ConvoBot

Overview

The Convolution Neural Network Robot (ConvoBot) project creates a platform to investigate several key aspects of training convolutional neural networks (CNN).

1. Access to significant amounts of labeled data.
2. Complementary networks to address model complexity.
3. Time required to train in real world environments.

Labeled Data

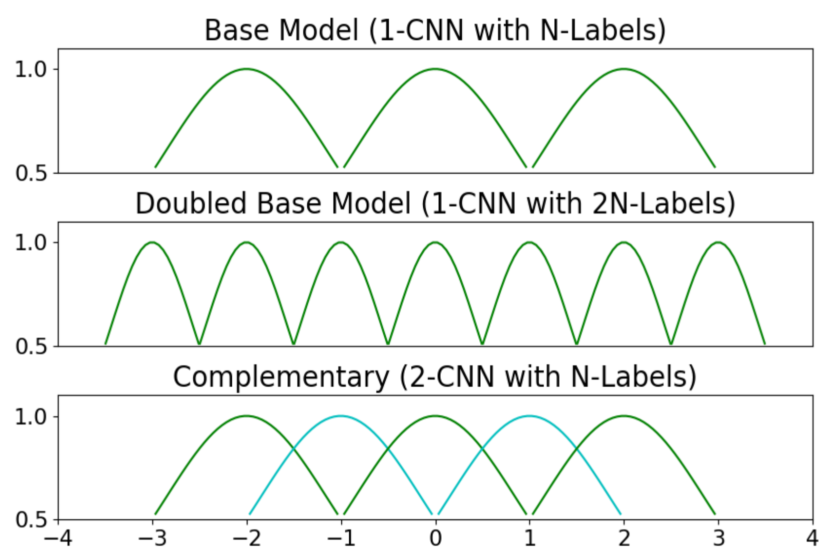
Training of neural networks requires significant amounts of labeled data. It is frequently quite costly to gather or label existing real world data. This project aims to address this problem by generating labeled image date through a virtual environment. The virtual environment is created in Blender, a leading 3D animation program. Through the capabilities of Blender the virtual environment can be made progressively more life like over time. This virtual environment is capable of generating 10^6 images overnight on a MacBook Pro.

Complementary Networks

Using CNN to predict the location of the robot based on images presents a number of problems. In this project predicting the location of the robot in 2D space will drive a very large number of labels. If we were to predict in 15º increments and 0.5" radial displacements over a 6" range, we would have

360/15 * 6/0.5 = 288 labels!

Any increase in accuracy comes at the cost of a more complex network and increased training time. More complex networks require more training time and more data. By building overlapping CNN we can continue to build simple networks at the cost of training more of them. Additional simple networks can be trained in parallel in the same time it takes to train one.



Separate networks can be trained to predict different components of the robots location in 2D space. One set can address the angular location and another the radial location in a radial coordinate system. Again each of these networks is composed of simple complementary networks.

With this model we can train the following networks.

Angular: 360/30 = 12 Labels x 2 CNN

Radial: 6 * 1 = 6 Labels x 2 CNN

12 * 6 * 4 = 288 Effective Labels

Real World Training Time

Training in the real world can be costly on many axis including time, safety, and funds to secure resources. Time may be reduced by using parallel environments, but that almost certainly comes at increases in the other costs and additional complexity. If we can minimize the amount of time required to train in the real world we can reduce these costs. There are two methods to explore in this area.

1. Capture images from the real world and increase the fidelity of the virtual world.
2. Partially train the CNN in the virtual world and then complete the training in the real world.

Project Scope

1. Virtual World Only
   a) Generate image dataset in Blender
   b) Sort images into labeled sets to train the Complementary CNN
   c) Build controller to select among CNN results to predict location
   d) Generate visual environment to see networks working together and predicting against additional simulated images.
2. Robot Image Capture (Stretch 1)
   a) Implement image capture on robot platform in closed environment that is very similar to the virtual world.
   b) Remotely position robot, capture image and predict location.
3. Increase Fidelity of Virtual World (Stretch 2)
   a) Robot captures representative background images.
   b) Images are integrated into Blender
   c) Leverage AWS compute resources to rapidly regenerate training images, train networks and deploy new CNN to controller.
   d) Demonstrate increased accuracy with improved training data fidelity.
4. Implement Supervisor (Ongoing hobby)
   a) Implement supervisor to measure robot location in 2D space.
   b) Implement controller to move to target based on CNN predicted location.
   c) Measure accuracy of predictions with feedback from Supervisor.