Project Ideas

**ROUGH DRAFT REV 01**

CSC485B Machine Learning in Robotics

Contents

[Overview 2](#_Toc129234519)

[Image Processing 2](#_Toc129234520)

[Optical Character Recognition 2](#_Toc129234521)

[Image Recoloring 2](#_Toc129234522)

[Camera Calibration 2](#_Toc129234523)

[Interscan 1500 Calibration: 1 dot and a robot 3](#_Toc129234524)

[Vision Calibration (Funky trig plus optical distortion) 3](#_Toc129234525)

[Converting several odd-angle measurements to real-world data 3](#_Toc129234526)

[Robot Calibration 3](#_Toc129234527)

[Cyline: calibrating motion speed on cylindrical objects using a standard robot 3](#_Toc129234528)

[Robot Vision to Motion Calibration 3](#_Toc129234529)

[BrightonBeachLab: a 6DOF calibration from 2 cameras to robot movements 3](#_Toc129234530)

[Physics 3](#_Toc129234531)

[Modeling Friction 3](#_Toc129234532)

[Fluid Dispensing 3](#_Toc129234533)

[Rocket to the Moon 3](#_Toc129234534)

[Agent-based Modeling, Cellular Automata 4](#_Toc129234535)

[Classification Ideas 4](#_Toc129234536)

[Checkweighing 4](#_Toc129234537)

[Good Part Bad Part 4](#_Toc129234538)

[Good vs. Bad Art 4](#_Toc129234539)

[Maze Running 4](#_Toc129234540)

[Clustering Ideas 5](#_Toc129234541)

[Measure Data on Operator Performance 5](#_Toc129234542)

[Classification or Clustering 5](#_Toc129234543)

[Word Analysis 5](#_Toc129234544)

# Overview

This document is very much a work-in-progress collecting ML project ideas for our class.

# Image Processing

## Optical Character Recognition

LEocr and next steps.

## Image Recoloring

From discussion with Michael Walters, 2/7/2023

So, if we take an arbitrary 24bit RGB color image and replace each pixel with (R+G+B)/3, we get a great monochrome image. Some folks use slightly different weights on the RGB since people are more sensitive to green, but it basically turns out the same.

In this operation, you've lost 2/3 of the bits in the image... so an arbitrary color image which has been thus converted cannot be converted back...

unless there is other information.

For example, if you gave me a black and white image of the US flag, I could colorize it, although if it had been bleached before you took the picture, I'd guess wrong!

So, what we'd need to do is create color images of some set of objects that have some common characteristics... say, images of triangles, squares, and circles, where each shape is a different color. If we trained an image analyzer to recognize the shapes and pair them with their colors, then we could go from monochrome image to color image. We could even go from more reduced features, like a list of shape types, sizes, and orientations, and recreate the color image from that. This combines image analysis, dimensionality reduction, and image reconstruction into a fun project.

# Camera Calibration

Lenses distort in funky ways and things move.

## Interscan 1500 Calibration: 1 dot and a robot

## Vision Calibration (Funky trig plus optical distortion)

## Converting several odd-angle measurements to real-world data

# Robot Calibration

## Cyline: calibrating motion speed on cylindrical objects using a standard robot

## Robot Vision to Motion Calibration

## BrightonBeachLab: a 6DOF calibration from 2 cameras to robot movements

# Physics

## Modeling Friction

Friction to help with part handling, gripping, part rejection mechanisms….

Way mysterious and lots of independent variables

<https://www.20sim.com/webhelp/modeling_tutorial_friction_staticdynamicphenomena.php>

## Fluid Dispensing

Fluid flow rate changes with temperature, time, and voltage or current on a dispensing plunger

## Rocket to the Moon

From discussion with Michael Walters, 2/7/2023

Mechanical and Aerospace Engineer here, I love rockets. One of my favorite simple demonstrations illustrating the value of calculus is deriving the Tsiolkovsky rocket equation from first principles. (ChatGPT does a great job except every time it seems to be dead wrong about the math and sometimes doesn't even conjure up the correct equation.)

Anyway, what I'm lecturing on today and into next week is how we can use ML for things that data scientists don't use ML for. Like robotics and physics! So I love this. The three main pillars of ML are:

* Regression: Come up with Y = F(X) where Y and X are matrices and F could be anything. We do it by building functions or by comparing to all our training instances.
* Classification: Define X to be a member of 1 or more known classes of things (by building a model or by comparing to all the training instances)
* Dimensionality Reduction: (Like the coloring problem above...)
* Reinforcement Learning: If none of the above work, just start "doing something" to X and see if you can nudge it toward Y, and then improve your choice of what "do something" is. X could be a state vector and we are a controller, or X could just be data or an image that we are trying to transform in some way.

So, when I think about your rocket to the moon problem, it's like:

1. How many stages?
2. What fuel is used in each stage? JP1, H2 O2, SRBs, etc.
3. What engines... weight vs. specific impulse, maybe even cost?
4. What mass to deliver to LEO vs. Lunar Orbit vs. return to Earth?
5. Mission profile... launch east or west? Orbit Earth first? What kind of trans-lunar trajectory (Orion just did a nonsense thing since it had no significant fuel up in the last stage). What kind of trans-Earth trajectory? How to re-enter? Land on water, land, or vertical like SpaceX first stages??

We could build out the physics equations easily but then this becomes an optimization problem.

* What are we optimizing? Cost? Weight? Safety? Some mix of all three subject to specific requirements (sounds like dynamic programming…)

How would we normally solve this?

1. Monte Carlo?
2. Dynamic Programming?
3. Sequentially optimize each mission step and then hope to optimize as a system later…

Then, we come up with a novel application of the ML engines to do a better job. Maybe:

1. Use Monte Carlo/Random selection to build a series of mission profiles and their value as training data, then train on those to find some sort of model, then try to numerically differentiate or gradient descent that model to find the lowest cost??
2. Try Reinforcement Learning and adjust one parameter at a time and sort of learn how to tune ourselves to a best-case without getting stuck in some local minimum?

# Agent-based Modeling, Cellular Automata

Complex behaviors from simple rules that defy direct mathematical description (NetLogo Ants)

# Classification Ideas

## Checkweighing

Say a parcel is supposed to contain so many of item1, so many of item2, and so many of item3. We don’t know the weight of each item but we have training data with total weight and quantities expected for each number.

Can we train a classifier to return GOOD or BAD on a large set of synthesized data?

## Good Part Bad Part

Based on weight, average color, area

## Good vs. Bad Art

Build a database of sketches and categorize as good or bad. Feature extract and go from there.

## Maze Running

An algorithm generates a set of outcomes… can we classify the algorithm as having a certain outcome and then select that algorithm when we want that outcome?

# Clustering Ideas

## Measure Data on Operator Performance

Try to figure out what parts came from what operator, given N operators.

# Classification or Clustering

## Word Analysis

Say we are getting strings of characters that represent actions in a control system. We don’t know what they mean, but we can train on what we’re seeing and then look for anomalies. Can be done with supervised data or as straight cluster/anomaly.