Deep Convolutional Networks

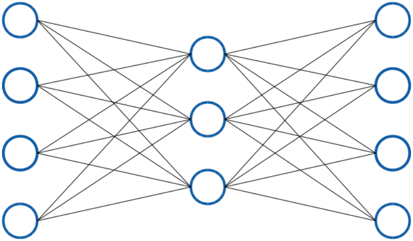
Christian Andreas Mielers
Phil Yannick Schrör

Ruhr-University Bochum
Institute for Neural Computation
Study Project

24th of February 2016

Convolutional Neural Networks

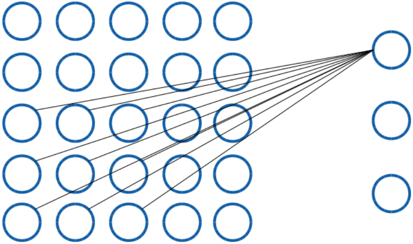
- Learns the weights of convolutional filters
- Exploits spatial structure in the input
- Convolving entire input with filter implies shared weights
- Reduced amount of weights allows lots of filters
- Filters specific to color channels



Convolutional Neural Networks

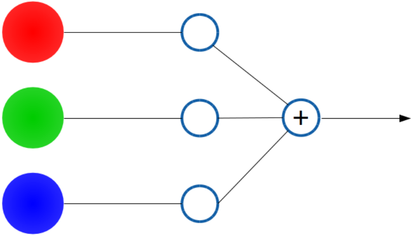
RUB

- Learns the weights of convolutional filters
- Exploits spatial structure in the input
- Convolving entire input with filter implies shared weights
- Reduced amount of weights allows lots of filters
- Filters specific to color channels



Convolutional Neural Networks

- Learns the weights of convolutional filters
- Exploits spatial structure in the input
- Convolving entire input with filter implies shared weights
- Reduced amount of weights allows lots of filters
- Filters specific to color channels



Network Structure

RUB

Layer	Type	Configuration	Activation function
0	Convolutional	100 filters of size 7×7 per channel	tanh
1	Max Pooling	Pool size 2×2	-
2	Convolutional	150 filters of size 4×4 per channel	tanh
3	Max Pooling	Pool size 2×2	-
4	Convolutional	250 filters of size 4×4 per channel	tanh
5	Max Pooling	Pool size 2×2	-
6	Dense	300 neurons	tanh
7	Dense	43 neurons	softmax

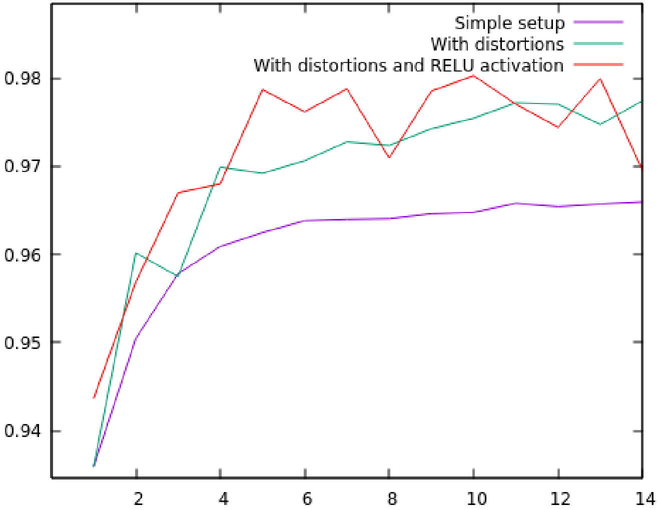
German Traffic Sign Recognition Benchmark

- German Traffic Sign Recognition Benchmark
- What is the task?
- Show some images

Simple Setup

- Describe Simple Setup
- Present Results

Results on GTSRB



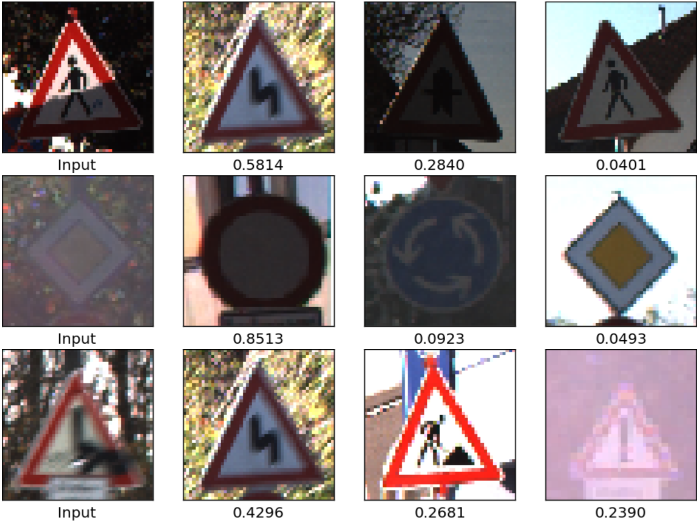
Input Distortions

- Mention input distortions
- Explain them
- Present distortion parameters
- Maybe add one or two images before and after the transformations

