## horizontal line



Traffic Sign Classification

Using CNN

**─**

# Cover Sheet

## Faculty Name

Faculty of Computers and Artificial intelligence Helwan university

## Course Name

Selected Topics in Computer Science 2.

## Team Number

61

## Team Members

محمود محمد عباس 20180570

يوسف عبد العاطى شحاته 20180710

محمد ابو نعمان عبد الفتاح 2018456

محمد ناصر تمام 20180543

محمد محسن عيسى حافظ 20180530

يوسف عصام احمد محمد 20180712

محمد خالد حسن 20180487

# Paper Details

## Paper Citation

Authors’ Names: Bailke, Preeti & Agrawal, Kunjal.   
Paper Name: Traffic Sign Classification Using CNN  
Publishers: Bailke, Preeti & Agrawal, Kunjal.  
Year of Publication: 2022  
Journal: International Journal for Research in Applied Science and Engineering Technology.10. 10.22214/ijraset.2022.40224.

## Used Dataset

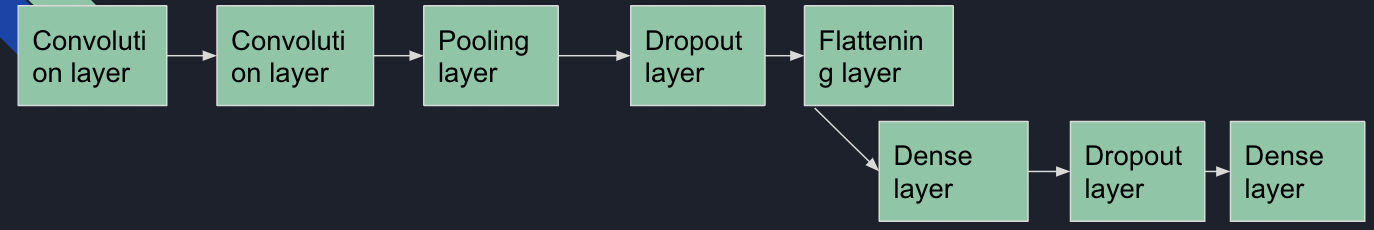
GTSRP

Total Traffic Sign images = 50,000

Classes = 43

## Implemented Algorithm

**CNN Model**

****

1- In the convolutional layer, the number of filters is specified. It performs the convolution operation on the original image and

generates a feature map.

2- The ReLU performs the maximum function to convert the negative values to zero without changing the positive ones and

generate a rectified feature map. The Pooling layer takes the rectified feature map and performs a down-sampling operation

(like Max Pooling or average pooling) and thus reduces the dimensionality of the image.

3- The flattening layer is used to convert the input feature map to a 1-dimensional array.

4- The dropout layer is used to avoid overfitting by setting some of the input neurons to 0 during the training process. The dense

layer, on the other hand, feeds all the outputs from the preceding layer to all its neurons and perform the matrix- vector

multiplication (the row vector of the output from the preceding layer should be equal to the column vector of the dense layer),

to generate a m-dimensional vector.

## Results

* Training Loss 9%
* Validation Loss 16%
* Test Accuracy 93%

# Project Description

## General Information on The Selected Dataset

* We found Traffic Sign Dataset - Classification dataset on kaggle.com: <https://www.kaggle.com/datasets/ahemateja19bec1025/traffic-sign-dataset-classification?select=traffic_Data>
* It contains total number of 6674 images
* The dataset has folder DATA which contains 58 folders labeled from 0 to 57, each folder represents a class of traffic signs but the last 3 folders are unknown.
* The type of images is RGB and images are of different sizes.
* The dataset has folder TEST which contains random images.
* We made some changes on this dataset :

- we removed last 3 classes because they were unknown so we have 55 classes  
The labels are:

