German University in Cairo
Faculty of Media Engineering and Technology
Deep Learning for Self-Driving Cars
CSEN1094

Project 2: German Traffic Sign Classification

You are going to build a deep neural network to classify traffic signs.Your network should be able to take an image and decide if it is a stop sign, a yield sign, a different type of sign or maybe no sign at all. You will implement the project using Python and Tensorflow or Keras.

Step 1: Load the Data

Download the dataset from here. This is a pickled dataset in which images are resized to 32x32.
The dataset has three .p files of 32x32 resized images:

- train.p: The training set.
- valid.p: The validation set.
- test.p: The testing set.

use Python pickle to load the data.

Step 2: Dataset Summary

The pickled data is a dictionary with 4 key/value pairs:

- **features** is a 4D array containing raw pixel data of the traffic sign images (num examples, width, height, channels).
- **labels** is a 1D array containing the label/class id of the traffic sign. The file signnames.csv contains id to name mappings for each id.
- **sizes** is a list containing tuples, (width, height) representing the original width and height the image.
- **coords** is a list containing tuples, (x1, y1, x2, y2) representing coordinates of a bounding box around the sign in the image.

First, You are required to illustrate the following:

- Number training examples.
- Number of testing examples.
- Number of validation examples.
- Number of classes.

Then, plot sample images from each subset. And finally, plot a histogram of the count of images in each unique class.

Step 3: Data Preprocessing

In this step, You can apply as many preprocessing techniques as you need to the input images to achieve the best possible results.

Preprocessing Techniques:

- **Shuffling**
- **Grayscaling**
- **Local Histogram Equalization**
- **Normalization**

Step 4: Network Architecture

In this step, You are required to implement a deep learning model that learns to recognize traffic signs from the dataset. You can use one of the following architectures:

- **LeNet-5**

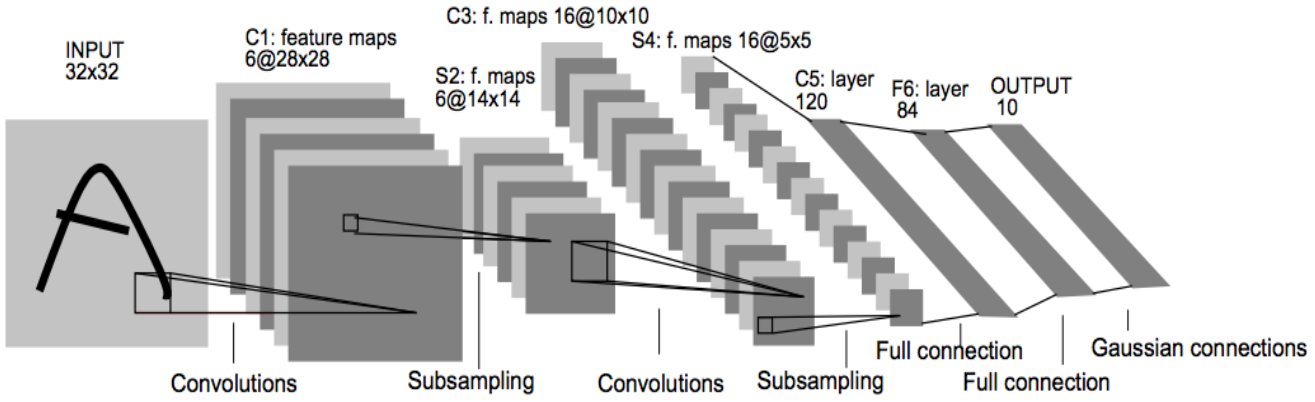

Figure 1: LeNet-5

- **VGGNet**

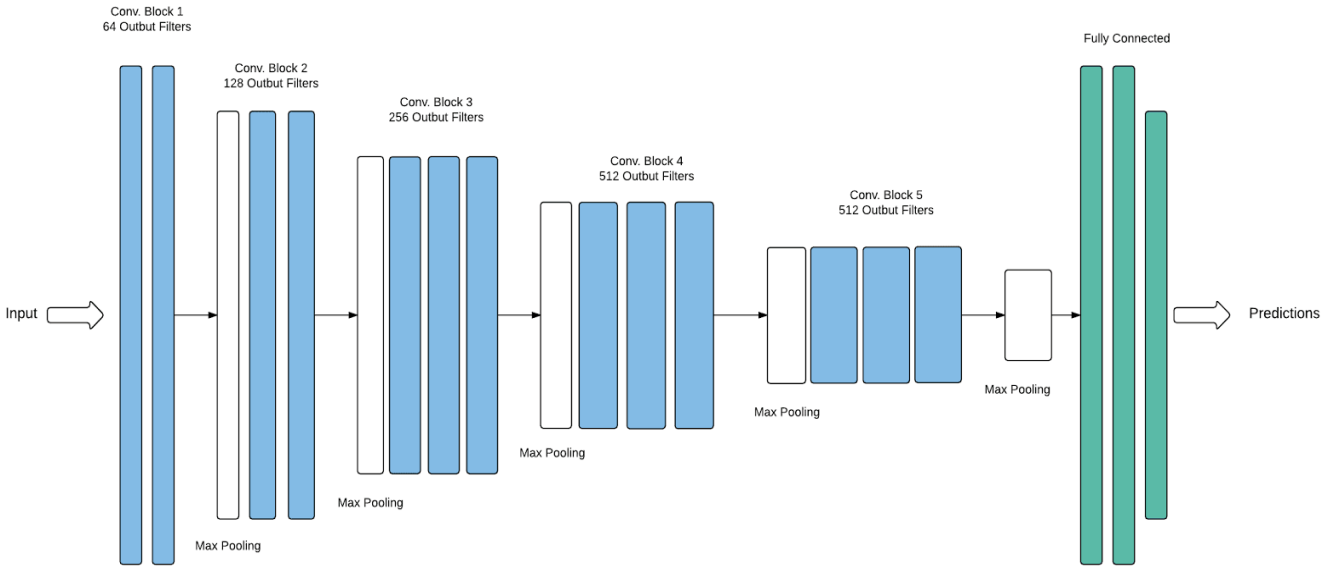

Figure 2: VGGNet

Step 5: Training and Testing

Train and validate your model on the preprocessed training and valiadtion sets and then test your model performance using the test set. You are required to visulaize the results of training, validation and testing.