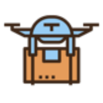
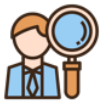
LAB 1: IMAGE RECOGNITION

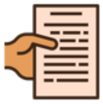
Due: 11:55pm on Tuesday, September 1st

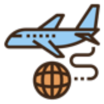
Individual assignment

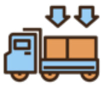
The objective of the first lab is to use image processing and machine learning tools to correctly classify images. Specifically, you will develop an algorithm that will enable your Cozmo to distinguish the following seven symbols from each other, and from other random scenes the robot might see. Additionally, you will use RANSAC to fit a line to the arena wall.

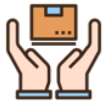

drone

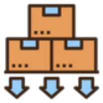

inspection


order


plane

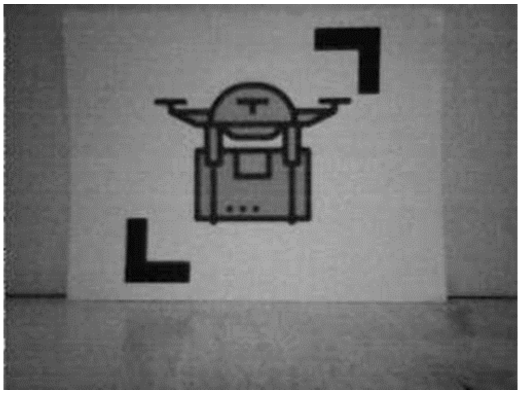

truck


hands


place

Your algorithm will be given images taken by the Cozmo as input. Your algorithm must then determine which of the above symbols if any is in the image. You will also implement line fitting using RANSAC in order to detect arena edge lines.

Long-term purpose: The classifier algorithm that you develop in this lab will be used on the Cozmo/Vector robot for future labs for behavior control and localization of the robot.



Images for classification: We have provided an image dataset containing grayscale images taken by the robot. The dataset contains 8 types of images: one for each of the above symbols, plus the type "none" for pictures not containing a symbol (just a picture of the empty arena, cube or wall). The dataset is further split into separate testing and training datasets.

Images for line fitting: We have provided an image dataset containing grayscale images of the arena wall for testing with RANSAC.

Software Installation: You will be developing your algorithm in Python 3. You will need scikit-image and scikit-learn libraries for this lab. Installation links and instructions here:

- Python3: https://www.python.org/downloads/
- Scikit-image: http://scikit-image.org/docs/dev/install.html
- Scikit-learn: http://scikit-learn.org/stable/install.html

Also install numpy if you haven't already. This is a very helpful page with instructions on how to install packages and manage virtual environments.

Reading: Read through the "Getting Started" and "A crash course on Numpy for images" section of the scikit-image user guide. Also read "an introduction to machine learning with scikit-learn" of scikit-learn user guide. You may use any function from either of those libraries for your task.

Lab: We have provided the following in `imgrecognition.zip`.:

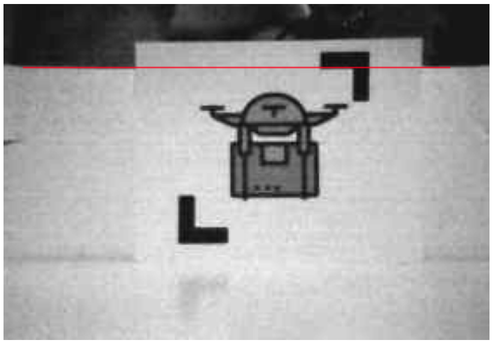
`imgrecognition/`

`train/` - directory containing images to be used for training your classifier. The correct label of each image is listed in its file name.

`test/` - directory containing images to be used for testing your classifier. The correct label of each image is listed in its file name.

`wall/` - directory containing images to be used for line fitting using RANSAC.

`imgclassification.py` – this is the main file where you will enter your solution. The code already contains functions for reading in the images from the train/, test/ and wall/ directories and for formatting the data into an array. Add code to convert pixel values of the images to features that can be used for training a classifier. Then train a classifier of your choice and test its predictive performance against the test set. You will need to fill in three functions in the `ImageClassifier` class, namely, `extract_image_features`, `train_classifier`, and `predict_labels`.



For `line_fitting` you will need to fill in code to use RANSAC to estimate the best line (ex. red line on the left image) that fits on the wall of the arena. For reliable execution, this will run on wall images, which are images with the arena wall without the symbol cards. As mentioned in the lecture, you should run RANSAC on edges extracted from an edge detector in order to reduce noise. You're free to use any edge detector you desire, such as Canny edge detector.

Do not modify the header and the return type for each function as that will interfere with the autograder and affect your grade. You may add helper functions as needed but make sure your code is self-contained because this is the only file you will be submitting.

Evaluation: We will evaluate the performance of your algorithm on 186 images from a withheld test set containing images that were taken under the same conditions as the ones you have been provided, but that were not included in your dataset. The classification training set will remain the same. Your grade will be determined as the percentage of the images classified correctly.

Wall images within the withheld test set will be used for line fitting evaluation. Note that in this assignment, all tested wall images will contain a visible line indicating the boundary between the arena wall and empty space.

Submission: The assignment is due by 11:55 pm on Tuesday, September 1st. You must not rename the file from `imgclassification.py` and submit it on Gradescope. Do not zip the file. Do not upload any additional files. Make sure your code is contained entirely within this file. If you relied significantly on any external resources to complete the lab, please reference these in the file comments. This assignment is to be completed individually.

Grading Rubric:

Image Recognition *70 pts* total based on percentage of images classified correctly:

$$\frac{correctly\ classified\ images}{total\ images} \times 70$$

Line Fitting *30 pts* based on percentage of lines identified correctly:

$$\frac{correctly\ identified\ lines}{total\ images} \times 30$$