German traffic sign classification with 99,38% accuracy

Dataset description

Original 2012 dataset contains 50k images of 43 different road traffic signs types.

Images are sized from 15x15 pixels to 250x250 pixels according to GTSDB, but Udacity's version sized 32x32 pixels.

A photographer takes sign photos from inside of a car while it moves towards the sign.

First shots were taken from far away, so the resolution is low, and signs are 15x15 pixels small, blurry, and hard to recognize. An average person can read 98,84% of those signs, and a human best individual can read 99,22%.

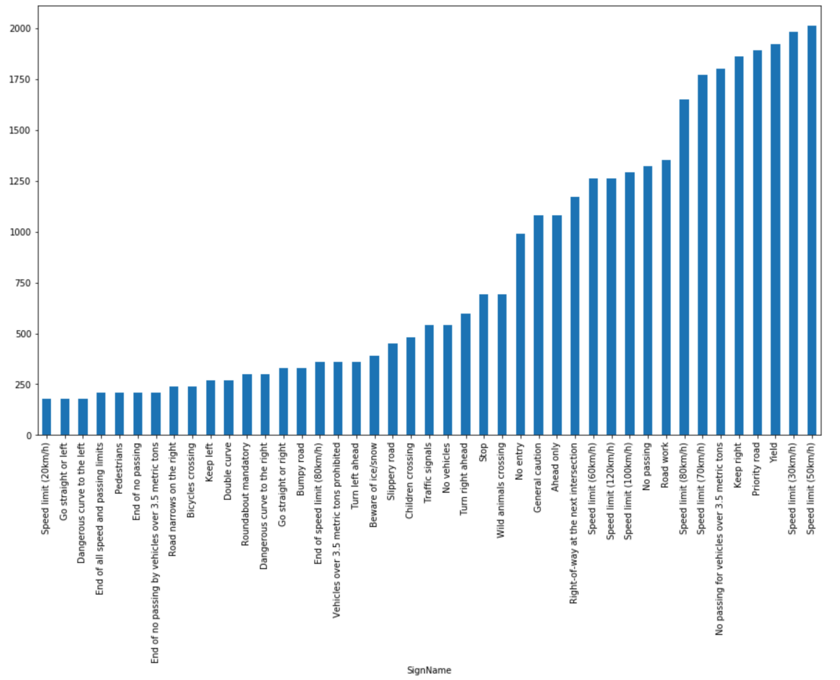
Example of low resolution image

By the way, if you think that 15x15 resolution is low for 2020, note that iPhone 11 Pro's camera with its 120-degree angle 4k main camera will see 60x60 cm standard traffic sign as 15x15 pixels picture from 25 meters distance. Note that 29 meters required to fully stop a car moving at 50 km/h, which is standard city speed in Germany (and

Europe), and this distance increases four times each time the speed doubles. This is the reason why self-driving cars use a bunch of wide and narrow viewing angle cameras.

The dataset contains 34k training examples, 4k validation examples, and 12k test examples.

Image distribution by sign is not even varying from less than 250 images per sign for «Speed limit 20 km/h» to more than 2k images per sign for «Speed limit 50 km/h».



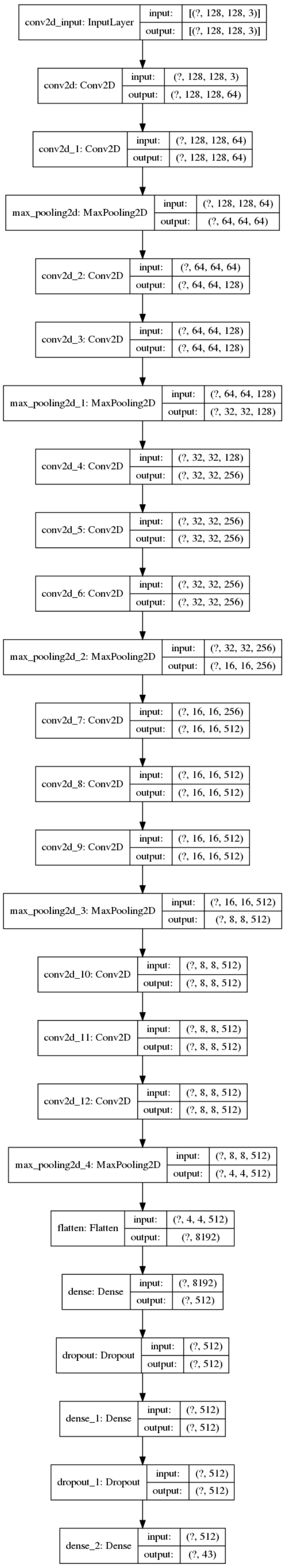
Quantity of images per class

Image distribution is not something you expect to see in Germany, e.g. «Wild animal crossing» traffic sign represented three times more frequently than «Bicycle crossing» - I have limited visitor's experience in Bavaria, Germany. However, I still remember lots of bicycles and don't remember wild animals, that's why I calculated class weights for further model training.

