Convolutional Neural Network – Drive Through Maze

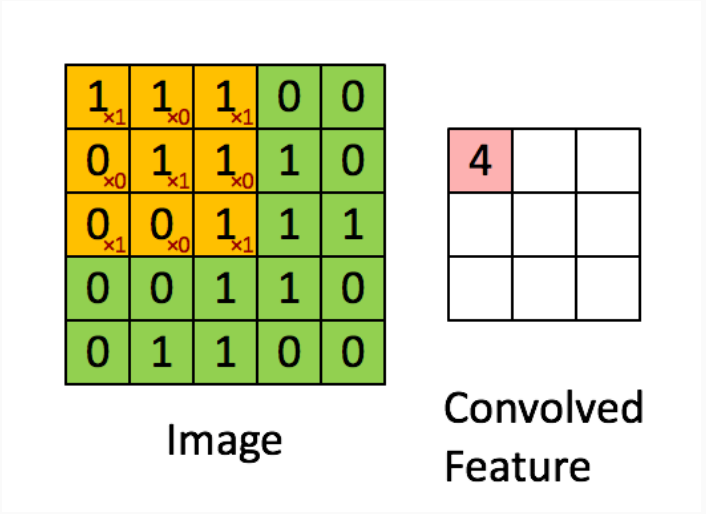
By: Rohan, Vignesh and Jesse

Introduction

- In this project, our goal is to create a robot which can drive through a maze which is built on ROS using convolutional neural network.
- We use convolutional neural network to classify images just like how we humans classify anything we see into categories that best describes the image. It takes an image as a matrix of image width and height with its pixel values.
- The convolutional neural network uses the images that is received from the robot to classify the maze into left, middle and right. We use this image classifications probabilities for the robot transverses to the center of the maze.

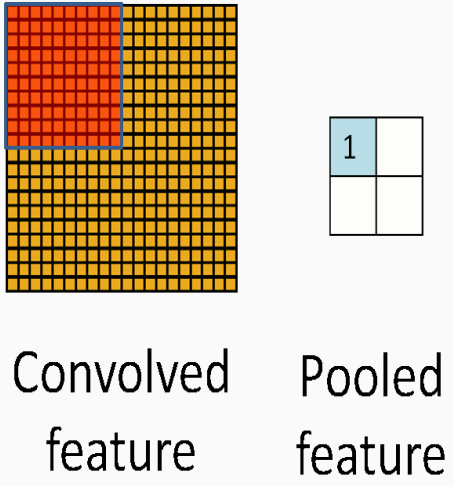
Convolution

- Natural images have the property of being ‘stationary’, meaning that the statistics of one part of the image are the same as any other part
- Features that we learn at one part of the image can also be applied to other parts of the image
- Feature extraction is done by taking small patches from the larger picture
- Use activation function on the patch extracted to generate Convoluted feature

