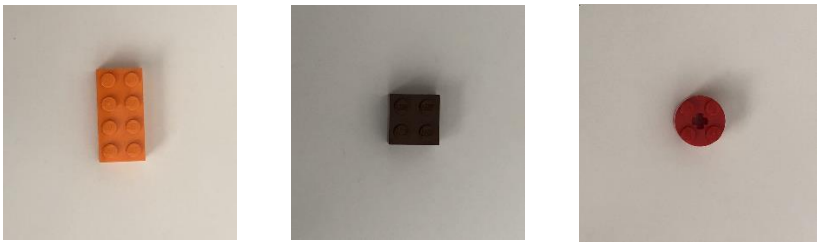
ENGR 418

Project – Stage 1

Sorting using raw Images

You have been hired by a company to develop machine learning algorithms for a sorting facility. The requirement is that the sorting device takes images of items on a conveyor belt, and then uses and machine learning algorithm to classify the items into classes. Then, the items get routed through different routes on the conveyor belt depending on their class.

For the sake of this project, you are given RGB images, and your focus will be on developing the classification algorithms. Also, for simplicity, it is assumed that items are Lego pieces of three different types with the following shapes (top view): Rectangles (2x4), squares (2x2), and circles (2x2). Examples are shown below.



Your algorithm should be able to classify these three classes with an acceptable level of accuracy. **It must use the raw image (grayscale conversion and scaling/cropping are acceptable) as an input to a single neuron classifier with < 4097 weights (trainable parameters)**. You are given two datasets to achieve this goal, each containing multiple images of each class. Use the dataset in the folder 'training' for training, and the one in the folder 'testing' for testing. Do not change the names of the folders or files.

Your tasks are as follows:

1. Develop your code in Python and submit it through Canvas.
    a. Your submission must be a python notebook named 'Group_x_stage_1.ipynb', where x is your group number.
    b. Each cell in your python notebook must have a clear header comment box indicating what this cell does.
    c. The first cell in your notebook must contain the group number, group member names and student ID numbers.
    d. Training and testing must be done in separate cells.
    e. Your code must be well commented.
    f. Your code must be able to read all files in the training and testing folders, without any assumption on the number of files. During marking, your code may be run on folders with a different number of files.

g. Your training cell must display the performance of the algorithm on the training set, displayed in the form of a confusion matrix as well as accuracy (no. of correctly classified samples divided by the total no. of samples)
h. Your testing cell must display the performance of the algorithm on the testing set, displayed in the form of a confusion matrix as well as accuracy (no. of correctly classified samples divided by the total no. of samples)
    - Include your testing code in a callable function. The callable function must be called "test_function(path,arg1,arg2,…)" where a path will be passed (like "C:/Folder1/Folder2/") in addition to any other inputs that are necessary for your implementation (you can change the names arg1, arg2, etc.. This path will be replaced with the path of a testing set that will be used to test your classifier for grading.
i. When your code is ready, test your solution and study its performance on both the training and testing sets as a function of the number of weights (trainable parameters < 4097)

2. Write a report explaining your work, and submit it through Canvas.
    a. Submit your report in PDF format.
    b. The report template is given on Canvas. Use the same template.
    c. The maximum page length is 5 pages (including references) with 1 inch margins, and the minimum font size is 12pt, and you must use the Time New Roman font.
    d. Do not include code in your report (code snippets are acceptable). Instead, you can explain your code using a flowchart or a pseudocode if needed.
    e. Your report must explain clearly how your algorithm work, and must comment clearly on its performance for different numbers of weights (trainable parameters). Be sure to explain why your algorithm works and where it does not work why it does not.

All group members are expected to contribute to the projects. Failure to contribute to a project may lead to a deducted mark or a mark of zero on this stage of the project.

Any questions about the project must be directed to the instructor. The instructor will explain unclear issues and may amend this description to answer students' questions.

This stage of the project is worth 15% of your overall course grade, divided into 7% on the report, 7% on the code, and 1% on a project survey the will be posted on Canvas to assess each student's contribution to the project.