Model accuracy without augmentation

Having this dataset, I was able to achieve a 95% accuracy (which is below human average) with the VGG-16 network which structure depicted below.

Model accuracy without augmentation



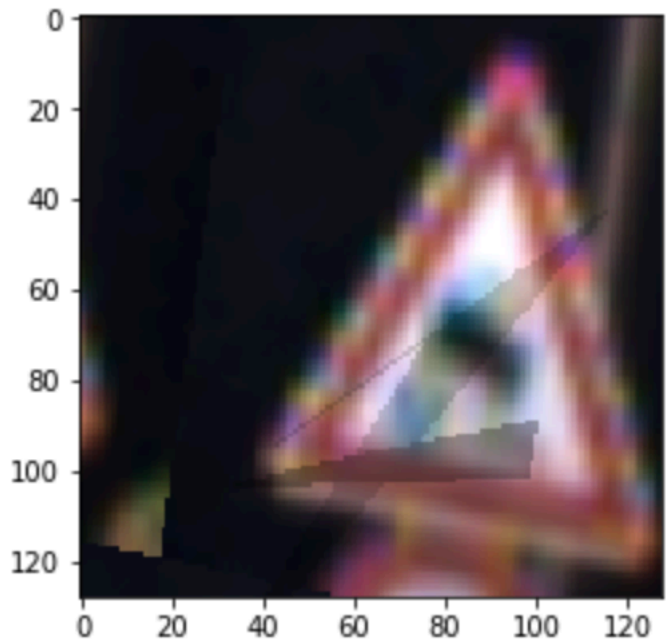
Model accuracy without augmentation

I decided to increase the number of training images by augmenting them.

Data augmentation and hyperparameter tuning

I've studied training and validation examples by eye and found that signs inside image have different position, size, inclination, brightness, there are shadows and sun flares.

Typical augmented image looks like one below. Gray triangles stand for shadows and in case you doubt their usefulness they increase final model accuracy by 1,5%.

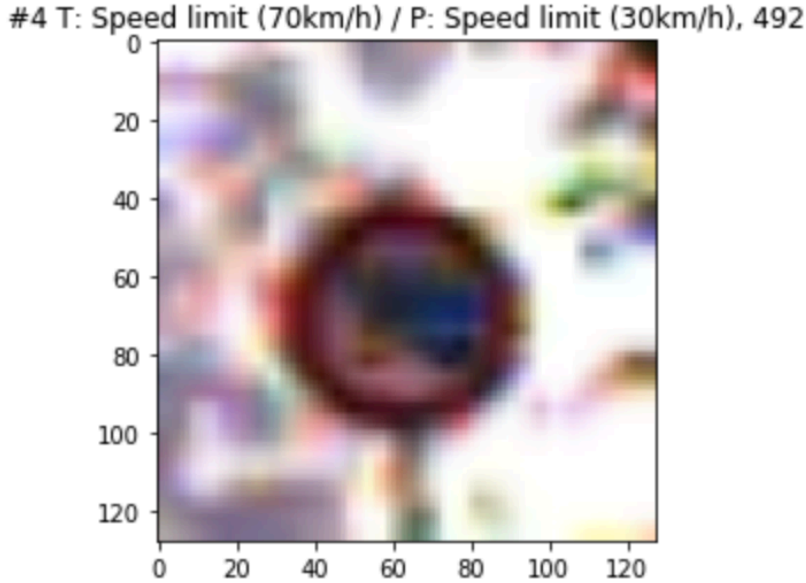


Augmented image example

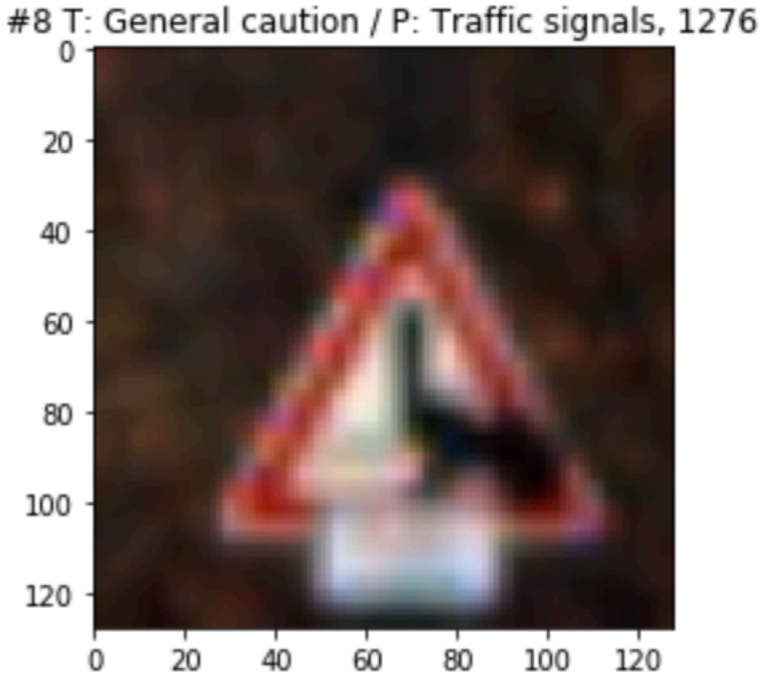
Final model with RMSProp optimizer achieved 99,38% accuracy.

