**🧾 🎯 Project Title: CUSTOM OBJECT DETECTION FOR MOBILE ROBOTS  
📅 Project Timeline:** February 2021 – October 2021  
🎥 YouTube Demo: [https://youtu.be/p0wAyzASfN0](https://youtu.be/p0wAyzASfN0?utm_source=chatgpt.com)  
📦 GitHub Source Code: <https://github.com/IvanSicaja/2021.02.07_GitHub_PRJ_Custom-Object-Detection-for-Mobile-Robots>   
----------------------------------------------------------------------------------------------------------------

🏷️ My Personal Profiles: ⬇︎  
🎥 Video Portfolio: To be added  
📦 GitHub Profile: [https://github.com/IvanSicaja](https://github.com/IvanSicaja?utm_source=chatgpt.com)  
🔗 LinkedIn: [https://www.linkedin.com/in/ivan-si%C4%8Daja-832682222](https://www.linkedin.com/in/ivan-si%C4%8Daja-832682222?utm_source=chatgpt.com)  
🎥 YouTube: [https://www.youtube.com/@ivan\_sicaja](https://www.youtube.com/@ivan_sicaja?utm_source=chatgpt.com)  
----------------------------------------------------------------------------------------------------------------

### 📚🔍 Project description: ⬇︎⬇︎⬇︎

### 💡 App Purpose

Development of a **custom object detection system** for autonomous mobile robots competing in the **RoboCup Rescue Maze category**. The project implemented **custom-trained convolutional neural networks (CNNs)** for detecting and classifying objects, with strong emphasis on **data preprocessing, network training, testing, and speed optimization** to ensure deployment feasibility on a **Raspberry Pi 4B**.

The main goal was to provide **fast and reliable object detection** on resource-limited hardware. The system allowed the robot to recognize **characters and shapes** in the maze environment, contributing to navigation, decision-making, and competition efficiency.

### 🧠 How It Works

* **Data Preprocessing**: Image transformation, contour extraction, resizing, normalization, and preparation of training datasets.
* **Custom CNN Training**: Convolutional network trained on alphanumeric characters and simple shapes for maze detection tasks.
* **Real-Time Detection**: Using **OpenCV + Keras**, the system processed live camera input, extracted contours, and classified objects via the trained CNN.
* **Optimization**: Integrated **profiling tools (cProfile, pstats, SnakeViz)** to measure execution time, optimize bottlenecks, and ensure speed suitable for real-time use on Raspberry Pi 4B.
* **Testing & Validation**: Combined GUI-based controls (OpenCV Trackbars) with step-by-step or continuous evaluation modes for flexible testing.

### ⚠️ Note

None.

### 🔧 Tech Stack

### Python, OpenCV, NumPy, Keras, TensorFlow, cProfile, pstats, SnakeViz, Raspberry Pi 4B, Custom CNNs, Image Preprocessing, Contour Detection, Real-Time Object Recognition, Test Tech, Custom Solutions, Prototyping, Research, Testing, Optimization, Speed Optimization

### 📸 Project Snapshot

Not available.

### 🎥 Video Demonstration

Not available.

### 📣 Hashtags Section

**# #ObjectDetection #Robotics #RoboCup #CNN #RaspberryPi #ComputerVision #DeepLearning #Optimization #Testing #CustomSolutions #STEM**