|  |  |
| --- | --- |
| ClassId | Name |
| 0 | Speed limit (5km/h) |
| 1 | Speed limit (15km/h) |
| 2 | Speed limit (30km/h) |
| 3 | Speed limit (40km/h) |
| 4 | Speed limit (50km/h) |
| 5 | Speed limit (60km/h) |
| 6 | Speed limit (70km/h) |
| 7 | speed limit (80km/h) |
| 8 | Dont Go straight or left |
| 9 | Dont Go straight or Right |
| 10 | Dont Go straight |
| 11 | Dont Go Left |
| 12 | Dont Go Left or Right |
| 13 | Dont Go Right |
| 14 | Dont overtake from Left |
| 15 | No Uturn |
| 16 | No Car |
| 17 | No horn |
| 18 | No overtaking by truck restrict end |
| 19 | Speed limit (50km/h) restrict end |
| 20 | Go straight or right |
| 21 | Go straight |
| 22 | Go Left |
| 23 | Go Left or right |
| 24 | Go Right |
| 25 | keep Left |
| 26 | keep Right |
| 27 | Roundabout mandatory |
| 28 | watch out for cars |
| 29 | Horn |
| 30 | Bicycles crossing |
| 31 | Uturn |
| 32 | Road Divider |
| 33 | Traffic signals |
| 34 | Danger Ahead |
| 35 | Zebra Crossing |
| 36 | Bicycles crossing |
| 37 | Children crossing |
| 38 | Dangerous curve to the left |
| 39 | Dangerous curve to the right |
| 40 | Go right or straight |
| 41 | Go left or straight |
| 42 | Unknown4 |
| 43 | ZigZag Curve |
| 44 | Train Crossing |
| 45 | Under Construction |
| 46 | Unknown5 |
| 47 | Fences |
| 48 | Heavy Vehicle Accidents |
| 49 | Unknown6 |
| 50 | Give Way |
| 51 | No stopping |
| 52 | No entry |
| 53 | No overtaking |
| 54 | Yield |

- we added random images from the internet for each type in the folder DATA

- we added images that were in the TEST folder to a suitable folder in the DATA folders, which made us able to split the data the way shown in the paper(65% train, 25% validation, 10% test).

## Implementation Details

* Dataset Split (65% train - 4338 images, 25% validation - 1668 images, 10% test - 667 images).
* Block Diagram:

Splitting dataset:  
- Using train\_test\_split()

Building and compiling CNN model

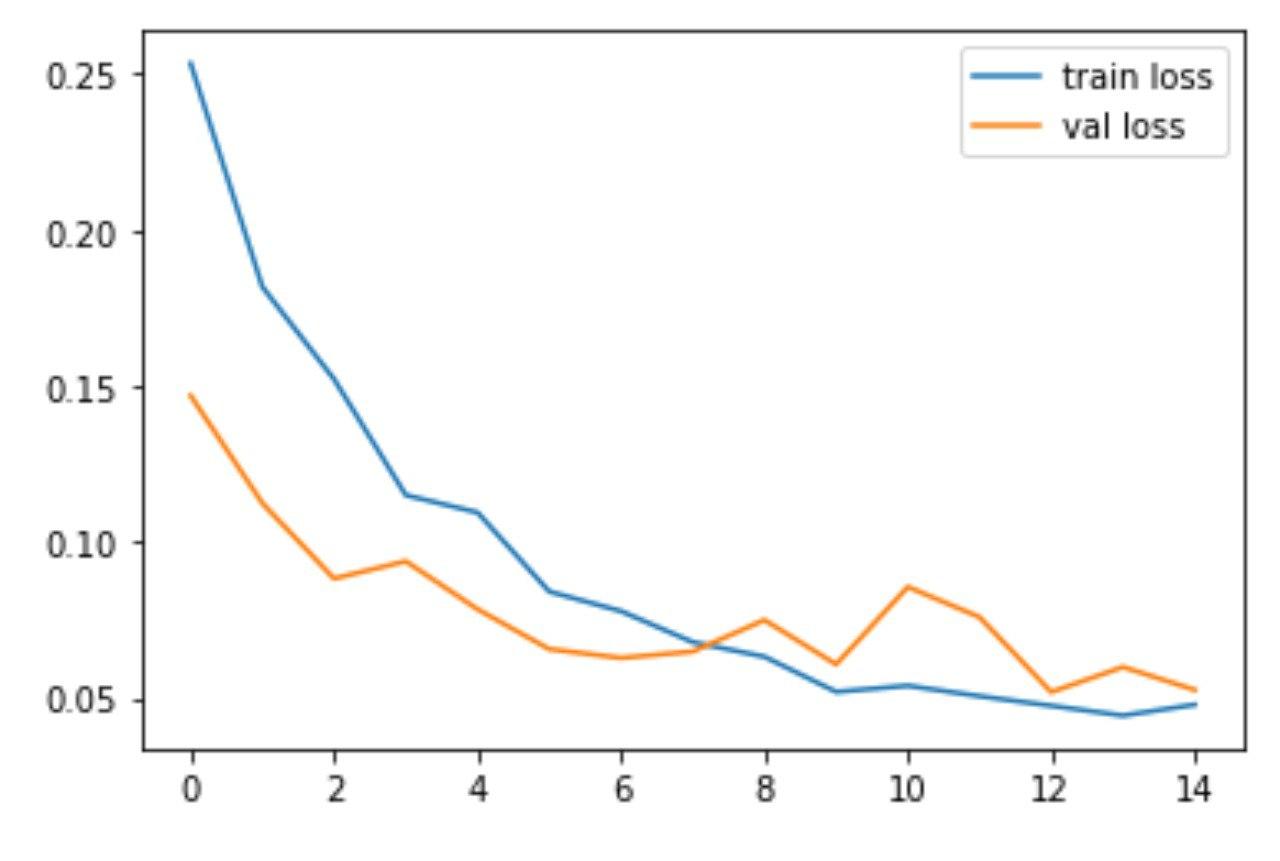
Training the model

Preprocessing dataset:  
- resizing  
- grayscale  
- histogram equalization  
- Normalization

