

# A Glimpse into Automotive Computer Vision Using Convolutional Neural Networks

by : Eric H. Lewis

Last Update : August 12th 2021



# Table of Contents

01

## Understanding the Motivation

- ❖ Describe Computer Vision
- ❖ Uses in Automotive Industry

02

## Concepts

- ❖ Image Classification
- ❖ Object Detection
- ❖ Image Segmentation

03

## Data

- ❖ Origins of the Data
  - German Dataset

04

## Data Augmentation

- ❖ Resize
- ❖ GrayScale
- ❖ Normalization

05

## CNN / Results

- ❖ CNN Accuracy : 0.98

06

## Pitfalls / Solutions

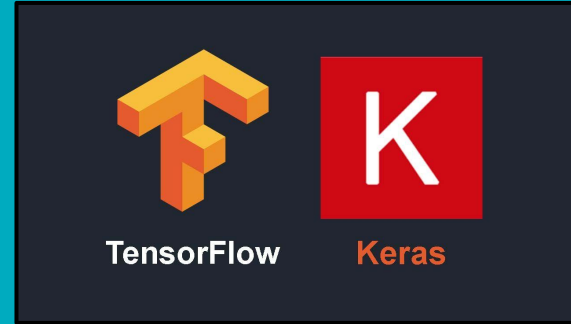
- ❖ Downsides of Augmentation

07

## Conclusion

- ❖ Proof of Concept for I.C

