Autonomous Vehicle Research Project

Students: Brendan Flood (University of Dallas)
Patrick Lau (City University of Hong Kong)
Julian Halloy (Maryville High School)
Mentors: A. Ayala (UTK), K. Wong (UTK)

Construct the Car
- Assemble the robot car
- Connect with Arduino Uno board

Setup Jetson Nano
- Install system packages and prerequisites
- Configure the development environment

Jetson Nano Car
- Connect Jetson Nano with Arduino Uno
- Send command from Nano by PySerial

Collect images
- Connect camera module to the Jetson Nano
- Drive the car and capture images for training data

Label the data
- Create folder for each object to predict
- Put the images in its respective folder

Model training
- Train by ImageAI with ResNet model
- Generate .h5 file and .json file

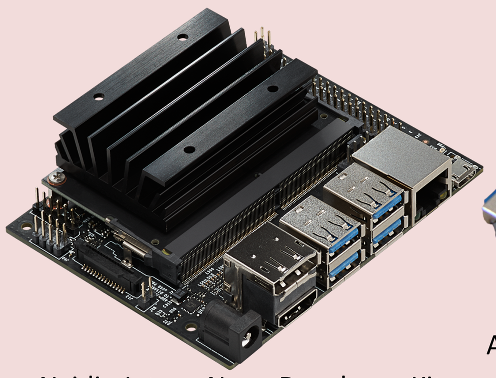
Translate to MAGMADNN
- Write, train, and test simple nn models in MAGMADNN

Introduction

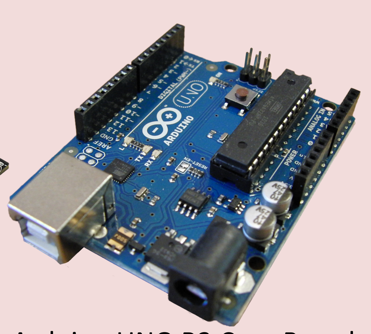
As technology advances, image recognition software becomes more capable than ever before, to the point that a computer with such a capability can potentially be used to replace a human driver in a car. The purpose of this experiment is to design a miniature version of a self-driving car which uses a convolutional neural network to recognize images of and navigate around a closed environment.