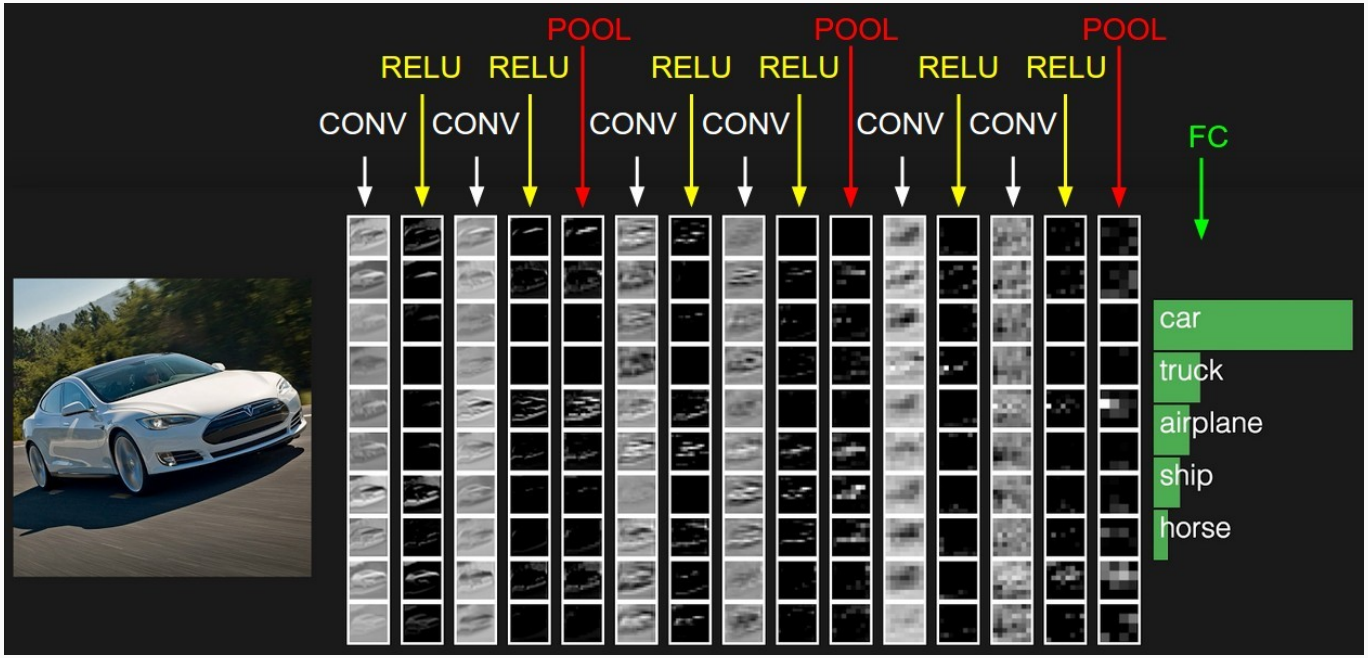


Pooling

- After obtaining our convolved features as described earlier, we decide the size of the region, say m×n to pool our convolved features over
- We divide our convolved features into disjoint m×n regions, and take the mean (or maximum) feature activation over these regions to obtain the pooled convolved features
- These pooled features can then be used for classification



Convolution Neural Network



ResNet

- Deep networks are hard to train because of the vanishing gradient problem—as the gradient is back-propagated to earlier layers, repeated multiplication may make the gradient infinitely small. As a result, as the network goes deeper, its performance gets saturated or even starts degrading rapidly
- The core idea of ResNet is introducing of “identity shortcut connection” that skips one or more layers
- Traditional neural nets will learn directly from the output whereas ResNet models the layers to learn the residual of input and output of subnetworks
- During backpropagation ResNet can choose to ignore the gradient of some subnetworks and just forward the gradient from higher layers to lower layers without any modification

