Traffic Signs Recognition

By Mada Abudahish and Asrar Abdulrhman

Introduction

In the world of **Artificial Intelligence** and advancement in technologies, many researchers and big companies are working on autonomous vehicles and self-driving cars.

So, for achieving accuracy in this technology, the vehicles should be able to interpret traffic signs and make decisions accordingly.

Problem Statement

Given a various traffic signs along with images dataset , we are challenged to classify traffic signs present in the image into different categories.

Dataset

Contains different traffic signs.
classified into 43 different classes.

size

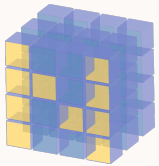
50,000 images

Source

From [Kaggle]

Tools

Data manipulation and Preprocessing ML libraries:


NumPy

Visualization libraries:

matplotlib

Deep learning:


TensorFlow

workflow

01

Data Loading

02

Preprocessing

03

Baseline Model

04

Build Deep Learning Model

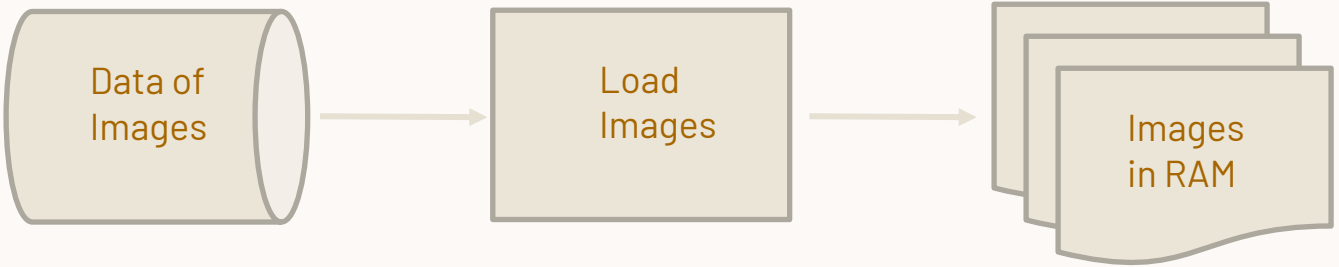
05

Applying Transfer Learning

06

Future work

01 Data Loading



02

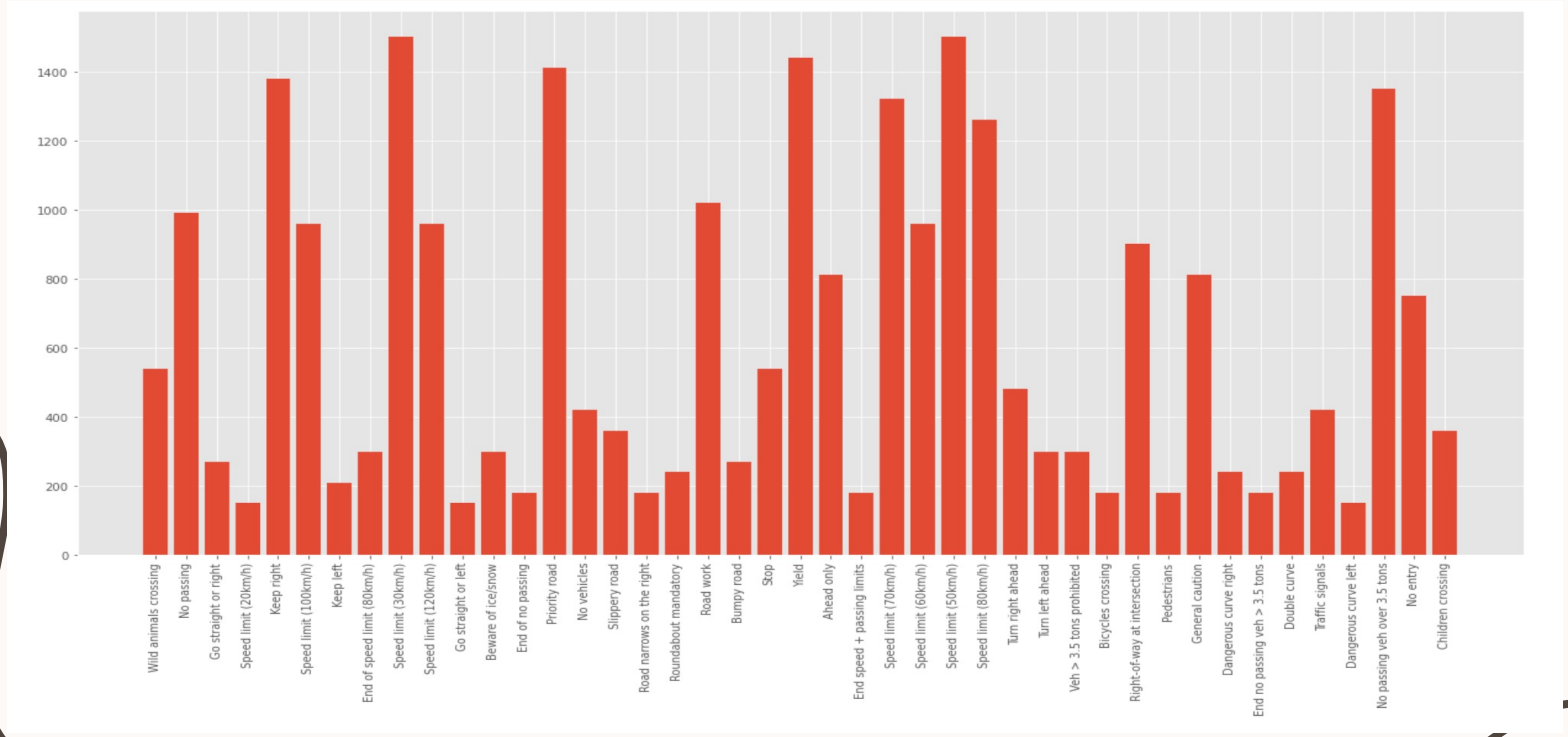
Image Preprocessing

- Set height and width to be 32x32
- Normalization, Divide by 255

Model Preprocessing

- Splitting the Data
- Sets the labels and check the distribution.
- Labels Encoded

