# Deep Learning – Case Study

## Crowd Counting Using Sequential Model

**Name:** Priyank Bhavsar

**Enrollment Number:** 19012012009

**Batch:** DL2

## 1. Introduction

This case study is designed to Count the number of people in image. It uses the Sequential for the images. The model built is able to grid the image and train model accordingly and then predict the sign from them.

This case study can be used across various sectors where there is a need to count number of people from image. Also, if this model is trained with more resources like GPU, RAM then it can be expanded to cover variety of traffic signs categories.

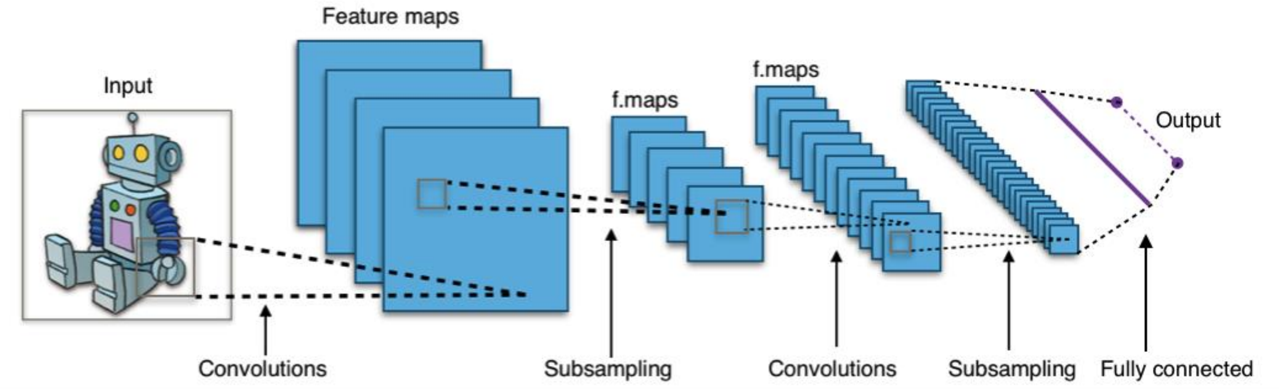
## 2. Tools and Technologies

| Tools and Libraries | Usage |
|---|---|
| **Keras** | This library is used for building the network architecture. It allows us to use several layers, callbacks, and InceptionResNetV2 model. |
| **Sequential** | A Sequential model is appropriate for a plain stack of layers where each layer has exactly one input tensor and one output tensor. |
| **matplotlib** | Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy. |
| **Kaggle** | Used as a platform to execute code as well as manage datasets and model weights |

## 3. Model Explanation and Architecture

In a sequential model, we stack layers sequentially. So, each layer has unique input and output, and those inputs and outputs then also come with a unique input shape and output shape. If you want to retrieve the weights a layer has, you can simply call get weights on layer and retrieve all those weights as as a list of numpy arrays.

First, we instantiate a sequential model, then we add layers to it one by one. Afterwards, we need to compile a model with a mandatory loss function and a mandatory optimizer, and if we like, also with optional evaluation metrics such as accuracy.



## 4. Working

At first we add all the necessary libraries that are required for the project after that we will upload the images and label them then images are loaded in vector format. After all the pre-processing we gave created a sequential model after that we add a learning rate monitor to get the lr with smoothest prediction.

After Model creating we will train the model and we change the learning rate to 1e-5 and re-run the model then it will train the model again and at last for better understanding we plot an graph.

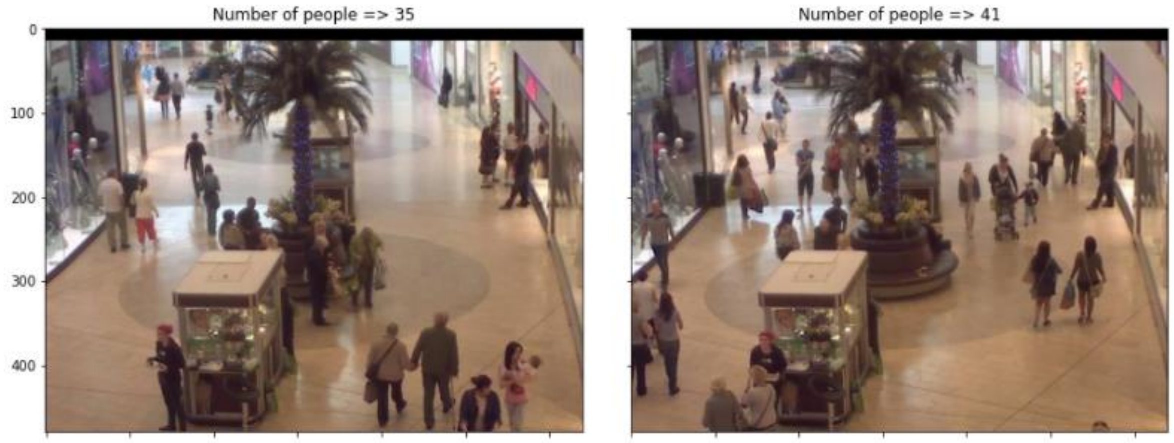
At last when the model is ready we apply any random image in that model to check whether it is trained properly or not.

## 5. Code

https://github.com/Priyank-17/Crowd-Counting/blob/main/crowd-counting.ipynb

** Code and Report is available in the GitHub repository.

## 6. Output



---------------