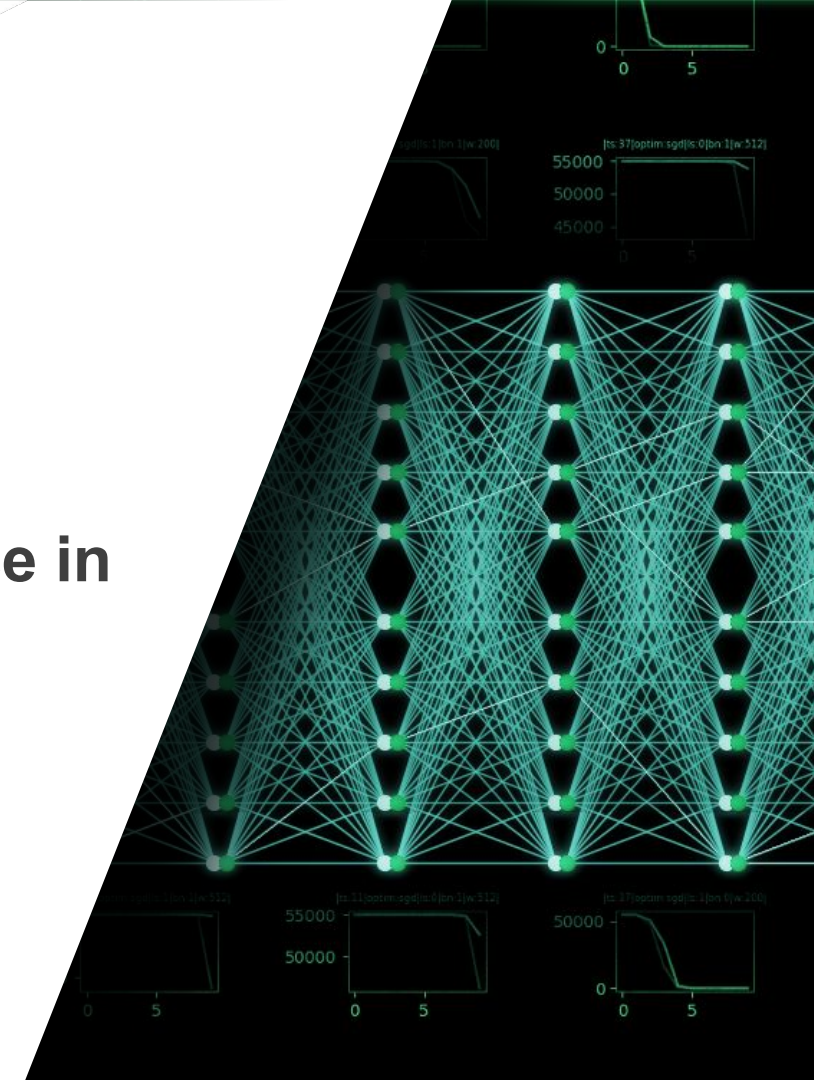
Stride and translation invariance in CNNs

Presenter: Coenraad Mouton
Co-authors: Christiaan Myburgh and Prof. Marelie H Davel

Multilingual Speech Technologies, North-West University
Centre for Artificial Intelligence Research (CAIR)



Overview

- Provide a theoretical overview of translation invariance and what contributes to it in CNNs
- Purpose a novel perspective surrounding stride and filtering
- Empirically test this theoretical perspective

Translation Invariance

- If a model’s output is unaffected by shifts of the input
- Commonly incorrectly assumed that CNNs are invariant to translation

Translation Equivariance

- While the convolution operation is not *invariant*, it can be *equivariant.*
- Equivariance - A shift of the input results in an equal shift of the output

Translation Equivariance

- Both pooling and convolution is translational in nature
- Intuitively equivariance should hold
- Consider an arbitrary input **I** and kernel **K**

I = [0,0,0,0,1,2,0,0,0,0] , K[n] = [1,0,1]

	Input	**I * K**
Input (untranslated)	[0,0,0,0,1,2,0,0,0,0]	[0,0,1,2,1,2,0,0]
Shifted by one	[0,0,0,0,0,1,2,0,0,0]	[0,0,0,1,2,1,2,0]
Shifted by two	[0,0,0,0,0,0,1,2,0,0]	[0,0,0,0,1,2,1,2]

