

# Bricks Sorter

By Ikram Ikramullah (All rights reserved: ikrambabai@gmail.com)



## Problem Statement

Educating smaller kids through fun and well-engineered games has proven to be a crucial defining factor for their education success in life. One of the most fascinating of these are the bricks/Legos. Legos teach kids on how to train their brains on starting with smaller building blocks and then proceeding to constructing bigger projects. From a simple toy car to a sophisticated spaceship that kids love to make, bricks and Legos can be fun and very educational especially towards their STEM skills – if done in an organized manner. Bringing this order is where my project comes into play.

A parent's experience with organizing and sorting hundreds of these pieces of bricks can be overwhelmingly, daunting and time-consuming. The 'learning' can really easily go down the ditch for kids if all they were looking for was a 3x2 brick they cannot find in a pile of hundreds of unsorted bricks. Suddenly all the excitement they had started the project with can go levels down. Learning needs a continuous level of excitement – or else it becomes just another chore.

## Solution Statement

Wouldn't it be cool if each *type* of these bricks were sorted into separate jars for the kids to pick from every time they embarked on a new exciting project? My Brick Sorter is here to here for the rescue.

All the parents / kids would do is to feed the mixed pile of bricks to a machine, let the machine know *how* they like to sort the bricks – and there, in a few moments – you have jars filled with separate type of bricks in each. Doesn't that look awesome (hope I do complete it ☺).

## Domain Background

I think I will put in Educational / STEM category since its helping with the basic idea of learning from building blocks. However, this is just my classification – one may also classify it as toys/games categories. I've passion for STEM and I will rather go with my bias here.

Domain: Education

Subdomain: STEM (Science Technology Engineering Math)

## Datasets and Input

The primary input to my ML Based Brick Sorter application will be an (image of) a brick it needs to sort. Another input will be to let the application know the type of sorting needed – by color, block-type or a combination of the two. The block-type can be one of the three: a brick, a plate or a tile.

**Brick:** Brick is a regular build block and is the most basic type of building block. It measures certain dimensions. It can get connected from the top as well as from the bottom with another component.

**Plate:** A plate is same as a brick except it is much thinner – 3 plates make up one brick in height. Just like a brick, a plate can also connect to components to its top and bottom.

**Tile:** Tile is same as a plate except it can only get connected from the bottom – the top side of a tile is a smooth surface having no studs. Usually these type of building blocks are used for making up of the smooth surfaces such as table tops etc.

There are other specialized building blocks too of course, but these three types of building blocks make up for most of the collection of the building blocks and that's where I will concentrate more on.

Note: the size of a brick is measured by the number of studs that fit on its short side x the number of studs that fit on its longer side. A tile, though doesn't have any studs on it, is also measured using the same rule. i.e. A plate with 2 studs on its shorter side and 3 on the longer side will be measured as 2x3 plate. A tile of the same size (though it won't have any studs) will also be measured the same way: 2x3 tile.

The images aren't yet ready – but I have the blocks which I will take the pictures of, convert to an acceptable dimension and feed in for the training.

The input to the training will therefore be the following.

1. Bricks
  - a. 1x1 Bricks (4 angles)
  - b. 1x2 Bricks (4 angles)
  - c. 2x3 Bricks (4 angles)
  - d. And some more sizes and colors (each in 4 angles)
2. Plates
  - a. 1x1 Plates (4 angles)
  - b. 1x2 Plates (4 angles)
  - c. 2x3 Plates (4 angles)
  - d. And some more sizes and colors (each in 4 angles)
3. Tiles
  - a. 1x1 Tiles (4 angles)
  - b. 1x2 Tiles (4 angles)
  - c. 2x3 Tiles (4 angles)
  - d. And some more sizes and colors (each in 4 angles)

The input to our learning model to get trained on to make those fascinating decisions will be many raw images of various colors of those bricks / Legos – each brick will be captured in various angles (normal position, turned sideways, turned upside down and so on) for the machine to properly learn.

## Benchmark Model

Unfortunately, I don't know if anything of this nature exists. But image recognition in general of course has many applications. I therefore plan to use a simple random model. I will have 4 colors equally represented in input data. A random pick will therefore be of 25% accuracy. My trained model should perform better than that. Similarly, when sorting by the 3 brick-type (brick, plate and tile) – which too will be equally represented in input data set, the random pick will have an accuracy of 33% - my model that sorts based on brick-type should have better accuracy than that.

## Evaluation Metrics

The best evaluation matrix for this project would be to see the accuracy of the output of the machine that puts an object in its respective class.

## Design Outline

I will be using a convolutional neural network with pooling layers support followed by fully connected layers to classify a brick into the intended class.

Following are the two diagrams showing the design outline – the first one is classification based purely on color, the second one uses more sophisticated classification putting color, shape and

dimensionality of the brick into consideration