# Tools /Technologies



# Understanding the Motivation

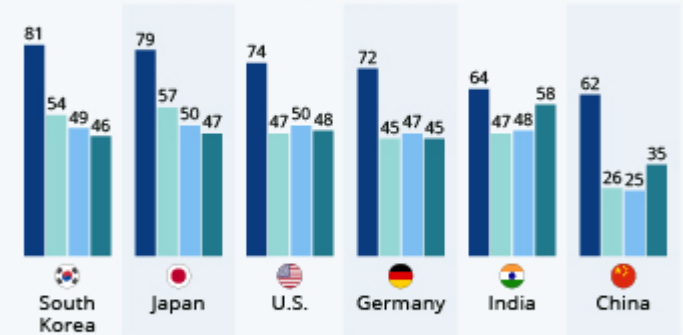


## Where People Are Warming up to Self-Driving Cars

Percentage of consumers who think self-driving vehicles will not be safe



■ 2017 ■ 2018 ■ 2019 ■ 2020



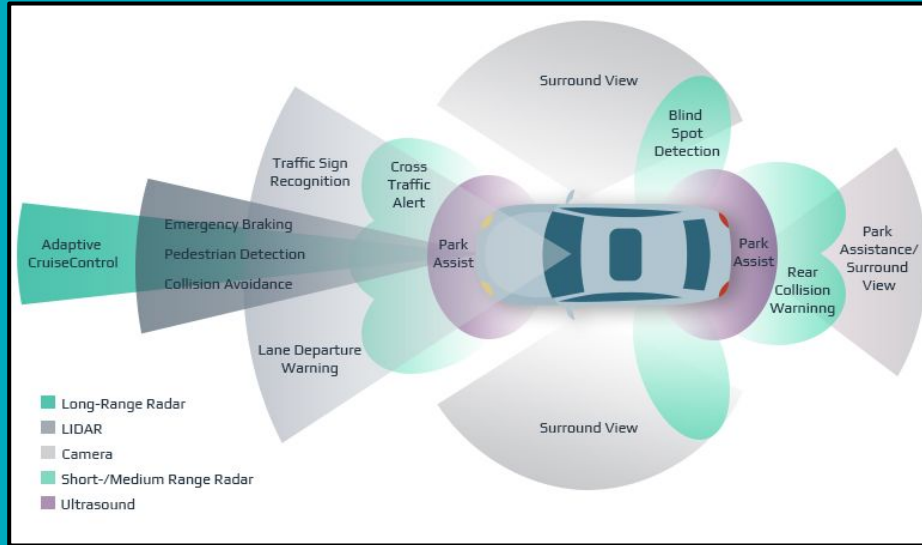
1,000+ consumers surveyed per country

Source: Deloitte

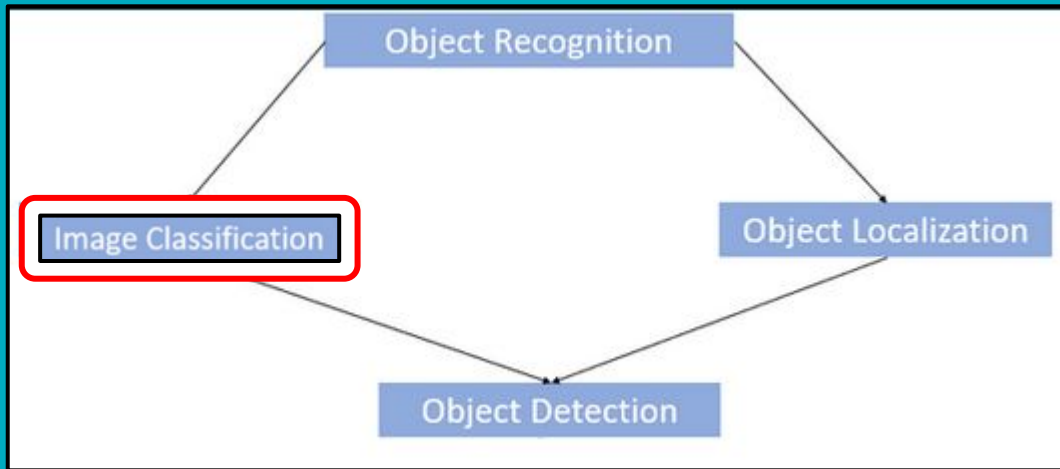


statista

# Understanding the Motivation



# Computer Vision Essentials

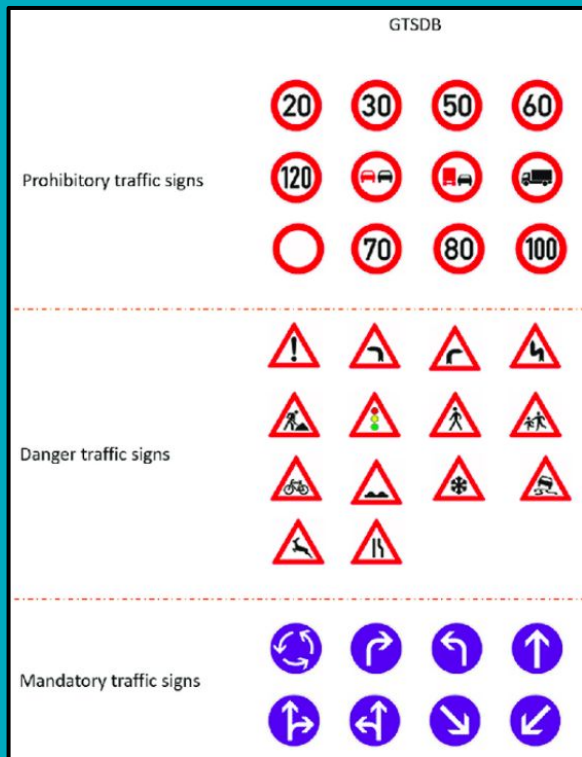


# Data

## Image Classification



# Dataset for Image Classification

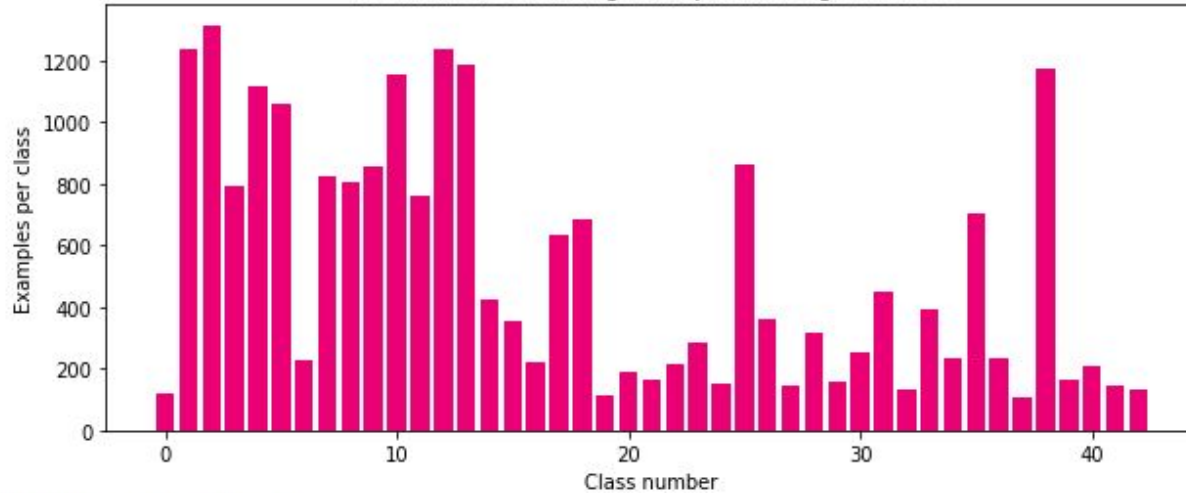


ClassId	Name
0	Speed limit (20km/h)
1	Speed limit (30km/h)
2	Speed limit (50km/h)
3	Speed limit (60km/h)
4	Speed limit (70km/h)