Equipment Setup

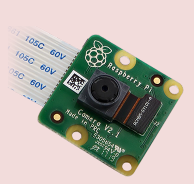
Hardware



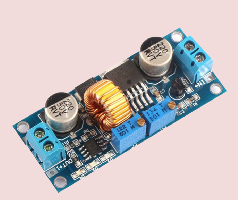
Nvidia Jetson Nano Developer Kit



Arduino UNO R3 Core Board



Raspberry Pi Camera Module V2



DROK Buck Converter Module

11.1V 2200mA LiPo Battery

PAUO5 Wi-Fi Dongle

Software

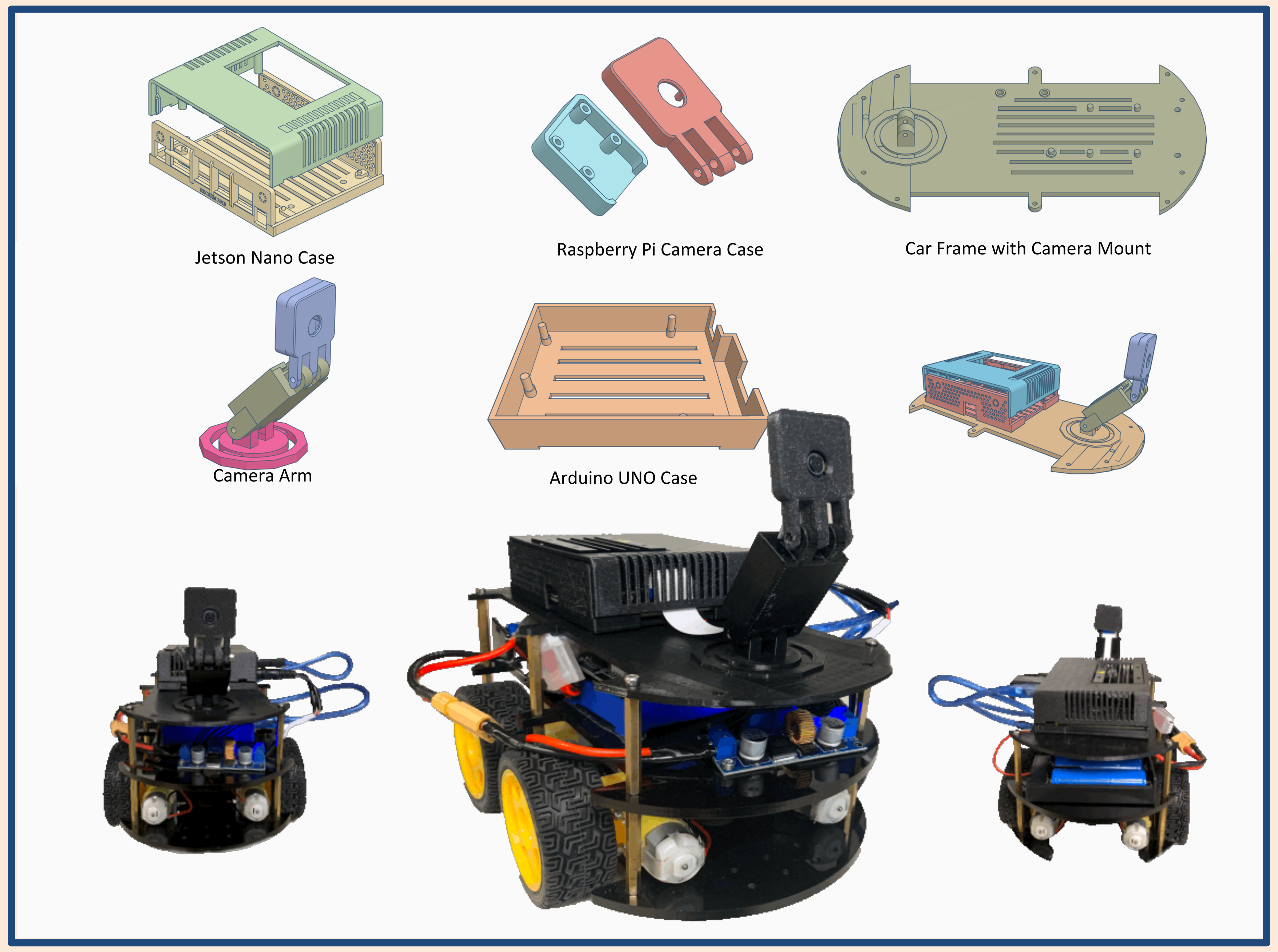
OpenCV
PySerial
TensorFlow
ImageAI
gstreamer
Arduino
TinkerCAD
NumPy
Keras

Neural Network

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix}3\times3, 64\\3\times3, 64\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 64\\3\times3, 64\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 64\\3\times3, 64\\1\times1, 256\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 64\\3\times3, 64\\1\times1, 256\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 64\\3\times3, 64\\1\times1, 256\end{bmatrix}\times3$
conv3_x	28×28	$\begin{bmatrix}3\times3, 128\\3\times3, 128\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 128\\3\times3, 128\end{bmatrix}\times4$	$\begin{bmatrix}1\times1, 128\\3\times3, 128\\1\times1, 512\end{bmatrix}\times4$	$\begin{bmatrix}1\times1, 128\\3\times3, 128\\1\times1, 512\end{bmatrix}\times4$	$\begin{bmatrix}1\times1, 128\\3\times3, 128\\1\times1, 512\end{bmatrix}\times8$
conv4_x	14×14	$\begin{bmatrix}3\times3, 256\\3\times3, 256\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 256\\3\times3, 256\end{bmatrix}\times6$	$\begin{bmatrix}1\times1, 256\\3\times3, 256\\1\times1, 1024\end{bmatrix}\times6$	$\begin{bmatrix}1\times1, 256\\3\times3, 256\\1\times1, 1024\end{bmatrix}\times23$	$\begin{bmatrix}1\times1, 256\\3\times3, 256\\1\times1, 1024\end{bmatrix}\times36$
conv5_x	7×7	$\begin{bmatrix}3\times3, 512\\3\times3, 512\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 512\\3\times3, 512\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 512\\3\times3, 512\\1\times1, 2048\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 512\\3\times3, 512\\1\times1, 2048\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 512\\3\times3, 512\\1\times1, 2048\end{bmatrix}\times3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

3D Model Design

Since we need to put additional components on the car, we did 3D printing designs to combine the individual parts into one piece.