Class Imbalance Struggle

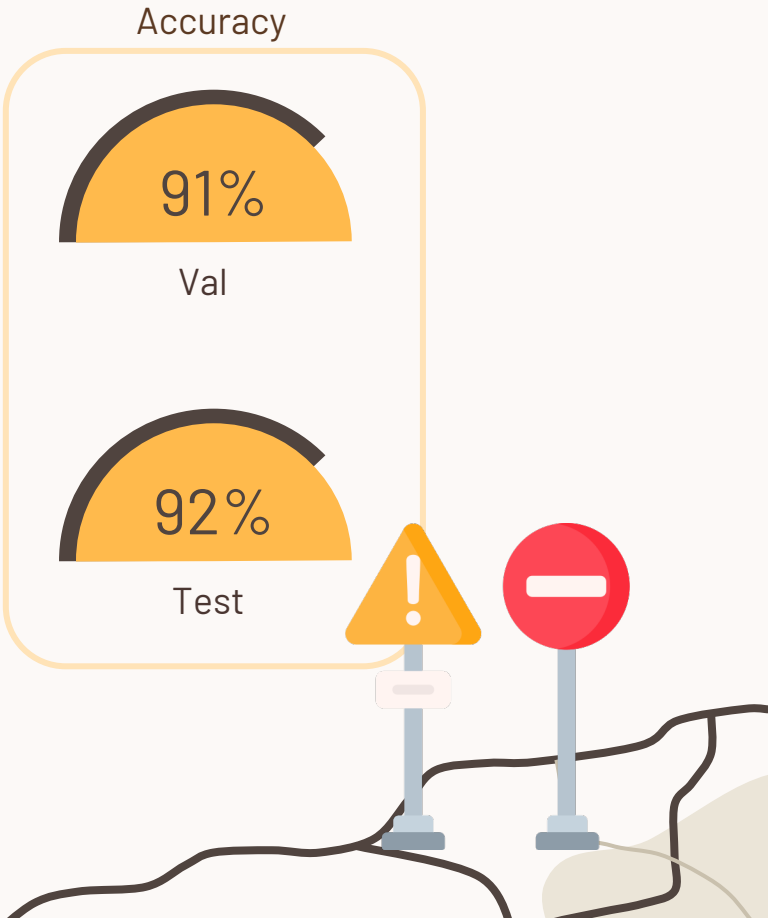


03

Baseline Model

Logistic Regression

- Reshape the image size to be in 2d array
- Scaling using Stander Scaler



04

Build Deep Learning Model

Convolutional Neural Networks