5	Speed limit (80km/h)
6	End of speed limit (80km/h)
7	Speed limit (100km/h)
8	Speed limit (120km/h)
9	No passing
10	No passing for vehicles over 3.5 metric tons
11	Right-of-way at the next intersection
12	Priority road
13	Yield
14	Stop
15	No vehicles
16	Vehicles over 3.5 metric tons prohibited
17	No entry



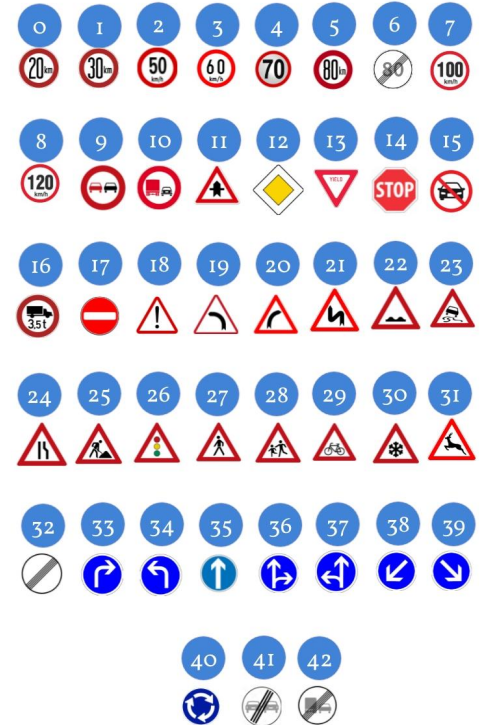
# Distribution of Data

Distribution of Training Examples Amongst Classes



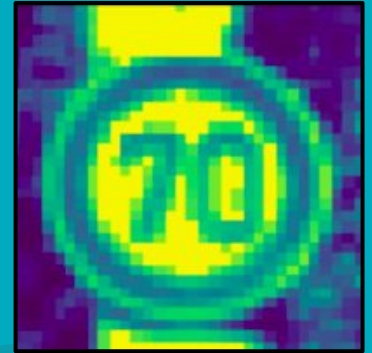
(22271, 32, 32, 3)

Road Signs and their Labels

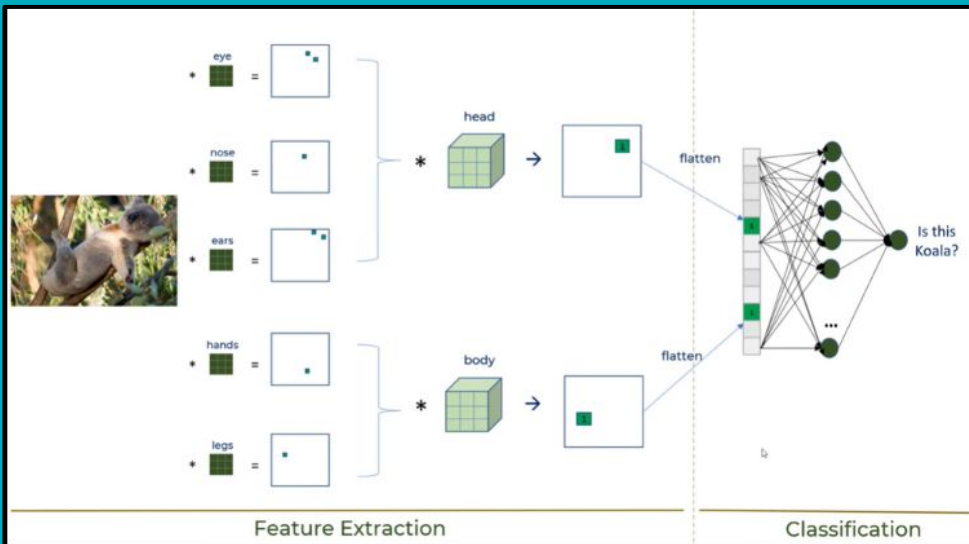


# Data Augmentation

- Resize (32, 32, 3)
- Grayscale
- Equalization
- Normalization



# Convolutional Neural Network



## Input Image Dimensions

(32, 32, 3)



