

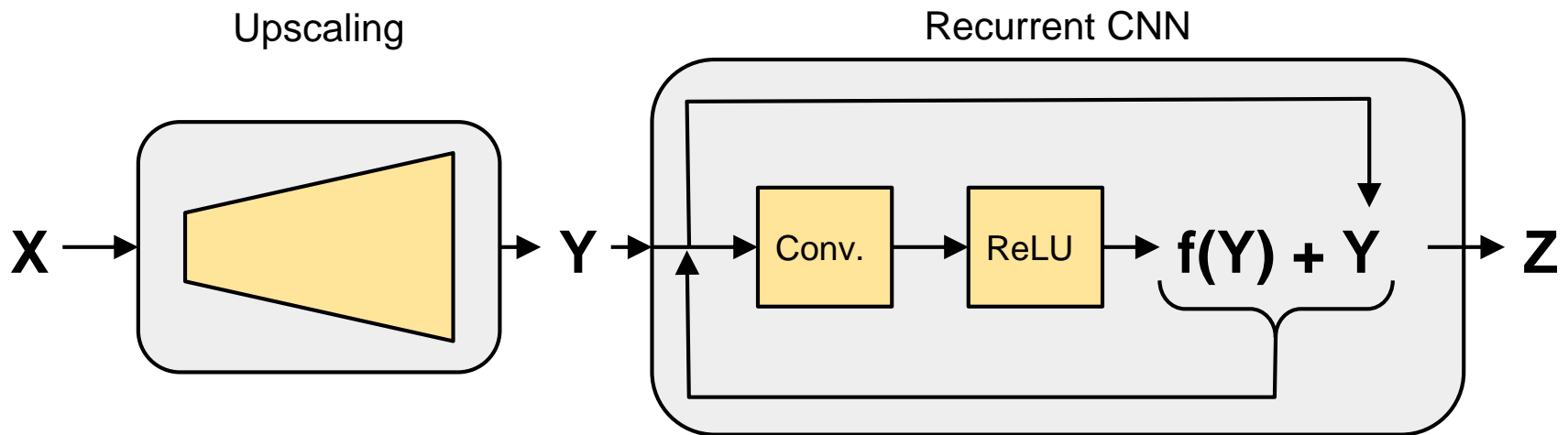
# Super Resolution

## Results, Critical Analysis and Conclusion

Vivianne Tanner  
Florian Jörg  
Adrian Wälchli

# Our Proposed Solution

- X: Low resolution input
- Y: Intermediate upscaling result
- Z: High resolution output



# Output

- Loss after each epoch
  - Training set
  - Validation set
- Gradient norm
- Processed images for test set
- Metrics for comparison
  - PSNR
  - SSIM
- Trained network model