I can label correctly 72% of signs that model classifies incorrectly.

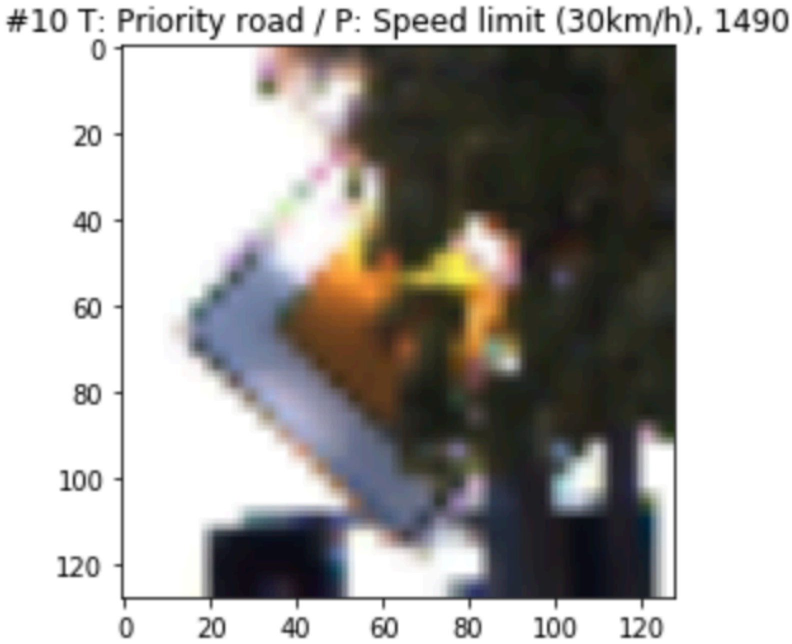
Model does not handle well blurry signs, «General caution» vs «Traffic signs», partially covered signs, heavy shadows and heavy sun flares. One image in test dataset was labeled incorrectly as «80 km/h» while it is clearly seen as «60 km/h». I believe everything except blurry signs can be fixed by providing more examples.