Model: "sequential"

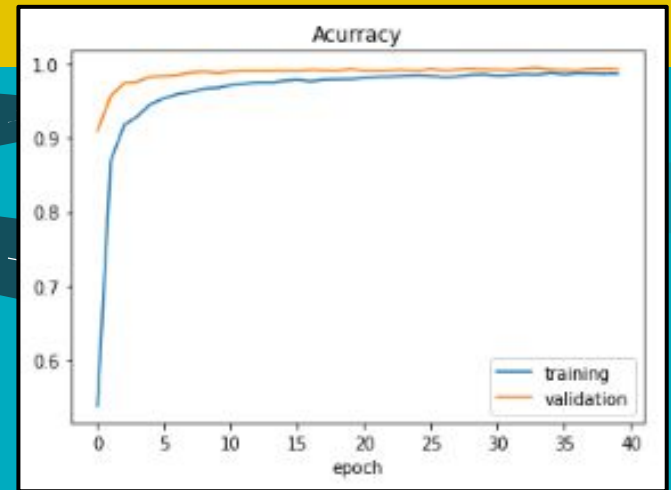
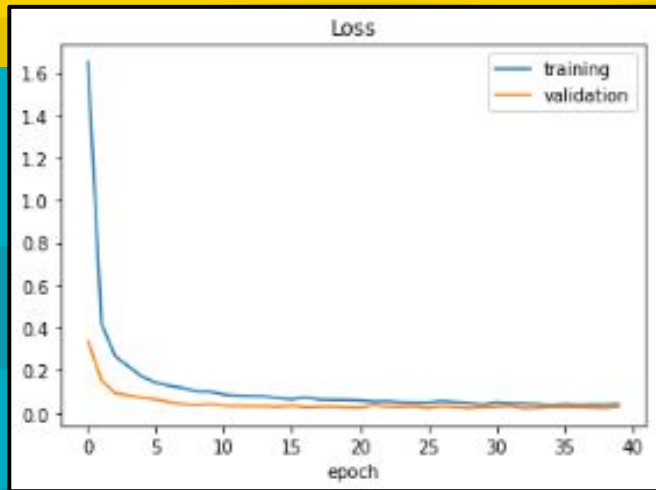
Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 28, 28, 60)	1560
conv2d_1 (Conv2D)	(None, 24, 24, 60)	90060
max_pooling2d (MaxPooling2D)	(None, 12, 12, 60)	0
conv2d_2 (Conv2D)	(None, 10, 10, 30)	16230
conv2d_3 (Conv2D)	(None, 8, 8, 30)	8130
max_pooling2d_1 (MaxPooling2D)	(None, 4, 4, 30)	0
dropout (Dropout)	(None, 4, 4, 30)	0
flatten (Flatten)	(None, 480)	0
dense (Dense)	(None, 500)	240500
dropout_1 (Dropout)	(None, 500)	0
dense_1 (Dense)	(None, 43)	21543