Experiment Setup Overview

Goal: Navigate Differential Drive Robot to center of the maze

Maze Generation

Testing and Training Data

Building Fastai Model

ROS + Python

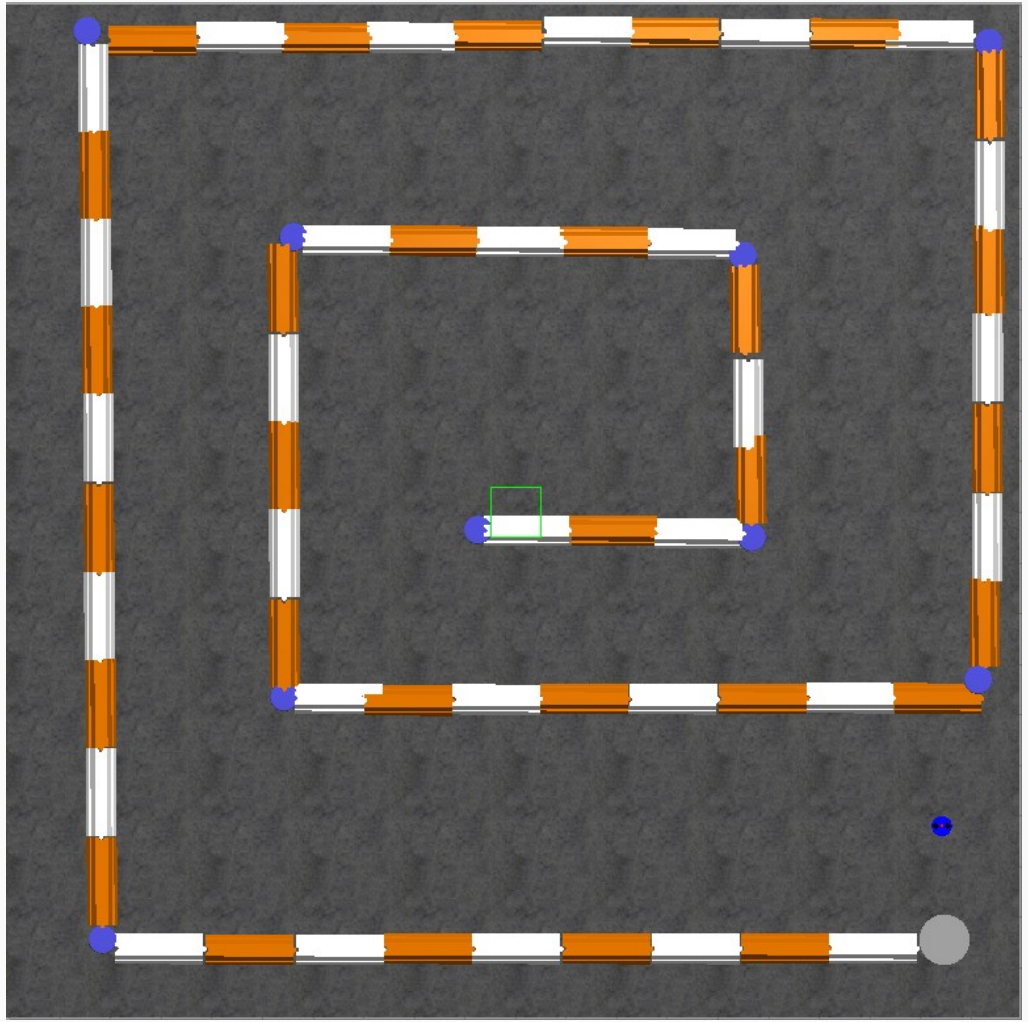
Maze Generation

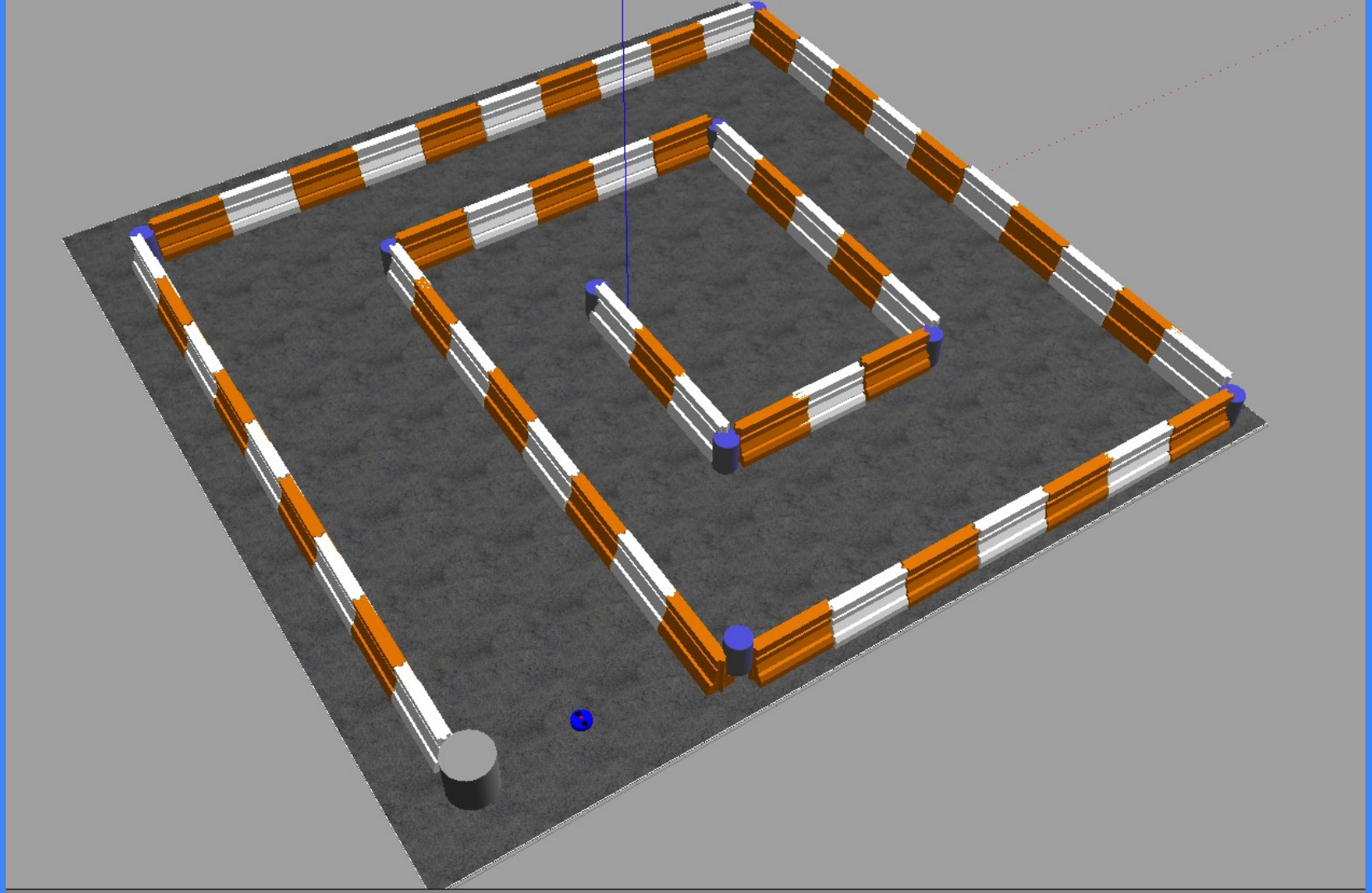
In order to create our various testing and training data, a gazebo world was generated.

