Traffic Sign Classification CNN

By Alonso Bravo

Why Traffic Sign Classification?

- Autonomous Vehicles
- Vehicle Driving Assistance Features

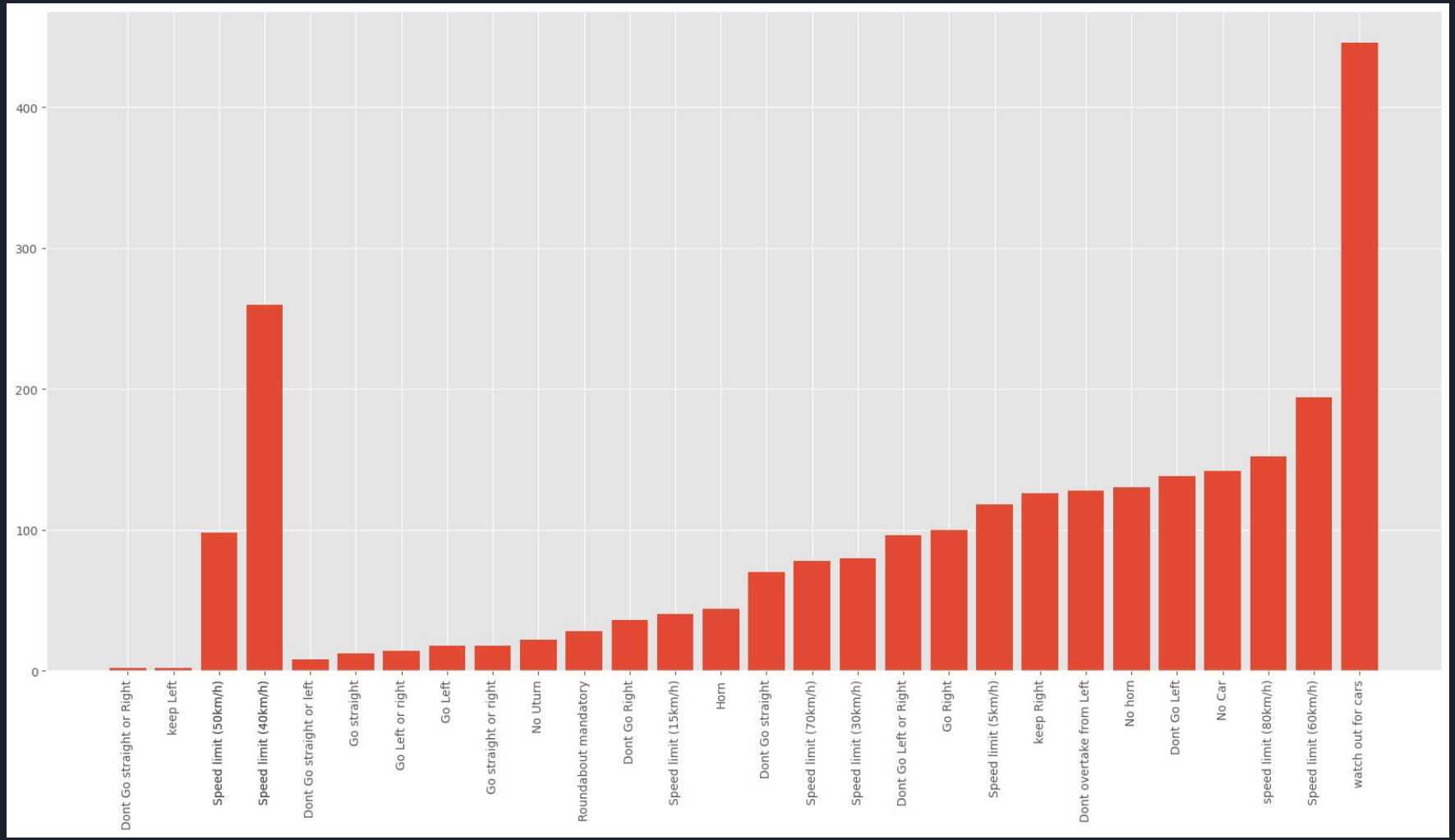
Dataset Exploration

German Traffic Sign Recognition Benchmark (GTSRB)

- 30 Classes
- Training Set: 2607 Traffic Sign Images
- Testing Set: 1200 Traffic Sign Images

- Class Inconsistencies

```
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)
```