Translation Equivariance

- Where does it fail in CNNs?
- Equivariance holds for dense pooling and convolution: stride = 1
- Subsampling (stride>1) breaks equivariance
- Consider previous example with a stride of 2

$I = [0,0,0,0,1,2,0,0,0,0]$, $K[n] = [1,0,1]$

	Input	**I * K**
Input (untranslated)	[0,0,0,0,1,2,0,0,0,0]	[0,1,1,0]
Shifted by one	[0,0,0,0,0,1,2,0,0,0]	[0,0,2,2]
Shifted by two	[0,0,0,0,0,0,1,2,0,0]	[0,0,1,1]

Shiftability

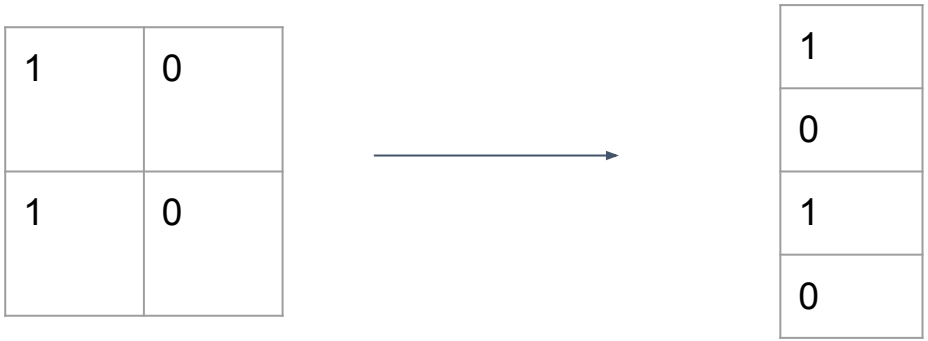
- Stride does not completely break equivariance
- Subsampling can ‘scale’ shifts - if factors of the subsampling factor

I = [0,0,0,0,1,2,0,0,0,0] , K[n] = [1,0,1]

	Input	**I * K**
Input (untranslated)	[0,0,0,0,1,2,0,0,0,0]	[0,1,1,0]
Shifted by one	[0,0,0,0,0,1,2,0,0,0]	[0,0,2,2]
Shifted by two	[0,0,0,0,0,0,1,2,0,0]	[0,0,1,1]

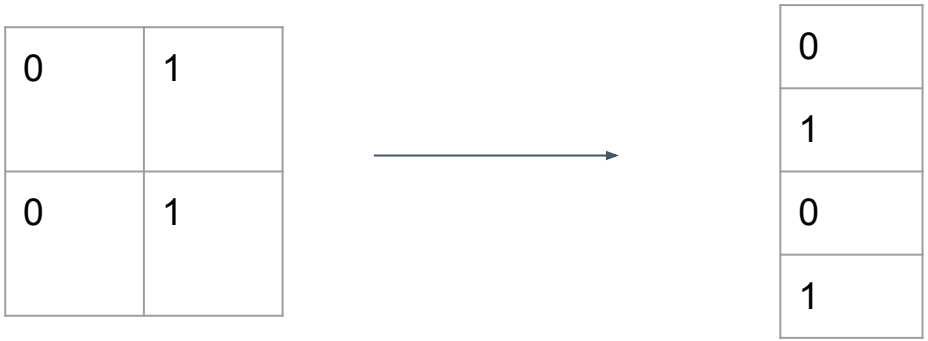
How does this relate to invariance?

- Fully equivariant
- Conv and pooling output is equivalent for translations
- Shift still occurs in the fully connected layer



How does this relate to invariance?