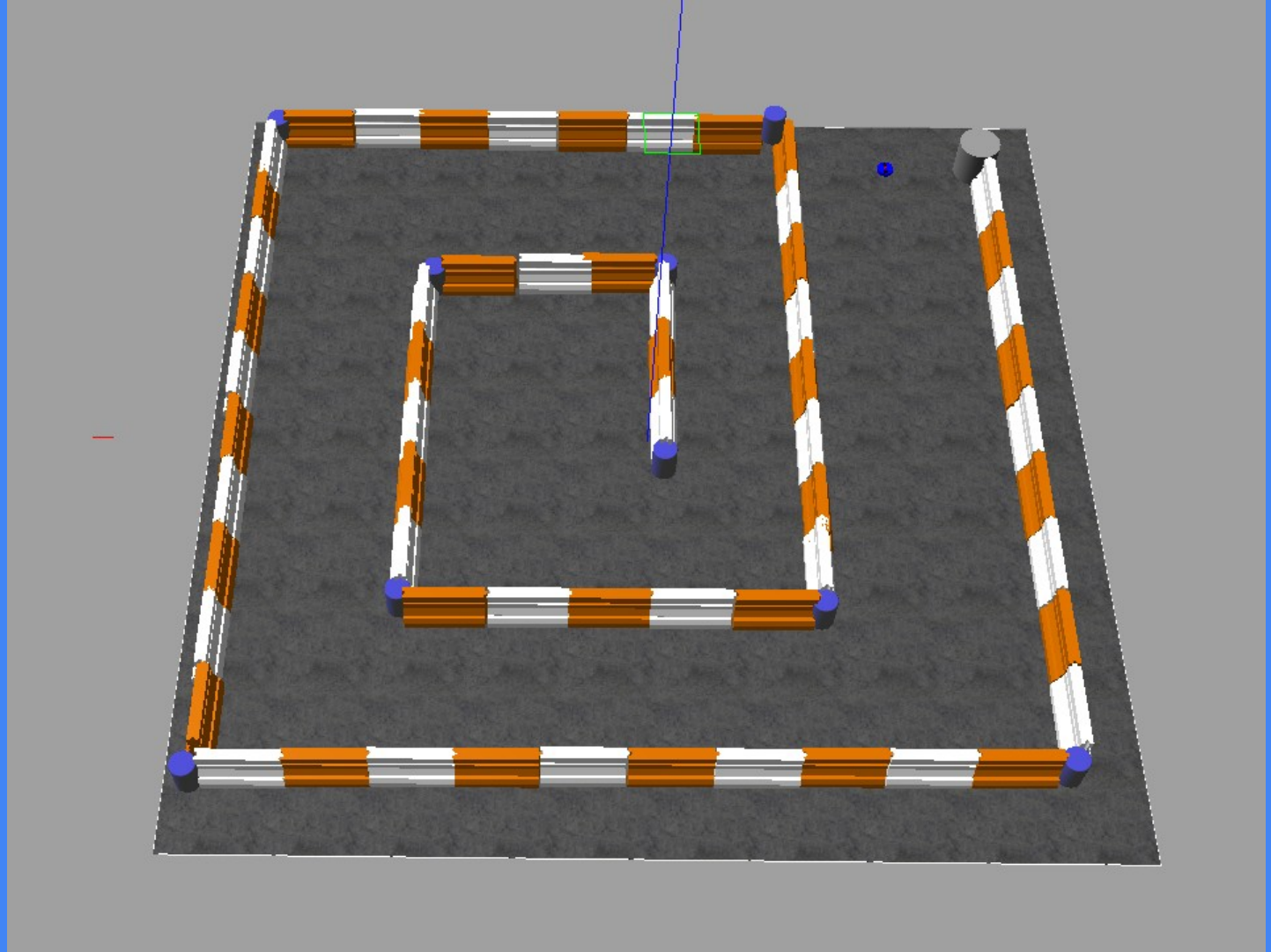
This world is a square-shaped maze that follows a clockwise path towards the center.