400
300
200
100
0
Dont Go straight or Right
keep Left
Speed limit (50km/h)
Speed limit (40km/h)
Dont Go straight or left
Go straight
Go Left or right
Go Left
Go straight or right
No Uturn
Roundabout mandatory
Dont Go Right
Speed limit (15km/h)
Horn
Dont Go straight
Speed limit (70km/h)
Speed limit (30km/h)
Dont Go Left or Right
Go Right
Speed limit (5km/h)
keep Right
Dont overtake from Left
No horn
Dont Go Left
No Car
speed limit (80km/h)
Speed limit (60km/h)
watch out for cars

9 Random Images from Dataset

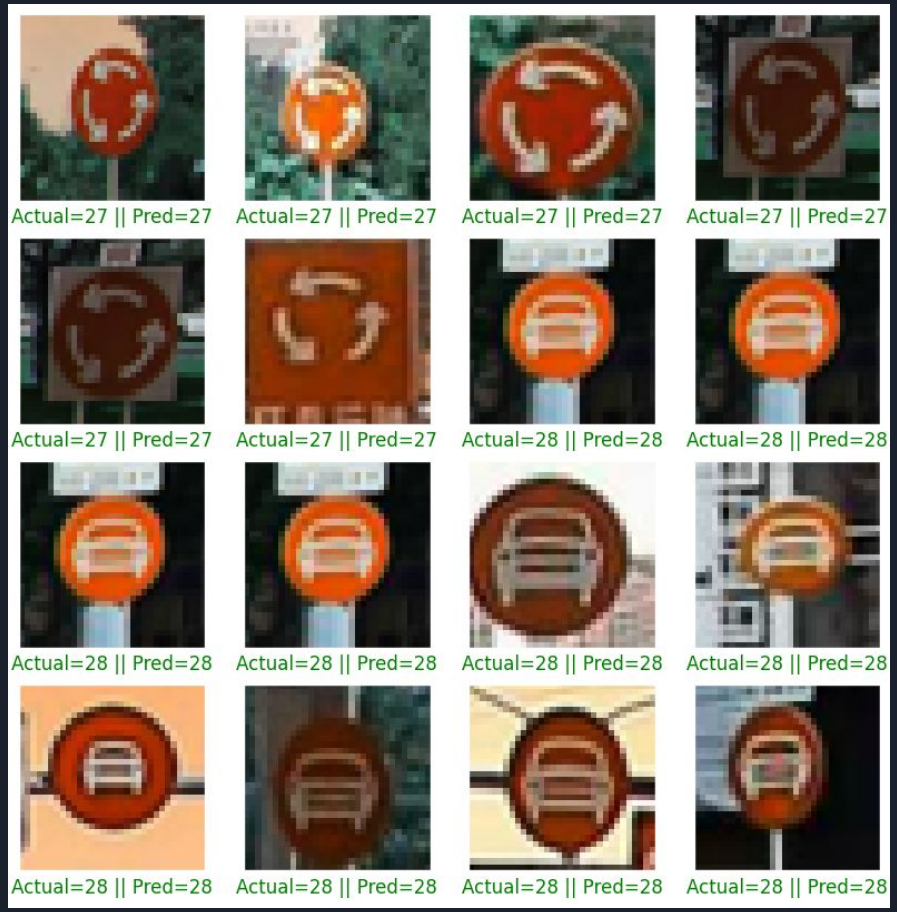
Image Pre-Processing

- Images Resized to 32x32 x3
- RGB Format
- Normalized Pixel Value: Pixel/255. (0 - 1)
  - Helps convergence in training

Image Diversity

- 10deg Random Rotation
- 15% Random Zoom
- 10% Horizontal Shift
- 10% Vertical Shift
- 15% Random Shearing

- Disabled flipping to ensure all traffic signs are read in the upright position
- Gaps in image are filled with values from the nearest pixel



Pre-Processed Images

Model Architecture

- Sequential Convolutional Neural Network
- Tensorflow / Keras
- Sequential API

- Total Parameters: 5,260,604
- Trainable Parameters: 1,753,086
- Non-Trainable Parameters: 1,344

```
Conv2D(filters=16, kernel_size=(3,3), activation='relu')
Conv2D(filters=32, kernel_size=(3,3), activation='relu')
MaxPool2D(pool_size=(2, 2))
BatchNormalization(axis=-1)
Conv2D(filters=64, kernel_size=(3,3), activation='relu')
Conv2D(filters=128, kernel_size=(3,3), activation='relu')
MaxPool2D(pool_size=(2, 2))
BatchNormalization(axis=-1)
Flatten()
Dense(512, activation='relu')
BatchNormalization()
Dropout(rate=0.50)
Dense(30, activation='softmax')
```

