup

- Camera Systems: Mech-EYE & Intel RealSense 3D cameras
- Initial Model Selection: YOLOv11 for balance of speed and accuracy
- Intended Application: Real-time object detection for robotic manipulation
- Integration Goal: Provide accurate visual context for VLM code generation

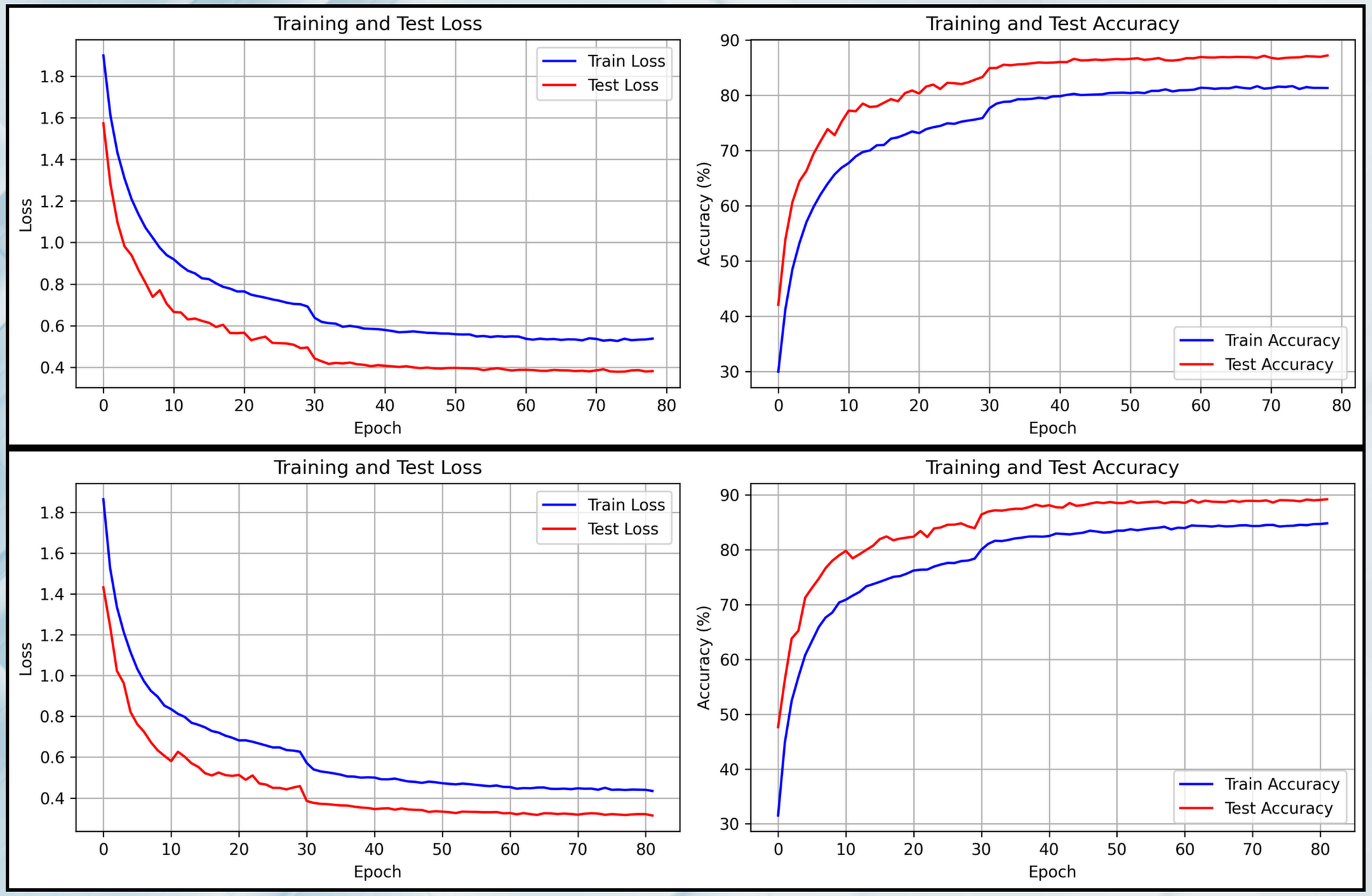
Technical Challenge & Adaptation

- Problem Identified: YOLOv11 too computationally intensive
- Hardware Constraints: Limited resources on deployment platform
- Solution: Custom MobileNetV2-based model from AITA-lab
- Advantage: Optimized for embedded systems while maintaining accuracy

Vision Models

Custom Vision Model & Experimental Validation

- **Model Architecture & Specifications**
  - Base Architecture: MobileNetV2 backbone with SSD detection head
  - Model Size: 0.34M parameters, 0.06 GFLOPs
  - Key Feature: Lightweight yet competitive performance
  - Deployment Ready: Suitable for resource-constrained robotic systems
- **Experimental Results & Performance**
  - Classification Task: CIFAR10 & industrial objects dataset
  - Object Detection: Pascal VOC benchmark evaluation
  - Training Performance: Stable convergence curves (Fig. 3)
  - Efficiency: Significant reduction in computational requirements
  - Practicality: Demonstrated viability for real-world robotic applications


Training and Test Loss
Train Loss
Test Loss
Loss
Epoch
Training and Test Accuracy
Train Accuracy
Test Accuracy
Accuracy (%)
Epoch
Training and Test Loss
Train Loss
Test Loss
Loss
Epoch
Training and Test Accuracy
Train Accuracy
Test Accuracy
Accuracy (%)
Epoch

Question and Answer...

Thank You!