

Lego Brick Classification

Objective: Create your own dataset to classify between three types of lego brick (2x1 , 2x2 , 2x4)

Before starting , we need install packages.

- 1- Python 3.6
- 2- Numpy
- 3- Pandas
- 4- Matplotlib
- 5- OpenCV
- 6- Tensorflow and kearas.

Part 1 ...

- We don't have any dataset, so we will create our data from scratch.
- Create 3 empty folders .. (lego_train , lego_test , lego_pred).

Lego_train => has 3 empty folders (2x1 , 2x2 and 2x4)

Lego_test => has 3 empty folders (2x1 , 2x2 and 2x4)

Lego_pred => is empty.

Open '1- Create_Data.py' and run it :

We initiate this file to make two things:

1- Take auto pic from your camera.

After running the file , the camera will open and will take auto pic each 2 sec and will save it in (lego_train /2x1) **press Esc.**

Then , camera will open again to take pic to lego 2x2 and will save it in (lego_train/ 2x2) **press Esc.**

Then , camera will open again to take pic to lego 2x4 and will save it in (lego_train/ 2x4) **press Esc.**

Then , camera will open again to take pic to lego 2x1 and will save it in (lego_test/ 2x1) **press Esc.**

Then , camera will open again to take pic to lego 2x2 and will save it in (lego_test/ 2x2) **press Esc.**

Finally, camera will open again to take pic to lego 2x2 and will save it in (lego_test/ 2x2) **press Esc.**

2- Increase our data by using (Data Augmentation).

Data augmentation is a strategy that enables practitioners to significantly increase the diversity of **data** available for training models, without actually collecting new **data**. such as cropping, padding, and horizontal flipping are commonly used to train large neural networks.

Part 2 ...

Open '2- Pre_Processing.py' and run it :

We initiate this file :

- 1- To check the number of images in each file.
- 2- To resize all the images with the same size (100x100)
- 3- save the images after pre_process in X_train and X_test, and the labels in y_train and y_test.

After running this file ,you will have this files in your project folder.

(X_train.pickle, X_test.pickle,y_train.pickle,y_test.pickle)

this files contain all the dataset after pre-processing.

Part 3 ...

Open '3- Train_model.py' and run it :

We initiate this file :

- 1- Import our data from pickle files.
- 2- Use Keras on train our model and save it in our project folder.
- 3- Check the accuracy and the loss of our testing data.

After running this file ,you will have this file in your project folder. (model2.h5)

Part 4 ... (Last part)

Open '4- Detect Lego.py' and run it :

We initiate this file :

To open the camera and take one pic , then classify it (2x1 ,2x2 or 2x4).

This pic will be saved in 'lego_pred' folder.