- Fully equivariant
- Conv and pooling output is equivalent for translations
- Shift still occurs in the fully connected layer



Signal Movement and Signal Similarity

- Signal Similarity - How much of the untranslated signal's output is preserved after translation

- Signal Movement - How far the translated output has moved from the original position of the untranslated output

Local Homogeneity

- How can we preserve signal similarity and reduce signal movement?
- Subsampling disregards intermediary samples
- Signal similarity can be preserved when subsampling if input is homogenous in accordance with the subsampling factor

Input: 0 0 0 0 0 0 2 2 3 3 1 1 2 2 0 0 0 0 0 0

Shift	**Subsampled Output**
0	0 0 0 2 3 1 2 0 0
1	0 0 0 2 3 1 2 0 0
2	0 0 0 0 2 3 1 2 0
3	0 0 0 0 2 3 1 2 0

Pooling/Filtering

- Pooling reduces the variance of a given input
- Provides more similarity between neighbouring pixels

- Strided pooling:
  - Improves signal similarity
  - Reduces signal movement by subsampling
  - Results in greater translation invariance

Measuring translation invariance

- Entire test set is randomly translated up to a maximum shift
- Translated and untranslated test set is passed through model
- Each sample’s two 10-dimensional output vectors are compared
- Compared using cosine similarity
- Average cosine similarity is taken across all samples

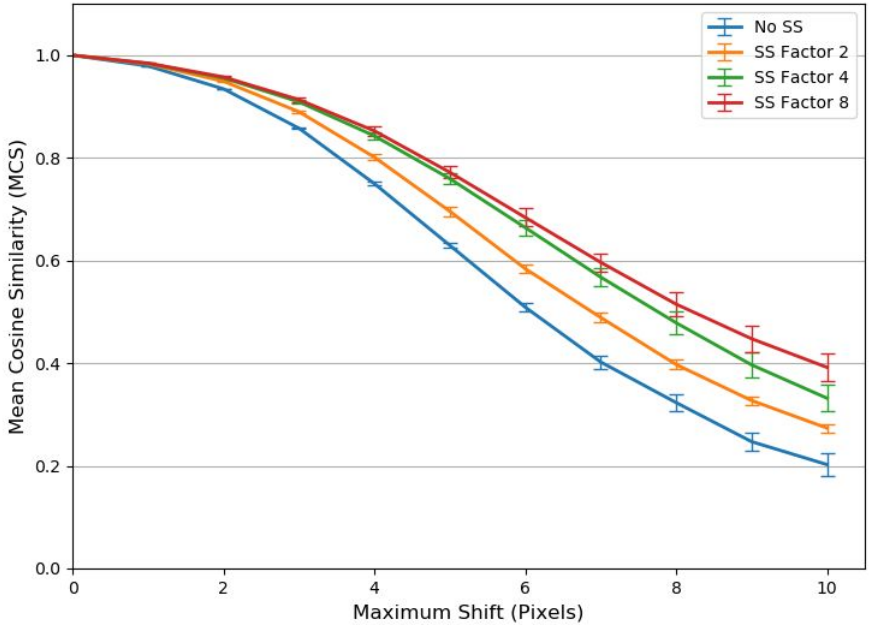
Experimental Setup

- Input zero padded to 40x40
- Four 3-Layer CNNs trained on MNIST data set
- Hyperparameters are optimized - converge to 100% train accuracy
- Besides early stopping - no regularization

MNIST Architecture

Layer	Size	Stride
Conv	3x3	1
Pool	2x2	1/2
Conv	3x3	1
Pool	2x2	1/2
Conv	3x3	1
Pool	2x2	1/2

Results - MNIST



Results - CIFAR10

- 16 three layer CNNs are trained
- Varying kernel size and subsampling factor
- Input is padded to 50x50

Subsampling Factor	Kernel Size			
	2x2	3x3	4x4	5x5
1	0.630	0.598	0.595	0.618
2	0.554	0.635	0.683	0.731
4	0.622	0.674	0.759	0.789
8	0.610	0.660	0.762	0.791

Results - CIFAR10

- 16 three layer CNNs are trained
- Varying kernel size and subsampling factor
- Input is padded to 50x50

Subsampling Factor	Kernel Size			
	2x2	3x3	4x4	5x5
1	0.630	0.598	0.595	0.618
2	0.554	0.635	0.683	0.731
4	0.622	0.674	0.759	0.789
8	0.610	0.660	0.762	0.791