# Baseline

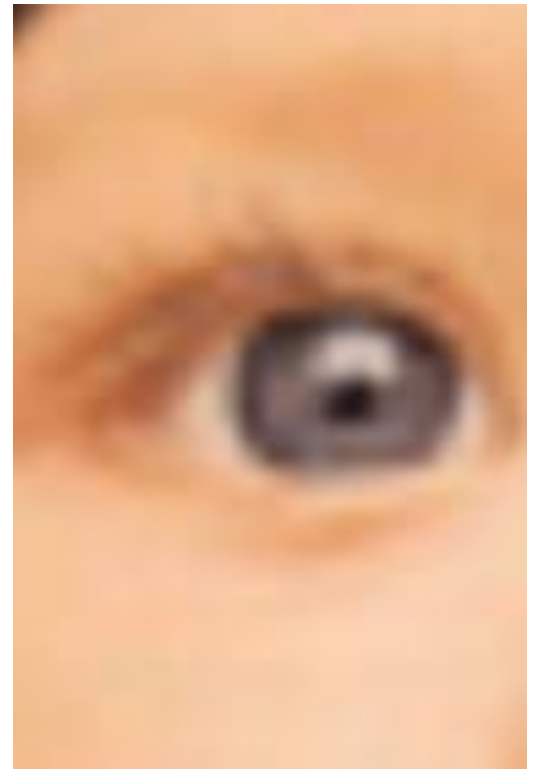
Upscale factor 4



Nearest



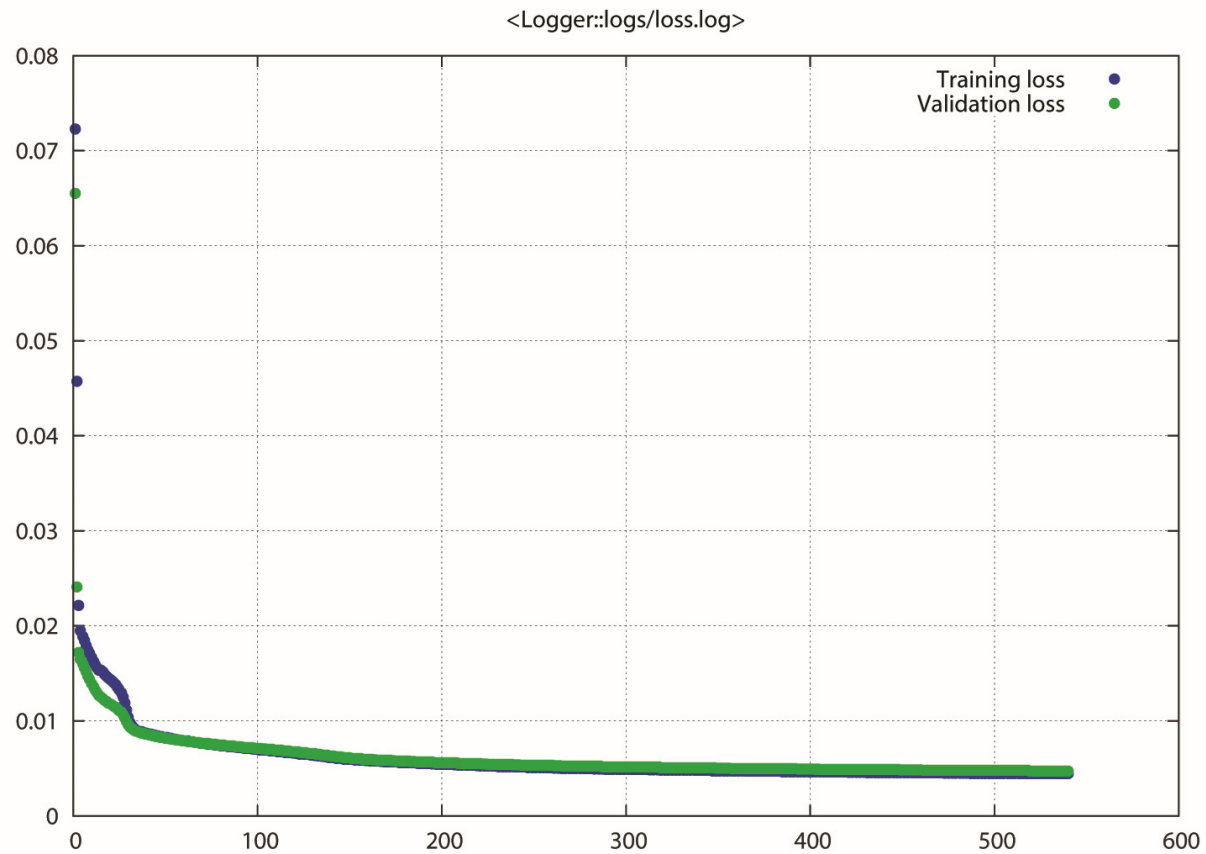
Bilinear



Bicubic

# Full Network

Euclidean loss vs. epoch



# Full Network

Test set: Set5



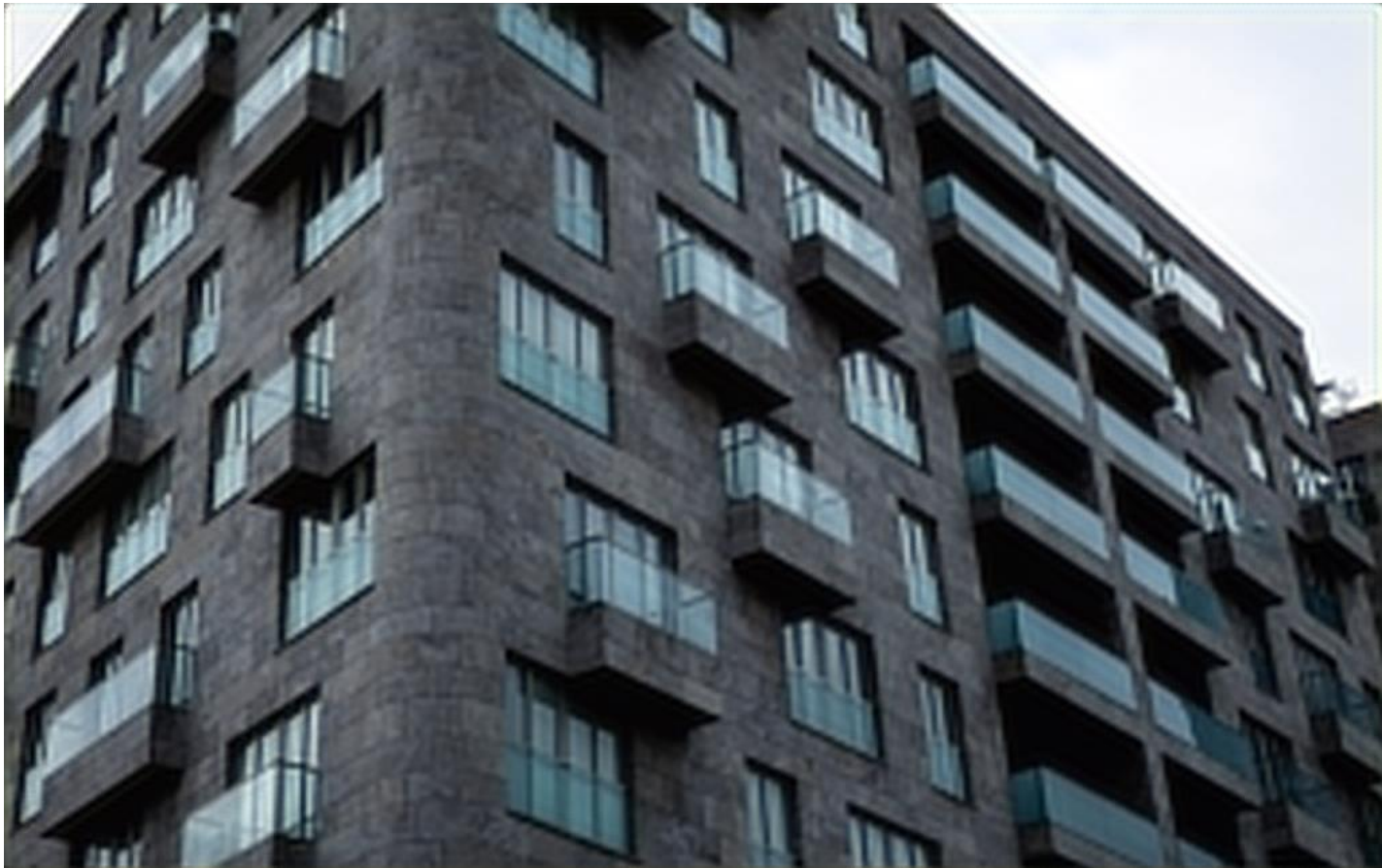
Ground Truth



Output

# Full Network

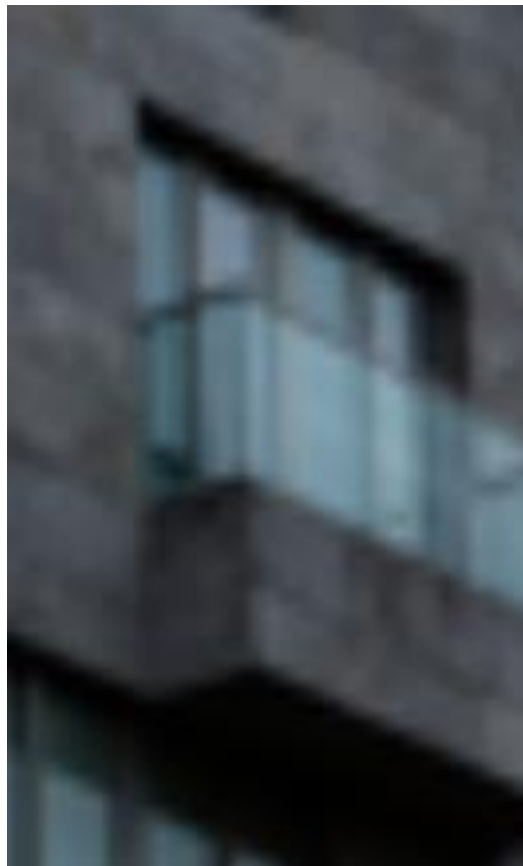
Different test set: Urban100



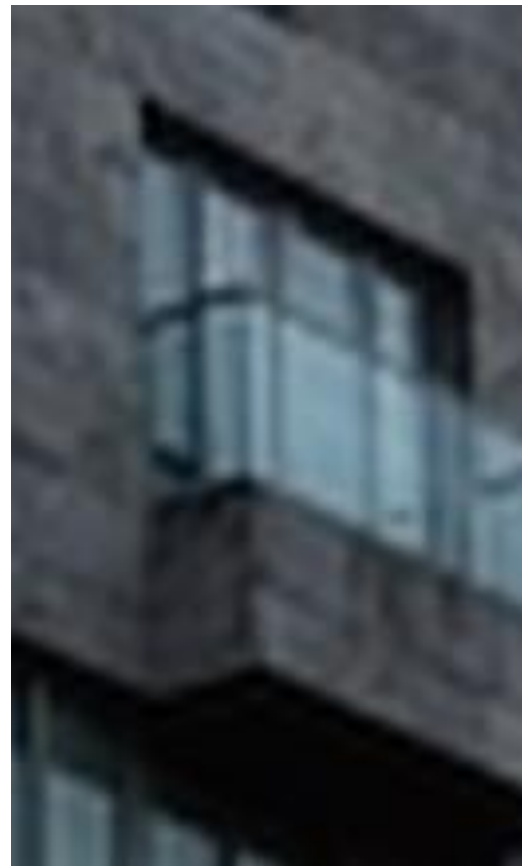
# Full Network



Ground Truth



Bilinear



Ours



# Full Network

Change of domain: Nature/Faces to Urban

	Nearest Set5	Bicubic Set5	Ours Set5	Ours Urban100
Avg. PSNR	24.607...	24.692...	25.469...	21.540...
Avg. SSIM	0.742...	0.773...	0.793...	0.641...
Loss	-	-	0.00364...	0.00881...

Dataset	Scale	Bicubic PSNR/SSIM	A+ [29] PSNR/SSIM	SRCNN [5] PSNR/SSIM	RFL [23] PSNR/SSIM	SelfEx [10] PSNR/SSIM	DRCN (Ours) PSNR/SSIM
Set5	×2	33.66/0.9299	36.54/0.9544	36.66/0.9542	36.54/0.9537	36.49/0.9537	37.63/0.9588
	×3	30.39/0.8682	32.58/0.9088	32.75/0.9090	32.43/0.9057	32.58/0.9093	33.82/0.9226
