**Woz Project Summary**

<https://github.com/bobhigs/woz-vision>

* Path recognition
  + On github 🡪 src 🡪 path\_detection.cpp
    - Dax and Devon worked on this a while ago, but it blacks out the path in the video that is read into it
* People detection
  + On github 🡪 src 🡪 people\_detect.cpp
    - Kelsey and Joanna have worked with this a lot
    - It takes in a webcam video, and will draw boxes using a HOG detector (OpenCV software built in element) to detect the people in the images
  + On github 🡪 src 🡪 webcamTracker.cpp
    - Ryan worked with this 🡪 what has been run on weekend events
      * Uses people\_detect.cpp
    - Follows a main point along the screen so that if the peopledetect all of a sudden doesn’t pick up a person in one frame, the person will still be tracked
* Resizing images
  + On github 🡪 src 🡪 resize.cpp
    - Resizes the images or video all at once so that the hog detector works more efficiently
* Training different objects using cascades
  + Cascades are in trainingfiles folder on github
  + Step by step directions on how to train are on the second page
* Installing OpenCV
  + <http://www.learnopencv.com/install-opencv3-on-windows/>
    - Shows you how to install openCV using visual studio
  + <https://www.learnopencv.com/install-opencv3-on-ubuntu/>
    - Shows you how to install OpenCV using ubuntu
* The jetson board (found in closet thing) has all of this set up already
  + All of the materials to set it up are in the box, and then grab a monitor, keyboard, and mouse

Important opencv commands

Resize images – figure out a way to resize images/ scale down images, maybe write a program

Program to generalize running the code

<https://abhishek4273.com/2014/03/16/traincascade-and-car-detection-using-opencv/>

new object:

- pos folder with images of the object

- neg folder with images that do not include the object anywhere

- data folder to hold cascades

- annotations folder to hold the annotations

(this will do all the pictures at once)

- opencv\_annotation --annotations=./bench.info --images=./pos/

opencv\_annotation --annotations=./bench9.txt --images=./pos/bench9.JPG

- navigate to folder that holds pos and neg folders

- find neg -iname ”\*.JPG” > bg.txt

//find pos -iname ”\*.JPG” > bench.info

For all pos files

in main folder

cat ./annotations2/bench33.txt >> ./main-annotations.txt

- create samples -num = number of pos samples

opencv\_createsamples -info main-annotations.txt -num 36 -w 60 -h 80 -vec bench.vec

- opencv\_createsamples -info bench.info -num 48 -w 60 -h 80 -vec bench.vec

Train

start with a smaller number of stages and then gradually increase them to get multiple cascade files

Rename each cascade file

Make a data file for each cascade file

opencv\_traincascade -data data -vec bench.vec -bg bg.txt -numStages 10 -nsplits 2 -minhitrate 0.999 -maxfalsealarm 0.5 -numPos 44 -numNeg 49 -w 60 -h 80

When running the code, you need to train multiple times with different stages or different arguments

Can’t figure out/remember how to create an executable

Test with different cascade files, starting with original cascade file

move cascade.xml file out of data directory for easier access

executable imagepath cascade.xml casc2.xml casc3.xml …

./bench ./BenchData/testbenchs/benchStatue.MOV ./BenchData/cascade.xml ./BenchData/casc2.xml

./bench BenchData/testbench1.JPG ./BenchData/cascade.xml ./BenchData/casc2.xml

./bench ./BenchData/testbenchs/bench1.JPG ./BenchData/cascade.xml ./BenchData/casc2.xml

./car\_detect ./BenchData/testbench1.JPG ./BenchData/cascade.xml ./BenchData/casc2.xml

#path to new annotations file, path to image

opencv\_annotation --annotations=./annotations/IMG\_3320.txt --images=./pos/IMG\_3320.JPG

opencv\_traincascade -data data3 -vec phone.vec -bg bg.txt -numStages 20 -nsplits 2 -minhitrate 0.999 -maxfalsealarm 0.5 -numPos 44 -numNeg 49 -w 60 -h 80

opencv\_createsamples -info test.info -num 48 -w 60 -h 80 -vec phone.vec

#1 phone

./phones ./Phone/phone2.JPG ./Phone/cascade.xml ./Phone/casc2.xml ./Phone/casc3.xml

./phones ./Phone/phone3.JPG ./Phone/cascade.xml ./Phone/casc2.xml ./Phone/casc3.xml

./phones ./Phone/phone3.JPG ./Phone/cascade.xml ./Phone/casc2.xml ./Phone/casc3.xml ./Phone/casc4.xml

#works

./phones ./Phone/phone2.JPG ./Phone/cascade.xml ./Phone/casc2.xml ./Phone/casc3.xml ./Phone/casc4.xml

./phones ./Phone/phone2.JPG ./Phone/cascade.xml ./Phone/casc2.xml ./Phone/casc4.xml

#2 phones

./phones ./Phone/phone4.JPG ./Phone/cascade.xml ./Phone/casc2.xml ./Phone/casc3.xml