Results - CIFAR10

- 16 three layer CNNs are trained
- Varying kernel size and subsampling factor
- Input is padded to 50x50

Subsampling Factor	Kernel Size			
	2x2	3x3	4x4	5x5
1	0.630	0.598	0.595	0.618
2	0.554	0.635	0.683	0.731
4	0.622	0.674	0.759	0.789
8	0.610	0.660	0.762	0.791

Results - CIFAR10

- 16 three layer CNNs are trained
- Varying kernel size and subsampling factor
- Input is padded to 50x50

Subsampling Factor	Kernel Size			
	2x2	3x3	4x4	5x5
1	0.630	0.598	0.595	0.618
2	0.554	0.635	0.683	0.731
4	0.622	0.674	0.759	0.789
8	0.610	0.660	0.762	0.791

Results - Generalization

- Improved generalization for some subsampling and filtering
- Too much filtering or subsampling decreases generalization
- Slight trade-off between invariance and generalization

Subsampling Factor	Kernel Size			
	2x2	3x3	4x4	5x5
1	72.33	75.00	76.10	76.00
2	74.43	77.00	77.57	76.69
4	73.94	76.72	77.25	76.76
8	72.53	75.31	76.69	75.95

Conclusion

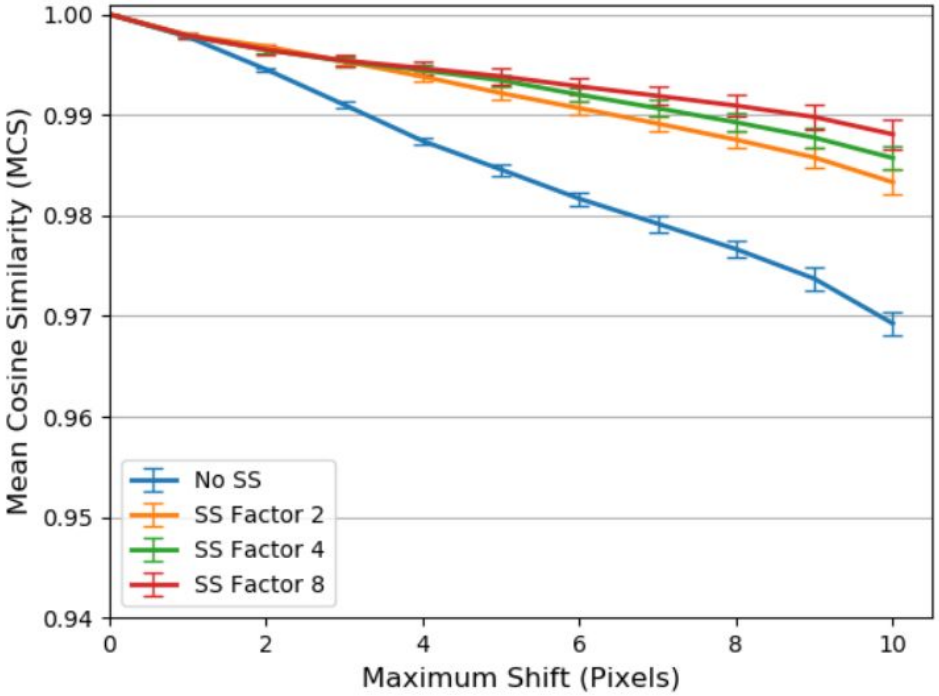
- Subsampling, when combined with sufficient filtering, improves translation invariance in CNNs
- Too much subsampling and/or filtering reduces test accuracy
- Several other things also tested, such as data augmentation, anti-aliasing filters, and global average pooling.

Thank you

Results - Learned Invariance

- Does this pattern hold when trained on a translated train set?
- Train set samples randomly translated up to 8 pixels before training
- Translation invariance is measured in the same way

Results - Learned Invariance



Results - Anti-aliasing

Subsampling Factor	AA	MNIST	CIFAR
1	No	0.248	**0.630**
	Yes	**0.329**	0.518
4	No	0.383	0.620
	Yes	**0.654**	**0.710**
8	No	0.447	0.611
	Yes	**0.638**	**0.690**

Translation Equivariance