Results with RELU

- Add RELU image
- Present results with RELU activation function

Missclassified images



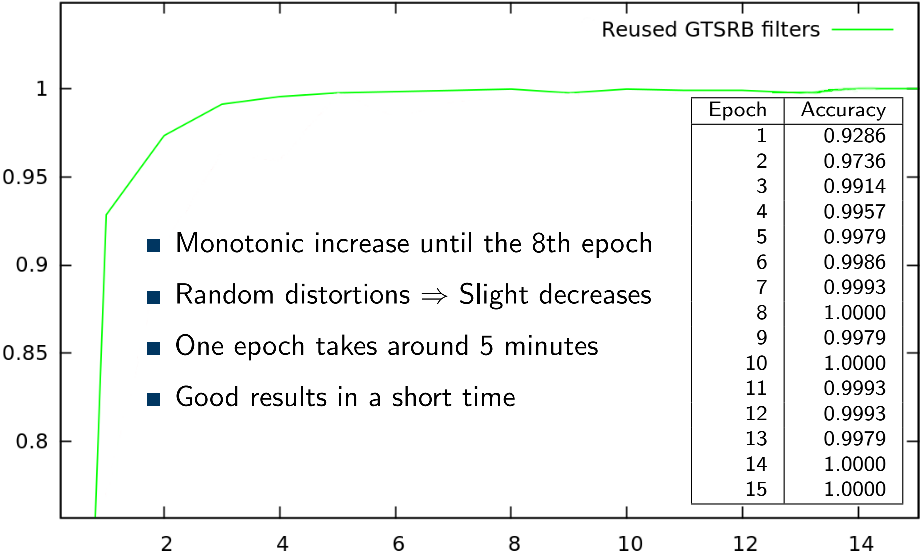
Filter Reuse

- How well do the GTSRB filters generalize?
- Initialize new network with same structure randomly
- Copy GTSRB filters to the new network
- Train only the fully connected layers!

COIL100

- Columbia Object Image Library 100 $\Rightarrow$ COIL100
- 100 different objects
- Objects turning on a black turntable
- One photo each time the object has turned by 5°
- 72 images per object, 7200 images in total
- Random separation into 58 training and 14 test images per object

COIL100 — GTSRB Filters Results



- Monotonic increase until the 8th epoch
- Random distortions $\Rightarrow$ Slight decreases
- One epoch takes around 5 minutes
- Good results in a short time

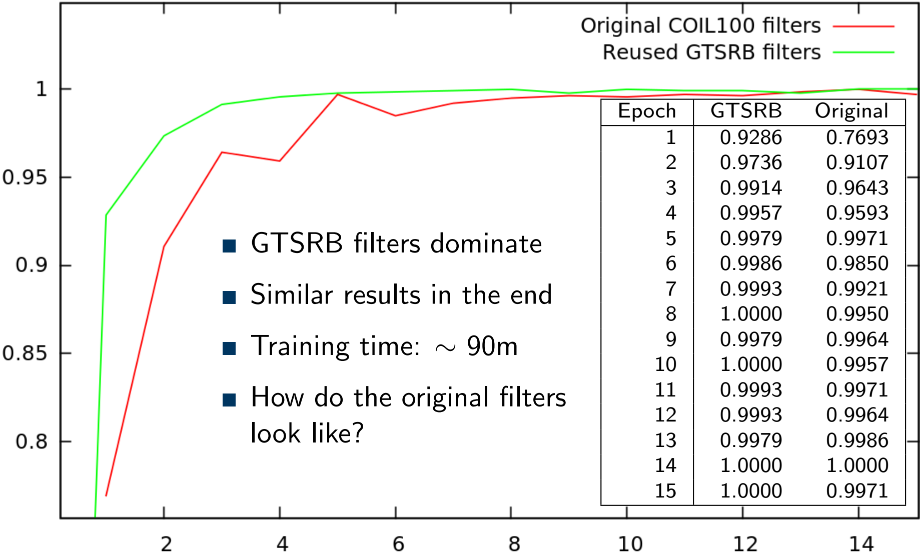
Epoch	Accuracy
1	0.9286
2	0.9736
3	0.9914
4	0.9957
5	0.9979
6	0.9986
7	0.9993
8	1.0000
9	0.9979
10	1.0000
11	0.9993
12	0.9993
13	0.9979
14	1.0000
15	1.0000

COIL100 — Original filters

- Which advantages does this approach have?
- We need data for a comparison
- Train a new network conventionally on COIL100
- Call the filters *original*, which are created this way
- Compare training time and results!

COIL100 — Original Filters Results

RUB

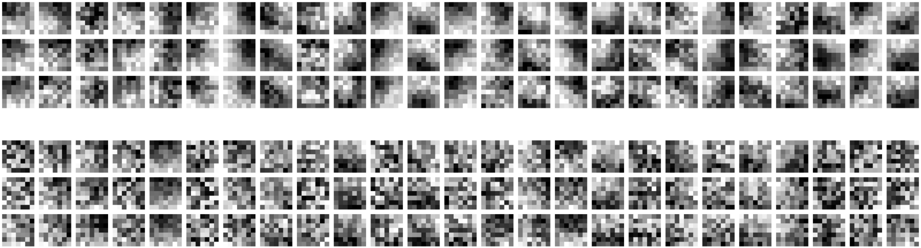


- GTSRB filters dominate
- Similar results in the end
- Training time: $\sim$ 90m
- How do the original filters look like?

Epoch	GTSRB	Original
1	0.9286	0.7693
2	0.9736	0.9107
3	0.9914	0.9643
4	0.9957	0.9593
5	0.9979	0.9971
6	0.9986	0.9850
7	0.9993	0.9921
8	1.0000	0.9950
9	0.9979	0.9964
10	1.0000	0.9957
11	0.9993	0.9971
12	0.9993	0.9964
13	0.9979	0.9986
14	1.0000	1.0000
15	1.0000	0.9971

COIL100 — How do the original filters look like?

RUB



- The spatial structure is not as distinctive as the one of the GTSRB filters
- One cannot assume a good generalization behavior of the COIL100 filters
- Maybe, the CNN is too oversized for the task
- The original filters exhibit more differences between the color channels
- Long training time, but probably overfitted filters

INRIA

- Describe INRIA dataset
- Show image
- Show results with reused filters
- Show results with original filters

Conclusion

- Summarize results

Questions?

Questions?