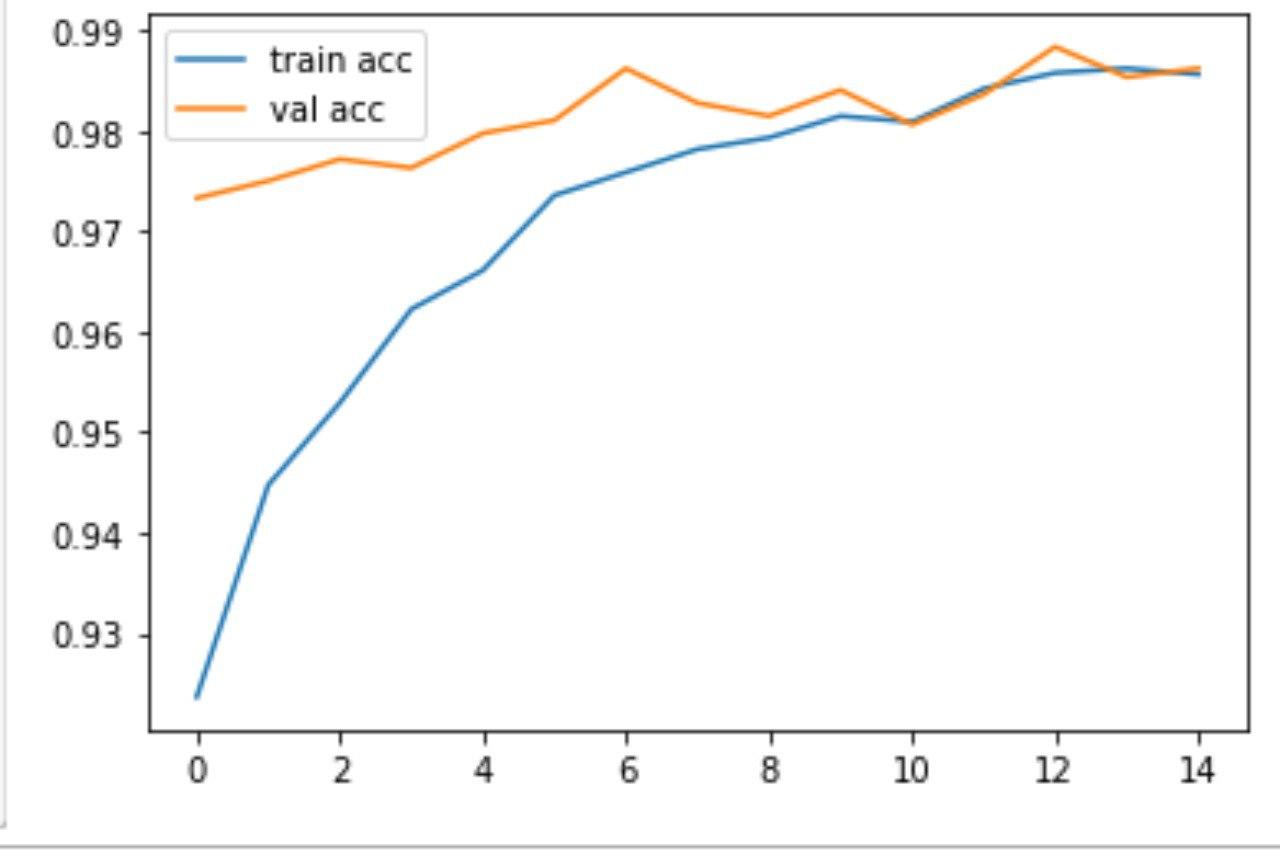
* Hyper Parameter:  
  epochs: 15  
  Adam optimizers default hyperparameters: learning rate = 0.001,..etc

## Result Details

* Training Loss: 0.04779958352446556
* Validation Loss: 0.05254281312227249



* Training accuracy: 0.9856507182121277
* Validation accuracy: 0.9862306118011475



* Confusion Matrix: array([[32, 0, 0, ..., 0, 0, 0],

[ 0, 11, 0, ..., 0, 0, 0], [ 0, 0, 36, ..., 0, 0, 0],

..., [ 0, 0, 0, ..., 27, 0, 0],

[ 0, 0, 0, ..., 0, 43, 0],

[ 0, 0, 0, ..., 0, 0, 6]], dtype=int64)