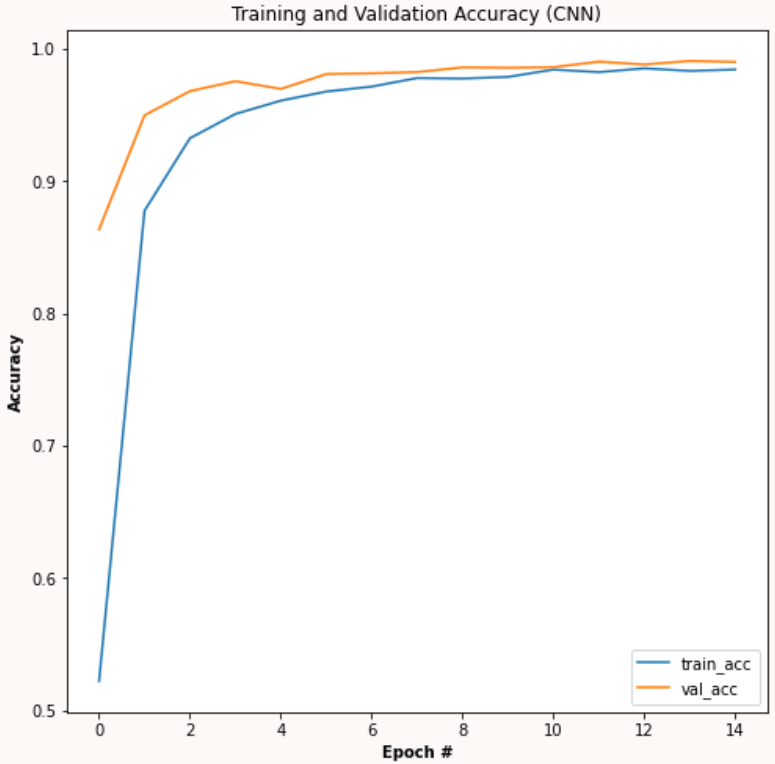
hyperparameter

Hidden layer :
- Three Convolution Layers
- Two simple NN layers

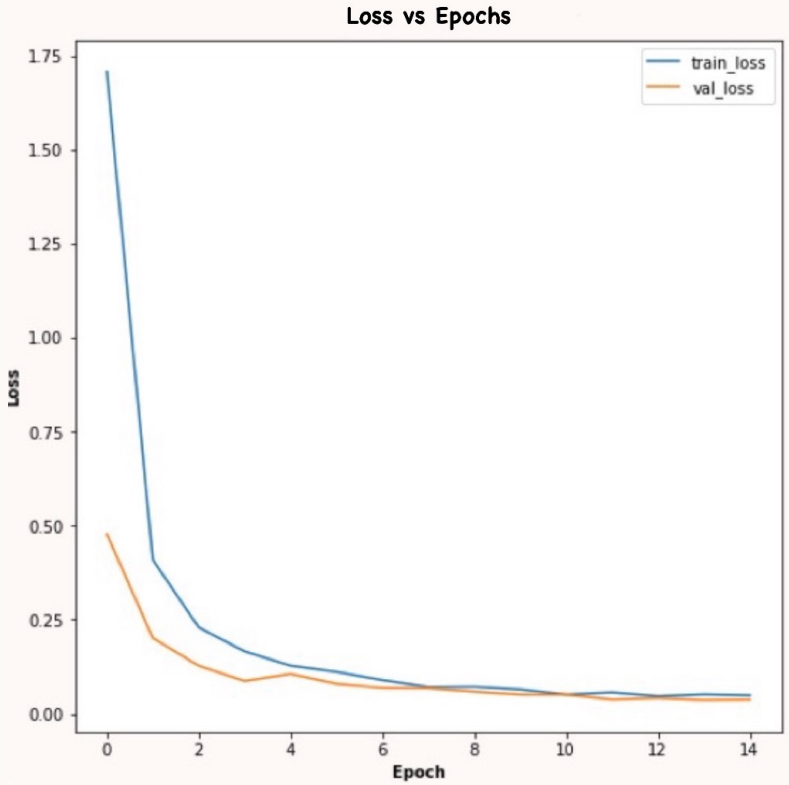
Kernel size: 3
Dropout: 0.25
Activation Function:
- Relu
- Softmax