This maze was constructed using three objects: orange barrier, white barrier, and blue cylinders for corners.

Simple, but can showcase robot movement.







Testing and Training Data

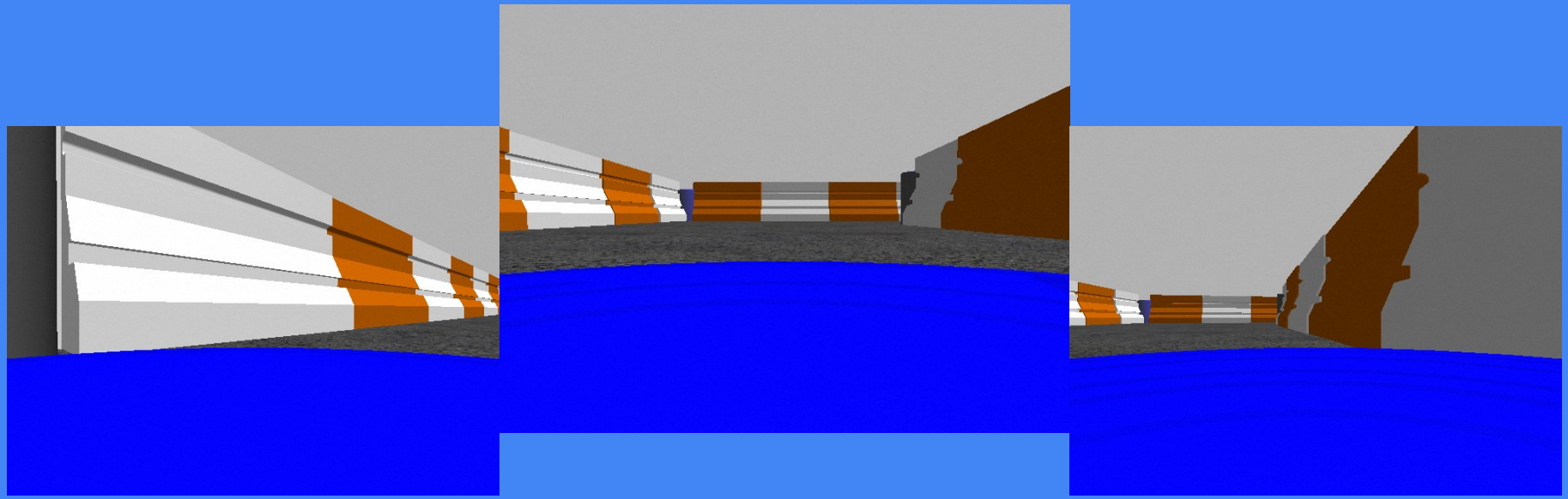
The generated map in gazebo was then used to generate testing and training data.

Used a rosnode that took user input from the keyboard, allowing the inputs: ‘w’, ‘a’, ‘s’, ‘d’ in order to control the robot.

Traversing the maze was done on the following paths: the left wall, center road, and the right wall.

As the differential drive robot travels down each source, using rqt image view, the camera data was then saved and properly labeled according to the direction the robot should drive to progress through the maze.