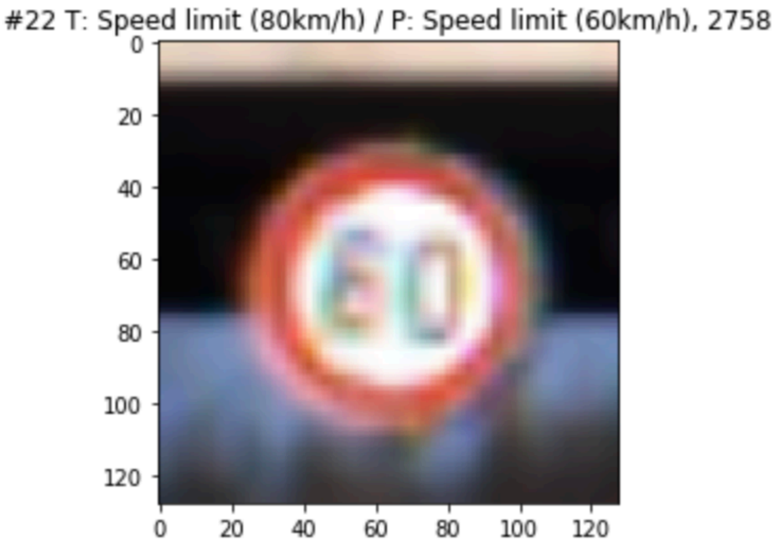
Example of blurry image in test dataset



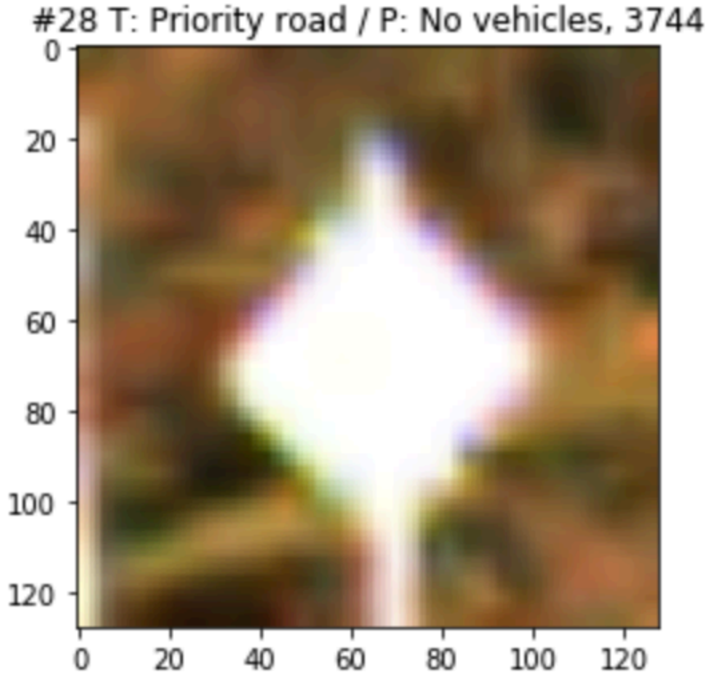
Example of mislabeling "General caution" with "Traffic lights"



Example of partially covered sign



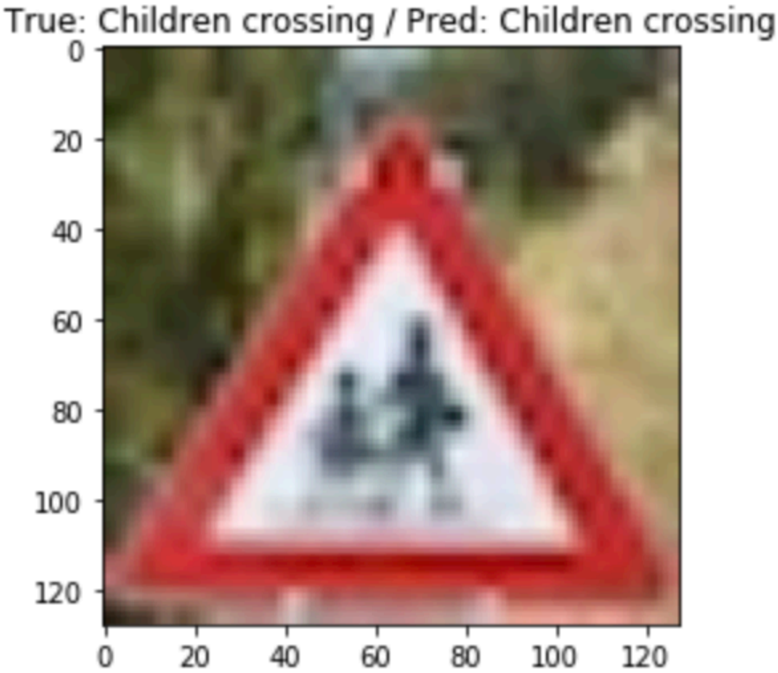
Example of mislabeling in test dataset



Example of sunflares in test dataset

Testing on images found in Internet

Model was able to recognize 5 of 5 images I found in internet, even blurry one.

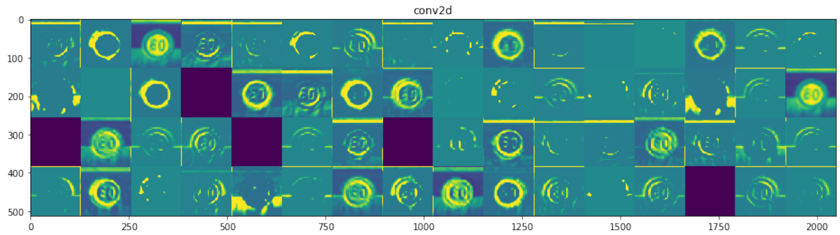


Blurry image recognized as "Children crossing".

Visualization of model's layers

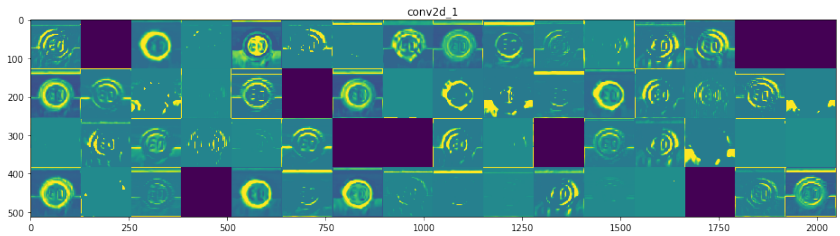
That's interesting how model sees 60 km/h sign.

1st convolution layer shows sign itself and ignores everything outside.



1st convolution layer

2nd convolution layer detects edges.



2nd convolution layer

Conclusion

Model was able to achieve 99,38% accuracy for German Traffic Sign Dataset which is better than human best (99,22%) and almost as good as 2012 winner (99,46%).