Divyansh Chawla
3rd Year UG student, Economic Sciences
IIT Kanpur
chdivy@iitk.ac.in
github

## What I did?

I have created a custom CNN model along the lines of AlexNet to classify 4 classes of images namely Beds, Cabinets, Chairs and Desks. It consists of three convolution layers using ReLU activation, each of them followed by a max-pooling layer. At the end are two fully connected layers using ReLU and softmax activation.

I have scraped images with a JavaScript Scraper which was built using the JSS library and zepto.js library. It's code has been included in the zip file submitted and can also be found here.(I've not written this scraper and it works amazingly so i chose it.)

Owning to the small number of images on the IKEA website(even when scraped from different IKEA websites for different countries), I used the technique of data augmentation with the help of keras.preprocessing.image.ImageDataGenerator class to generate augmented images. All the images I scraped can be found here in the data folder.

Next I've used numpy and matplotlib.pyplot to generate confusion matrix and pandas to generate a CSV of predicted values of testing images.

## Why I did so?

CNNs are the most popular neural network model used for image classification. CNNs are faster than traditional algorithms since they possess parameter sharing and local connectivity. Pooling layers  progressively reduce the spatial size of representation to reduce the amount of parameters and computations in the network. They are at least as accurate as a traditional neural network.

AlexNet is one of the fastest implementation of CNNs. It consists of 5 convolutional layers and 3 fully connected layers. I used 3 convolutional layers and 2 fully connected layers to reduce the computational costs. My implementation is also reasonably accurate.

Since it was a Multi-Class, Single Label Classification problem,  therefore I chose the

Last layer activation as “softmax” and loss function as “categorical_crossentropy”.

Data augmentation became necessary to expand my dataset because of the small sample size of training images. It also helps to reduce irrelevant features in the dataset and prevents overfitting.

## Which technique would perform better than my implementation?

Any of the techniques who won in ImageNet Large Scale Visual Recognition Challenge (ILSVRC) would perform better than my implementation, namely AlexNet, ZFNet, GoogleNet, VGGNet and ResNet. The main reason being that they contain more hidden layers to process the dataset and have been designed by experimenting on datasets by multiple hit and trials. Using pre-trained models on COCO dataset will also give better results since the COCO dataset already contains the classes ‘chairs’, ‘desks’, ‘bed’ and ‘dining tables’.