Training Samples



Building Fastai Model

Why use Fastai? Quick, very quick.

Time constraint, implement course knowledge.

Fastai has necessary tools to implement CNN.

Hopes of achieving 95-99% accuracy.

Experimented Values

Certain parameters affected the training accuracy.

- Learning Rate - Adjusting weights with respect to gradient loss
- Epochs - Number of iterations we run over training set.
- Architecture - Pretrained, efficient models. ResNet 34 and ResNet 50.

Goal: Finding optimal numbers for these values for accuracy goals.

First Experiment

Using a base learning Rate of 0.01

Running 1000 Epochs

Comparing ResNet 34 and ResNet 50.

Results?

Architecture	Learning Rate	Epoch's	Accuracy	Trained Images
resnet34	0.01	1000	90.63%	120
resnet50	0.01	1000	89.06%	120

Second Experiment

In order to see effect of adjusting the parameters, we said 90.63% was our best model at the time.

Learning Rate was then increased to 0.02.

Epoch's reduced to 500.

Comparing ResNet 34 and ResNet 50.

Results?

Architecture	Learning Rate	Epoch's	Accuracy	Trained Images
resnet34	0.02	500	90.62%	120
resnet50	0.02	500	88.28%	120

Third Experiment

Without seeing much accuracy change, started to assume 90.63% would be the best model.

Decided to adjust values slightly again.

Learning Rate was increased to 0.025.

Epoch's increased to 1000.

Comparing ResNet 34 and ResNet 50.

Results?

Architecture	Learning Rate	Epoch's	Accuracy	Trained Images
resnet34	0.025	1000	96.09%	120
resnet50	0.025	1000	91.40%	120

Final Experiment

With the large jump in accuracy, we were extremely pleased with the Third Experiment's model.

Could we make it better?

Learning Rate to 0.03.

Epoch's reduced to 200.

Results?

Architecture	Learning Rate	Epoch's	Accuracy	Trained Images
resnet34	0.03	200	96.31%	120
resnet50	0.03	200	87.50%	120

ROS and Python

ROS nodes were created in order to do the following:

- Subscribe to camera data, to save on Hard Drive.
- Wait for Fastai Script to output classification probabilities.
- Send movement commands to robot based on probabilities.

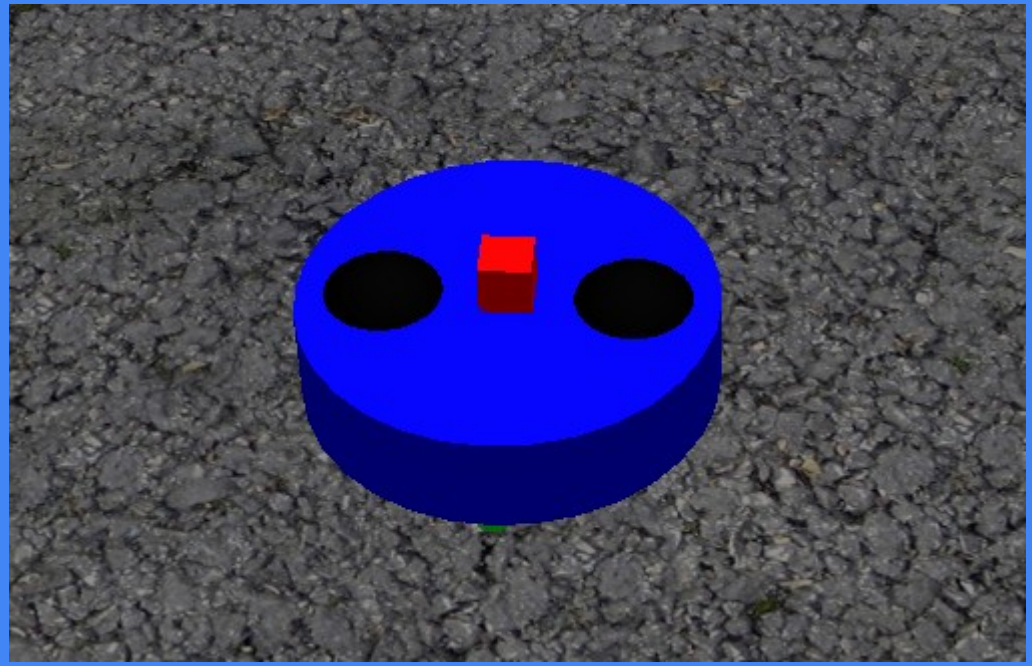
Python script to use Fastai model.