Epochs: 15
Class Weight:
- Balanced

Val Accuracy: **0.9920**
Test Accuracy: **0.9843**

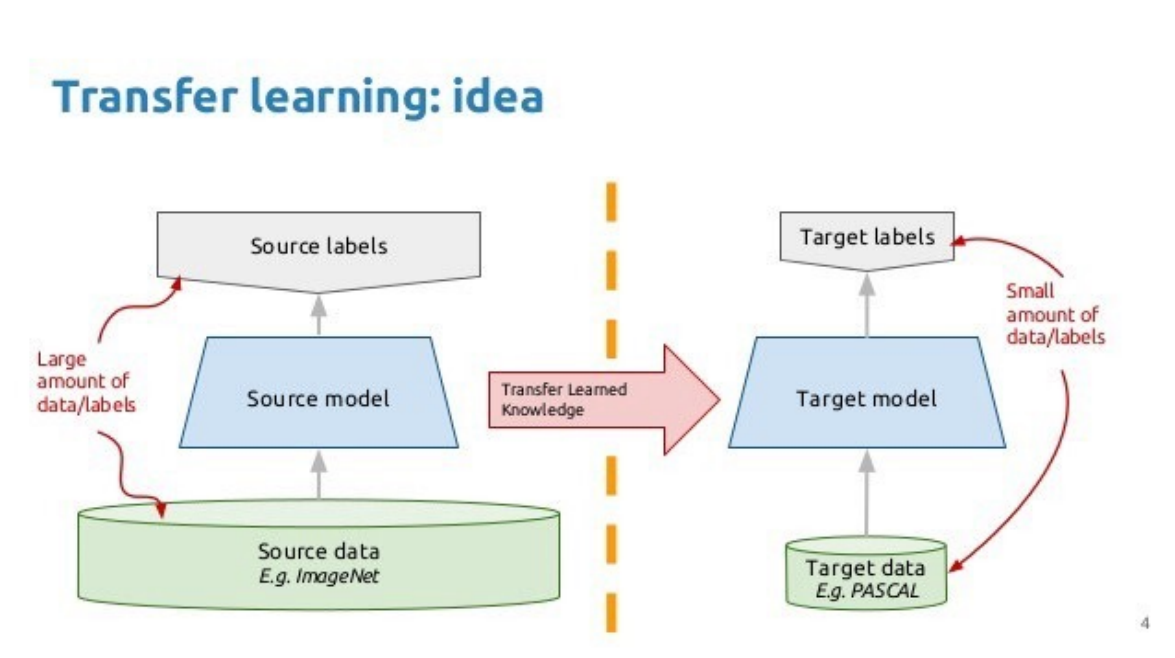


Convolutional Neural Networks



Loss: Categorical Crossentropy
Optimizer: Adam
Metrics: Accuracy

Applying Transfer Learning

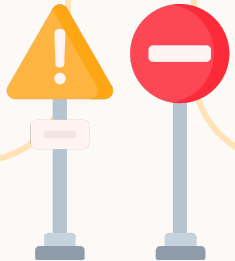


Applying Transfer Learning

VGG16: (CNN) architecture which was used to win ILSVR(ImageNet) competition in 2014. It is considered to be one of the excellent vision model architecture till date.

val Accuracy:
0.99

Test Accuracy:
0.99



MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications

val Accuracy:
0.91

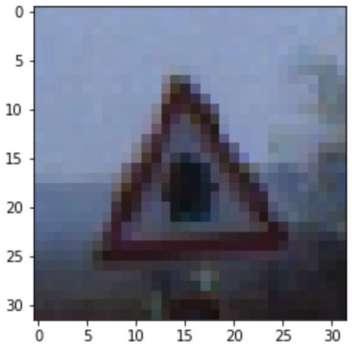
Test Accuracy:
0.96

Predicting with one Image from Test Dataset

In [83]:
```
image_index = 11
```

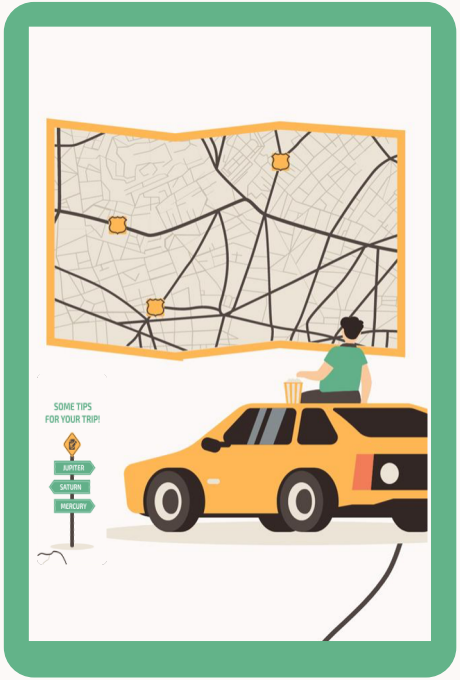
In [84]:
```
plt.imshow(X_test[image_index])
n = np.array(X_test[image_index])
print(n.size)
p = n.reshape(1, 32, 32, 3)
pred = classes[model.predict(p).argmax()]
print("The predicted image is {}".format(pred))
```

```
3072
The predicted image is Right-of-way at intersection
```



Future Work

- Increase the obtained result by using different DL approaches which are supported by extra feature extraction methods.
- Autonomous car system to Recognize Traffic Signs.



Thank you..