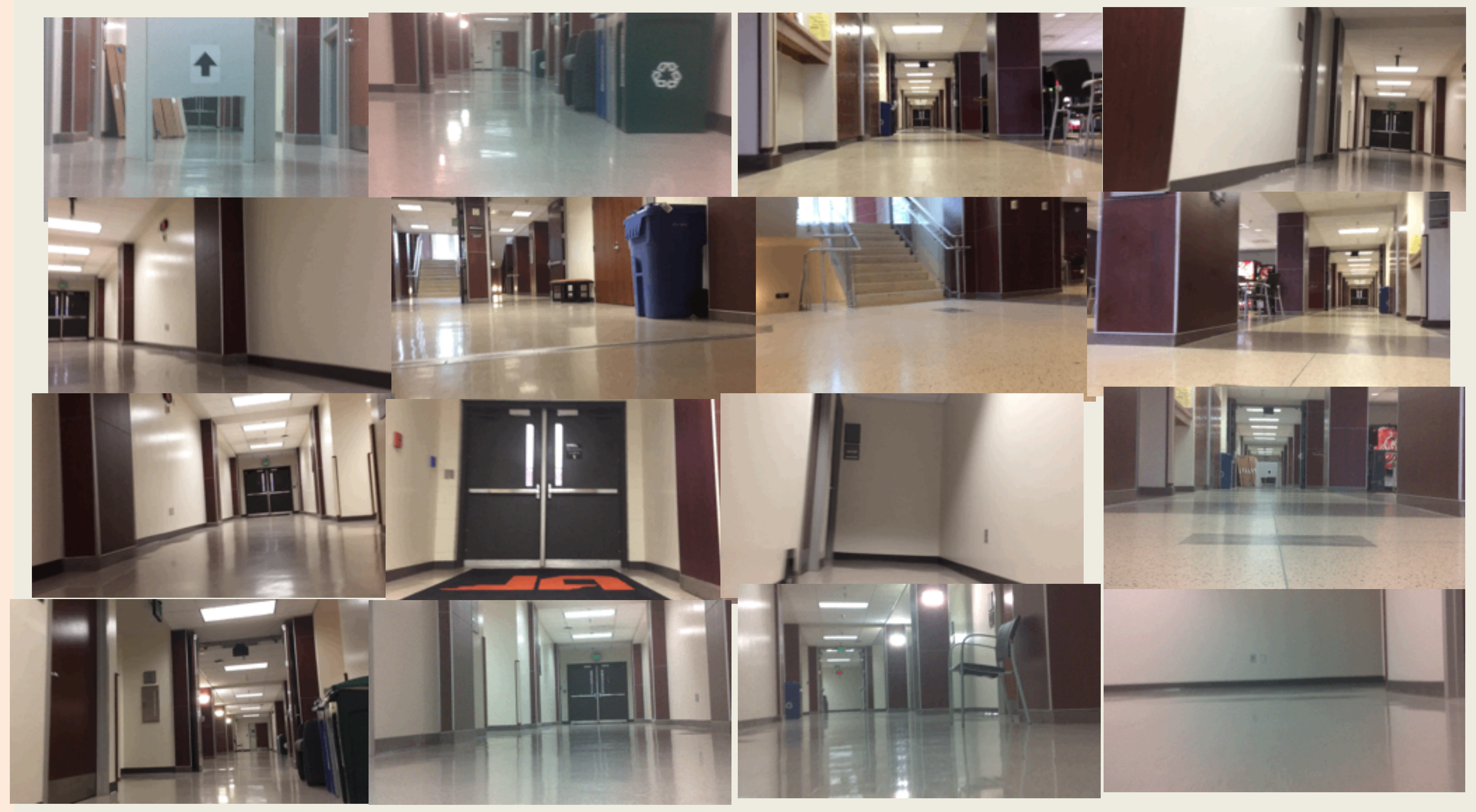
Testing

Limitation

- Memory of Jetson Nano - Data Size - Power Supply

Data Collection

To train the network, video footage from the camera was broken down into about 10 images per second, which were sorted into network output classes based on what was in the image and how the car should react. Some images included directive signs, while others just contained the natural features of the empty hallway. An estimated 6,000 images were used to train and test the network.



Future Work

1. Collect more data for model training
2. Fine-tune the data set and thoroughly train for increased accuracy
3. Develop and test more intricate MagmaDNN networks
4. Devise new ways to store, organize, and label data

Acknowledgements

- National Science Foundation
- The Joint Institute Computational Sciences
- Extreme Science and Engineering Discovery Environment (XSEDE)