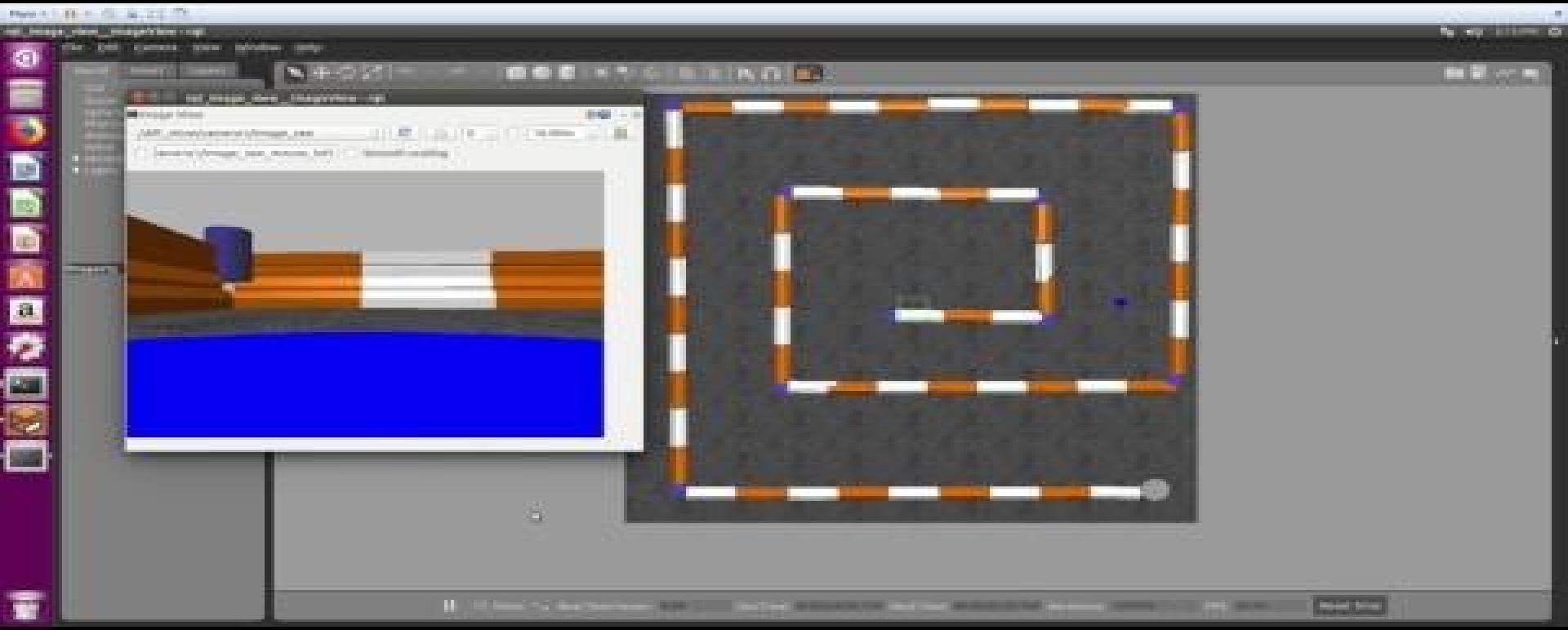
Ran in the anaconda3 environment on VM.

Finding optimal camera rate:

- Time the start time
- Time the end time for classifying image
- Convert seconds into camera hz

Experiment is ready to test.





Results of the project.

Using image classification is not optimal for robot control.

Very sensitive to camera update and robot speed.

Trial and Error tweaking to get it working.

Not 100% success on traversing the maze.

Future Work

Neural Networks to find values for movement parameters.

Use other image processing techniques:

- Semantic Segmentation
- Distance Calculation from Wall

Team Member Contribution

Jesse: Movement Script, Classifier Script, Fastai Model Training, Paper, and Presentation

Rohan: ROS Environment, Classifier Script, Fastai Model Training, Paper, and Presentation

Vignesh: Prepared Testing and Training Data, Paper, and Presentation

Questions?