5/8/24 B & A CS capstone

~~Model~~ Prior - manually labeled 1700 images (~6 circles/image)

1728 photos of wells < 20% 1300 for training

trained FR CNN faster region based NN

Resnet 50 : pretrained model on faces, buses, ...

(type of FRCNN) fine-tune to wells

(a lot of layers)

at talk, trained CNN Convol. NN (3-4 layers)

25-40 min to train on Resnet → weights

Test 41% precise Resnet

try to find centers of each circle - Resnet

not used now

CNN - could say it was a circle classifier. Couldn't say where circles were

5 min / picture

Heat mappings

5 sec to take big image, finds circles grid, spot pH sensor

w/ straight lines

1700 images
vary orientation
lighting
background (black, wood, white, double stack)
digitally
Edit in more angles

Full image ___ x ___
Cropped it down 777 x 936

CVAT to label circles on images (on web)

41% prune — are boxes drawn same way
box surrounds each circle

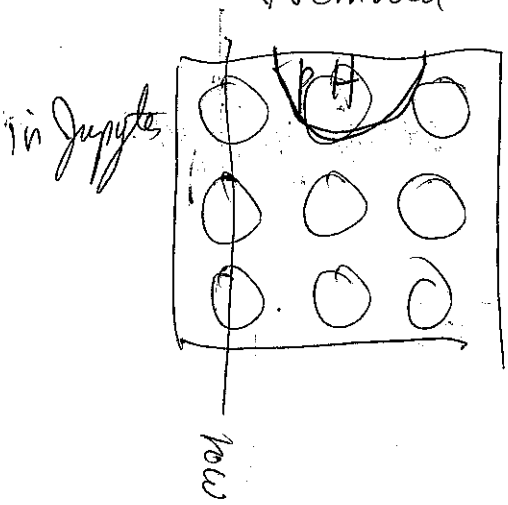
To move robot to denied circle, faulty logic

Jupyter Legola Demo
added section Testing OpenCV

final_config.yaml — Brendan's name

first all — does what manual mode does

Calc of adjustment — faulty
ImageAssessment.py.
Formula is wrong



Related pixels to Legolas coords
picture with a spot — pixel coords
moved Leg. —
Manual.py shows x, y position
Made ratio of Leg coords/pixel — made incorrect
Could be logic is faulty

~~Could be logic~~

Pi 7A 192.168.1.11 switchedback to old ip (see PPT how to do this)

7B 192.168.1.12

username: pruenbaf
password: raspberry

where problem is

Image Assessment.py
verify x pos & verify y pos defs

Core.py
line ~~336~~ 363

5/20/24 Audrey

https://loyola.zoom.us/j/6945057758

sudo apt-cache show python3-opencv

single snapshot

core.py

load model

def move_to_cell line 321

line 327 img_bytes – captures the image

329 array

330 Image decode from np.array to PIL Image

334 processed_image ties to ImageAssessment.py

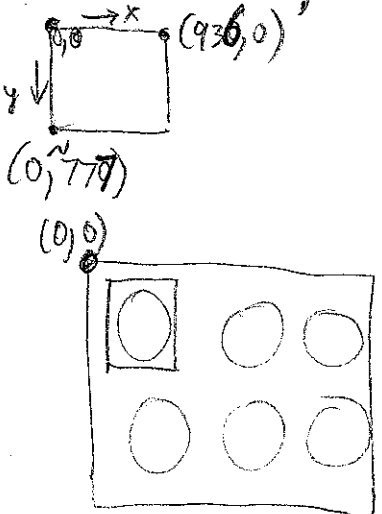
crops it → image

data_transform in load_model

reads in trained model

make_prediction

coords, make_grid_line



[[row
[col1] [col2] [col3]]

Bounding box for each circle

minx maxx miny maxy

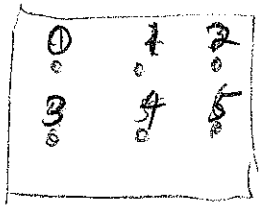
r = 140 pixels

know center of desired circle

~~ImageAss.py~~

core.py — Convert to coords
xs, ys : list of centers of circles
coordinates has list (x, y) for centers

ImageAss — make_grid_lines
get neighbors
Make list of neighbors



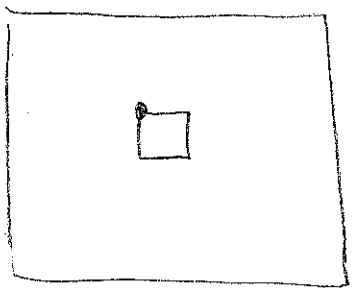
hor & vert lines — best fit. Won't go through centers necessarily

core — have guestimate of center of pft sensor
(394, 663) values are eyeballed
pft-pixels
could be wrong

pictures in report are flipped

Get nearest line to circle closest to pft

Motor units
pixel units



heat sink
took picture
Find coords of upper left corner
Move motors

$.55 = \frac{\text{motor units}}{\text{pix units}}$

Calib was done
movably 20 motor units

problem? verify_x_y_pos

returns offset from pt sensor
maybe math is incorrect