Total params: 378,023

Trainable params: 378,023

Non-trainable params: 0

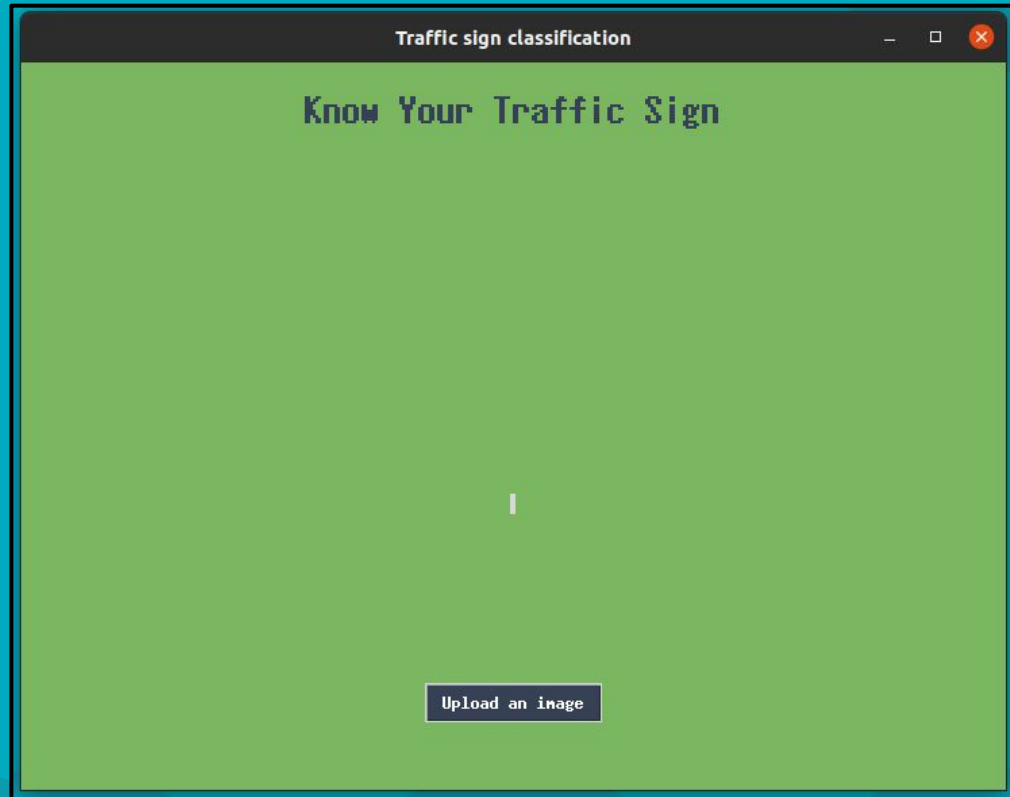
# Results



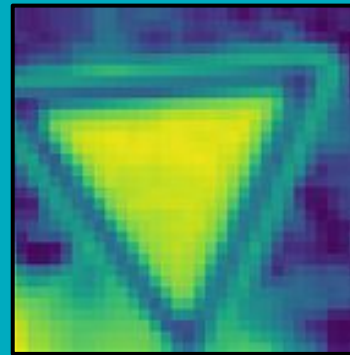
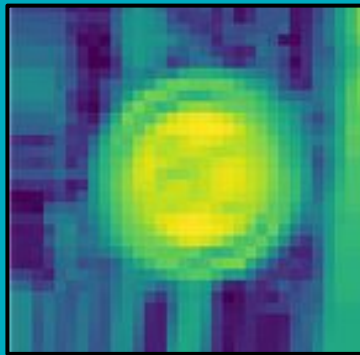
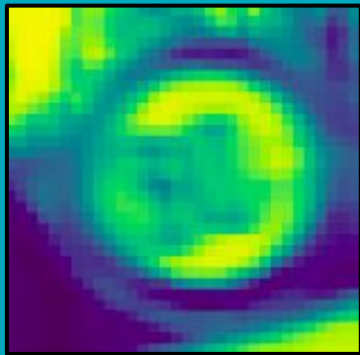
	Loss	Accuracy
CNN	0.0514	0.9842



# TKinter GUI

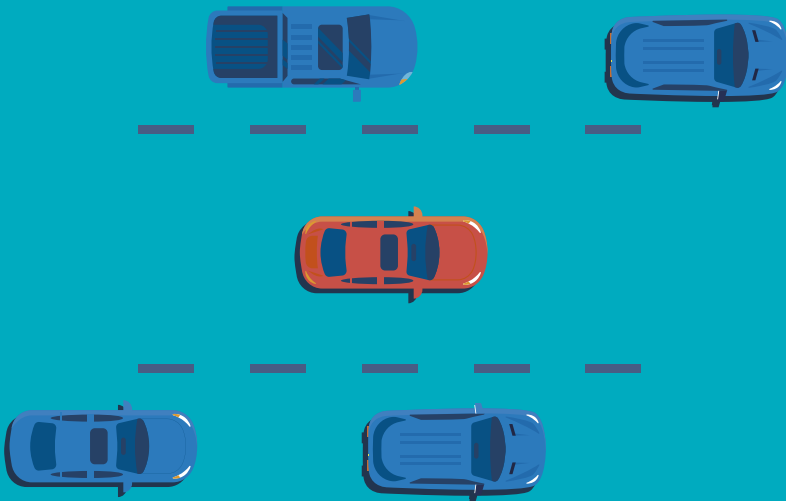


# Pitfalls



# Solutions

- Rolling Training Sets
- Gather more data.



# Thanks!

Do you have any questions?

