# LAB 1: IMAGE RECOGNITION

## Due: 5pm on Tuesday, September 3rd

### 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.



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.



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

**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). The dataset is further split into separate testing and training datasets.
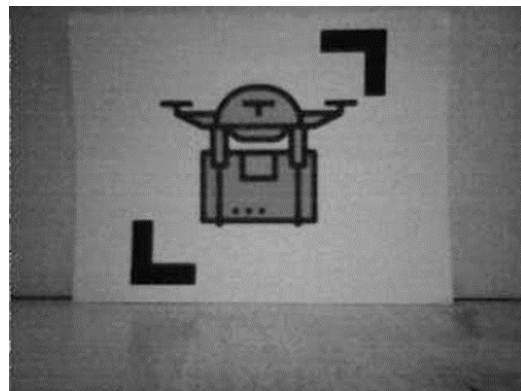
**Software Installation:** You will be developing your algorithm in Python 3. You will need scikit-image and scikit-learn libraries for this lab. Installation links and instructions here:

- 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 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`. 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 training set will remain the same. Your grade will be determined as the percentage of the images classified correctly.

**Submission:** The assignment is due by 5pm on Tuesday, September 3$^{rd}$. You must rename `imgclassification.py` to `yourlastname_yourfirstname.py` and submit it on Canvas. 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.