LEGO BRICK FINDER

By: Haysum, Noe, Jasim

The aim of the LEGO BRICK Finder is to recognize a set of Lego pieces observed (i.e. their model and color)

The Problem:

It is quite time consuming to find the appropriate LEGO pieces required for your ideas.

this disengaging task can take away from the experience of playing and creating with Legos

Our Solution:

Using this algorithm, you will easily be able to tell what color, brick type and how many you need to make that simple build be it a Car or a Hut.

What is the use of this AI?

To aid in the creative building process and save time and effort on a trivial task.

Instead of having to search around what piece would be good or which would be needed and what color is each piece,

you simply need a camera and this algorithm to tell you what the minimum pieces and their color are needed to build it.

Methodology

We have so far we have collected the data set and mentioned our references. Now we have to set up the code

To make the LEGO brick Finder we need to experiment with different programs and algorithms that match up and make the most

efficient program that can be used

we have a bunch of algorithms:

Convolutional Neural Networks:

is a class of neural networks that specializes in processing data that has a grid-like topology, such as an image

Linear Regression:

Linear Regression is a simple and mathematical way to obtain a result based off of two variables, a dependant and independent.

By plotting the variables in a graph, a correlation or link will be identified through various stages of training and testing, and a

straight solid line (best fit) will be produced.

Predictions can be solidified based on the line of best fit and errors can easily be identified.

TensorFlow:

TensorFlow is an open-source library which can be utilised by Python code. It is described as one the most specialised algorithms

to train and test various models.

TensorFlow uses symbolic math which allows for deep neural network training. The TensorFlow algorithm enables an ability where

the developers can create graphs / structures which can describe the data and how it flows, hence allowing it to be modelled.

We are also going to use a bunch of modules and functions from other sources as well

Such as :

Libraries:

1) Pandas — For handling structured data

2) Scikit Learn — For machine learning

3) NumPy — For linear algebra and mathematic

and use Confusion matrix so it can allow us to visualize the performance of an algorithm, typically a supervised learning one.

Steps:-

1. Loading libraries

2. Loading Datasets

3. Performing Exploratory Data Analysis

- this allows us to analyse the datasets main characteristics as in its a form om data selection

4. Data Preparation: to feed the model

5. Building and training the Deep Learning Model

6. Evaluating the performance of the model

Reference

Here are the links that gave us some information for our project:

Dataset: https://www.kaggle.com/code/kerneler/starter-lego-vs-generic-brick-image-6e1763c0-7

https://www.digikey.com/en/maker/projects/lego-brick-finder-with-openmv-and-edge-impulse/1411a4242d884158ae8f656d5b9b0d53

Dataset: https://www.kaggle.com/code/dannysanchezsanchez/train-lego-brick-classifier

Dataset: https://www.kaggle.com/datasets/joosthazelzet/lego-brick-images

https://www.youtube.com/watch?v=04JkdHEX3Yk&t=69s

<https://www.extremetech.com/extreme/303105-universal-lego-sorter-uses-ai-to-recognize-any-lego-brick>

Data Description

1st Dataset

About Dataset:

40,000 images of 50 different LEGO bricks

in the dataset, you will find 50 different LEGO bricks rendered by 800 different angles

2nd dataset

about the dataset:

Contains 12 classes across 6 brick types, and more than 20,000 images taken by 4 cameras

Classes

The classes in this set are both for Lego and Generic Bricks:

1x1 Lego & Bricks

1x2 Lego & Bricks

1x4 Lego & Bricks

2x2 Lego & Bricks

2x2\_L Lego & Bricks

2x3 Lego & Bricks

File Included

Two Csv Files:

1 - ImageKey.csv: Includes a list of all the images, folder location, the corresponding background image at the time of capture, and the classification

2 - EvaluationKey.csv: The evaluation sets don't have the class in their filename, and in this file you will find each files brick type class.

3 Sets of Images:

1 - Base Images: These are the original takes of each brick, using four cameras. You can use the timestamp of the imagekey.csv to figure out which ones should be grouped.

2 - Cropped Images: These already crop the brick section of the image.