

LAB 1: IMAGE RECOGNITION

Due: Tuesday, September 4th 5pm

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 Cozmo to distinguish the following seven symbols from each other, and from other random scenes the robot might see.



drone



inspection



order



plane



truck



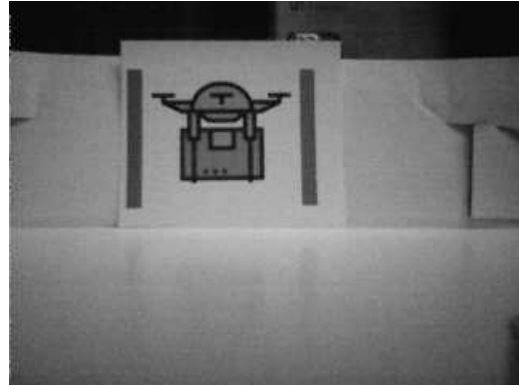
hands



place

As input, your algorithm will be given images previously taken by the robot, as shown on the right. You must determine whether one of the above symbols is in the image, and if so, which one.

Long-term purpose: The code we develop in Lab 1 will eventually be used with the Cozmo robot, enabling it to identify the above symbols in its environment. This capability will be leveraged in Labs 2, 5 and 6 for behavior control and localization.



Images: 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). Each of the symbols is flanked by vertical bars on the left and right, as can be seen in the image above. These can be used to locate the symbol in an image if needed.

Software Installation: For working with Cozmo you will need Python 3, and for this lab you will need scikit-image and scikit-learn libraries. Follow the installation instructions here:

- Python3: <https://www.python.org/downloads/>
- Also install numpy and matplotlib by typing `pip3 install numpy` and `pip3 install matplotlib`
- Scikit-image: <http://scikit-image.org/docs/dev/install.html>
- Scikit-learn: <http://scikit-learn.org/stable/install.html>

(Note that Lab1 can be completed with Python 2.7, but you will need Python 3 starting with Lab2)

Reading: Read through the “Getting Started” and “A crash course on Numpy for images” section

of the [scikit-image user guide](#). Then refer to the lecture notes 8/23 for additional links to useful pages in scikit-image documentation.

Also read “[an introduction to machine learning with scikit-learn](#)” of [scikit-learn user guide](#). Refer to the lecture notes from 8/28 for additional links to useful scikit-learn functionality.

For both scikit-image and scikit-learn, you are not restricted to only using the methods I mentioned in class, you can use the entire functionality of both libraries.

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

`imgrecognition/`

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

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

`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/` and `test/` 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`. To have your submission correctly graded, please do not modify the header and the return type for each function. 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 50 images. 25 images will come 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 other 25 images will be a random selection of the images you have already seen in the test set. The training set will remain the same. Your grade will be determined as the percentage of the images classified correctly.

Submission: By 5pm on Tuesday September 4th submit your code on Canvas, having modified the filename to be `yourlastname_yourfirstname.py`. Make sure your code is entirely contained 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.