	×4	28.42/0.8104	30.28/0.8603	30.48/0.8628	30.14/0.8548	30.31/0.8619	31.53/0.8854
Set14	×2	30.24/0.8688	32.28/0.9056	32.42/0.9063	32.26/0.9040	32.22/0.9034	33.04/0.9118
	×3	27.55/0.7742	29.13/0.8188	29.28/0.8209	29.05/0.8164	29.16/0.8196	29.76/0.8311
	×4	26.00/0.7027	27.32/0.7491	27.49/0.7503	27.24/0.7451	27.40/0.7518	28.02/0.7670
B100	×2	29.56/0.8431	31.21/0.8863	31.36/0.8879	31.16/0.8840	31.18/0.8855	31.85/0.8942
	×3	27.21/0.7385	28.29/0.7835	28.41/0.7863	28.22/0.7806	28.29/0.7840	28.80/0.7963
	×4	25.96/0.6675	26.82/0.7087	26.90/0.7101	26.75/0.7054	26.84/0.7106	27.23/0.7233
Urban100	×2	26.88/0.8403	29.20/0.8938	29.50/0.8946	29.11/0.8904	29.54/0.8967	30.75/0.9133
	×3	24.46/0.7349	26.03/0.7973	26.24/0.7989	25.86/0.7900	26.44/0.8088	27.15/0.8276
	×4	23.14/0.6577	24.32/0.7183	24.52/0.7221	24.19/0.7096	24.79/0.7374	25.14/0.7510

# Omitting Recurrent Part

Does our “refinement step” help or hurt?

	Ours Set5	Ours Urban100	No Rec Set5	No Rec Urban
Avg. PSNR	25.469...	21.540...	26.187...	21.862...
Avg. SSIM	0.793...	0.641...	0.811...	0.660...
Loss	0.00364...	0.00881...	0.00310...	0.00828...

Previously:

5 recursions

6 hidden channels in recurrent conv.

# What is the Problem?

- Remove residual connection
  - Bad performance
- Train only recurrent part, keep the rest fixed
  - Minor difference
- More channels in recurrent conv.
  - Training time too high
- Deeper, more recursions
  - Training time too high

# Experiments > Time

# Problems and Challenges

- Choice of hyperparameters
- Artifacts
  - Zero-padding
  - Ringing
- Resource usage
- Mismatching metrics
  - Baseline metrics do not match Kim et al.

# Conclusion

- Better than naive methods
- Worse than state of the art
- Learning outcomes:
  - Upscaling with convolutions
  - Recurrent neural networks
  - Perceptual metrics
  - Efficient workflow on cluster
- Plug and play: Modular implementation

Thanks to **group 2** for  
flexibility with time slots