USING COMPUTER VISION TECHNIQUES TO COUNT LEGO PIECES.

These are two “sister” lab assignments. The first one involves “old school” computer vision techniques to segment and classify LEGO pieces. However, the images’ background is relatively simple. The second lab assignment provides the students with a larger public dataset with LEGO pieces with more challenging backgrounds. They need to use deep learning techniques to count the LEGO pieces.

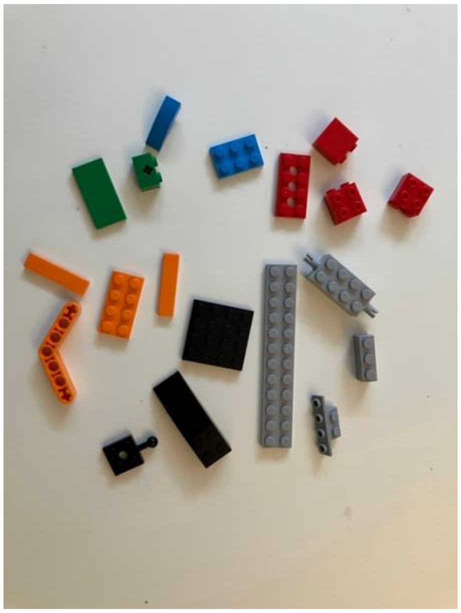
The first assignment (Lab 2 – LEGO time!) is given to students during week four of a sixteen-week course. The second assignment (Lab 4 – LEGO time. Again.) is given on week eight.

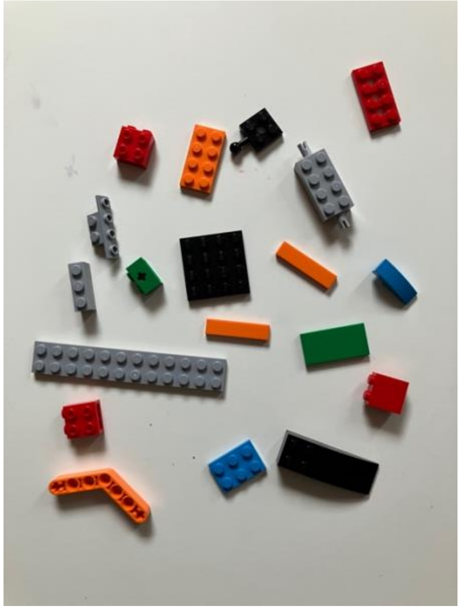
LAB 2 - LEGO TIME!

THE PROBLEM

We are going to try to solve one of the greatest challenges of playing with LEGO sets. No, I don't mean we are going to eliminate the pain caused by stepping on LEGO pieces. Instead, what I have in mind is to develop an automatic way of counting and classifying LEGO pieces.

I took a bunch of pictures of LEGO pieces. Here are two of these images. I have not enhanced the images in any way (I guess you can tell).





WHAT YOU ARE GOING TO DO

You are going to teach the computer how to count LEGO pieces and classify them by colour.

Download the images and **use three segmentation techniques to count and classify every LEGO piece.** Identify which of these segmentation techniques is more suitable for this project. Explain how you reached this conclusion. There are more than thirty images available. You do not need to use all of them but make sure you use enough to convince everyone that this process can (or can't) be automated.

You can pick any techniques that you want but here are some suggestions:

- One thresholding technique (either global or local thresholding) - the threshold must be found using an algorithm (not chosen by visual inspection!)
- One edge detection technique; for instance, Canny (no need to write the algorithm, OpenCV has done that for you... but do learn how to use it)
- I'd give the *rg chromaticity* technique a try since the previous two techniques highlighted here rely on grayscale images and I have the hypothesis that colour makes segmentation easier since the LEGO pieces have a variety of colours.

Before you apply these algorithms, check if some sort of image pre-processing (histogram equalization, white balancing, contrast enhancement) can help. This is a very common approach: we first pre-process the images somehow and then we apply the segmentation technique.

You can write your code in Python or in any language of your choice. Make sure you include all your code but also a clear description of everything you are doing.

In addition to your code, please write a report that includes the following sections:

- **Title** (also, your name).
- **Methods.** Describe the three algorithms that you used in this lab. Make sure you provide the values for thresholds and other parameters so that anyone who reads your report can recreate the experiment.
- **Results and discussion.** Show the results that you achieved in your work and offer an interpretation of those results. Acknowledge any limitations of your work.
- **Conclusion.** Summarize your key findings. Discuss benefits or shortcomings of your work.

You don't need to follow a template for your report but here is the US letter template from IEEE in case you want to use it.

Provide evidence for every claim you make (example: "Method A is better than methods B and C" - How can you prove this? What is the metric you are using? Think of the best way to help us, the readers, understand your perspective.

There is a lot of freedom in this lab, so I am expecting very creative approaches and unique results. Have fun and reach out if you have any questions.

LAB 4 - LEGO TIME. AGAIN.

For Lab 2, you worked with a small and relatively simple LEGO dataset (the background was uniform and contrasted with the LEGO pieces). For this lab, you will be working with a more complex LEGO dataset, but your goal will be very similar: to detect and count LEGO pieces.

DETECTING LEGO PIECES

You will train a deep learning model that detects and counts LEGO pieces. You do not need to classify them by colour or type. The only thing you need to do is to detect all the LEGO pieces in an image (using bounding boxes) and then count them.

The Dataset

- Please download the dataset available here. This dataset features 168K annotated images. The images are synthetic, and the annotations follow the PASCAL VOC format. These are a lot of images! You probably do not need all of them. Feel free to reduce your dataset.
- Find a way to make sure that the images you will use are properly annotated (remove images that you think are mislabeled).
- Divide the data into three datasets:
- Training (~70% of the images)
- Validation (~15% of the images)
- Testing (~15% of the images)
- Since the dataset is quite large, you probably won't need to perform data augmentation.
- The dataset features 600 unique LEGO pieces (which means 600 unique labels). No need to use these labels. You can just turn them into a single label called "lego" or something that works for you.

The Neural Network

You have complete freedom to choose any neural network architecture that performs object detection. You also have complete freedom when it comes to the framework you will use to train the neural network. Here are a few possibilities (but there are many more):

- Inception / SSD using TensorFlow.
- Mask R-CNN using PyTorch.
- YOLO using Darknet.

To assess the performance of your model, use the **mean average precision (mAP) when the IoU threshold is set to 0.5.** For more information on how to compute this metric, please see subsection 6.3.3 of Szeliski's book.

What to Submit

In addition to your code, please write a report that includes the following sections:

- **Title** (also, your name).
- **Methods**. Describe the deep learning model that you used in this lab. Make sure you provide the values for thresholds and other parameters so that anyone who reads your report can recreate the experiment.

- **Results and discussion**. Show the results that you achieved in your work and offer an interpretation of those results. Acknowledge any limitations of your work.
- **Conclusion**. Summarize your key findings. Discuss benefits or shortcomings of your work.

You don't need to follow a template for your report but here is the US letter template from IEEE in case you want to use it.

Do not submit the dataset! We already have it.

Important: You do not need to submit your model, but I might ask you to run it for me on a new set of images. Make sure you have a script that allows you to easily do this.

EXTRA QUESTION TO EXPLORE (IF YOU HAVE TIME AND CURIOSITY)

Can your new model detect the LEGO pieces in Lab 2’s dataset? Let me know if you run this experiment.