Layer (type)	Output Shape	Param #
conv2d_4 (Conv2D)	(None, 30, 30, 16)	448
conv2d_5 (Conv2D)	(None, 28, 28, 32)	4,640
max_pooling2d_2 (MaxPooling2D)	(None, 14, 14, 32)	0
batch_normalization_3 (BatchNormalization)	(None, 14, 14, 32)	128
conv2d_6 (Conv2D)	(None, 12, 12, 64)	18,496
conv2d_7 (Conv2D)	(None, 10, 10, 128)	73,856
max_pooling2d_3 (MaxPooling2D)	(None, 5, 5, 128)	0
batch_normalization_4 (BatchNormalization)	(None, 5, 5, 128)	512
flatten_1 (Flatten)	(None, 3200)	0
dense_2 (Dense)	(None, 512)	1,638,912
batch_normalization_5 (BatchNormalization)	(None, 512)	2,048
dropout_1 (Dropout)	(None, 512)	0
dense_3 (Dense)	(None, 30)	15,390

Layers and Parameters

Hyper Parameters

Learning Rate: 0.001

Optimizer: ADAM

Epochs: 30

Batch Size: 32

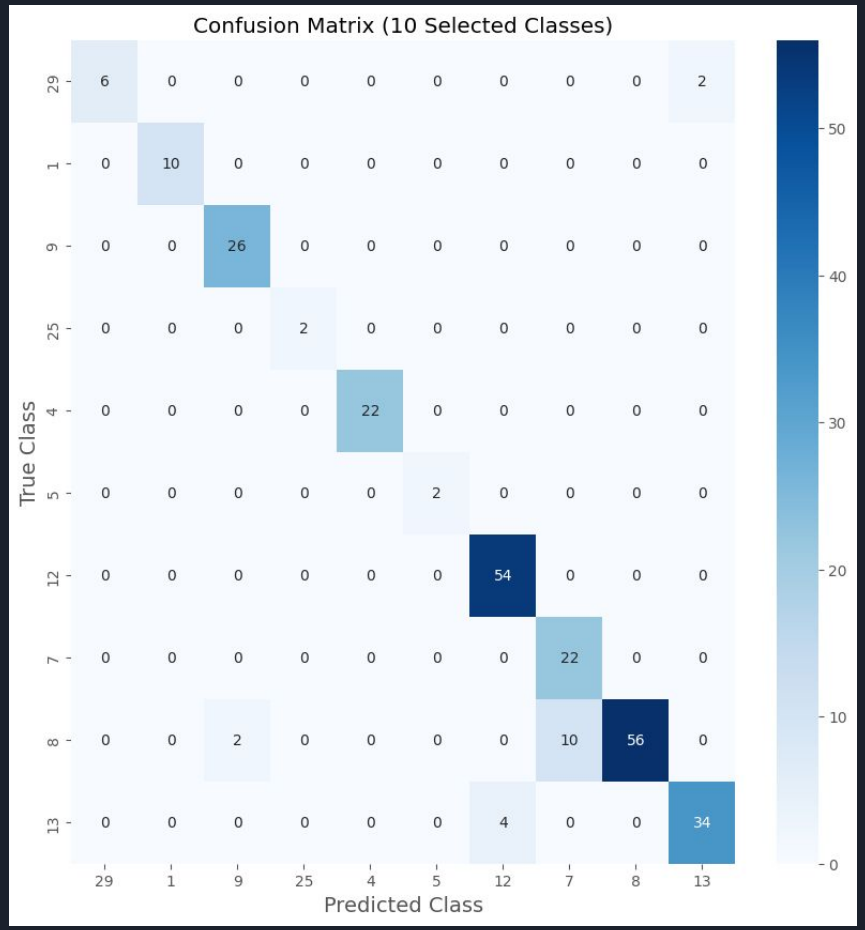
Layers: 13

- 4 Conv2D
- 2 Max Pool2D
- 3 Batch Normalization
- 1 Flatten
- 2 Dense
- 1 Dropout

Results

Test Data Accuracy: ~81%

	Precision	Recall	F1-Score
Macro	77%	80%	0.74
Weighted	88%	81%	0.81



selected_classes = ['29', '1', '9', '25', '4', '5', '12', '7', '8', '13']

Test Data Predictions starting at index: 675

How to Improve the Model?

- Larger and Higher Quality Dataset
- More Consistent Dataset
- More Robust Image Preprocessing

- Alternative:
  - Use a Pretrained Model (ResNet)

https://drive.google.com/file/d/1BZYhN4VqWjrFCABZzW-Hgy6xpT19xw7N/view?usp=drive_link