Department of Electrical Engineering, University of North Texas

Efficient Convolutional Neural Networks in Image Processing Applications

Nicholas Chiapputo
Colleen P. Bailey, Major Professor
Parthasarathy Guturu, Committee Member
Kamesh Namuduri, Committee Member

June 27, 2022

Outline

Motivation

Current State of the Art in Image Processing

- Models employ edge-based, residual, attention blocks, and transformers.
- Performs well, but requires hundreds of thousands to millions of parameters – not practical for implementation!
- Increase in popularity of edge-device machine learning (TinyML) necessitates improvements in memory and computation efficiency.

Objectives

- Reduce number of parameters and maintain image quality.
- Reduce memory requirements by removing parallel computation paths.
- Improve computational efficiency with fewer parameters, layers, and filters.

Model Design

UNT ELECTRICAL ENGINEERING

Approach

- Use a fully convolutional neural network for arbitrary input size.
- Use depthwise convolutional layers to reduce parameters.
- Use Parametric Rectified Linear Unit (PReLU) to improve metrics with fewer parameters.
- Use a pixel shuffling upsampling method to improve parameter/accuracy trade-off efficiency.

Memory-Efficient Single-Image Super-Resolution

Architecture

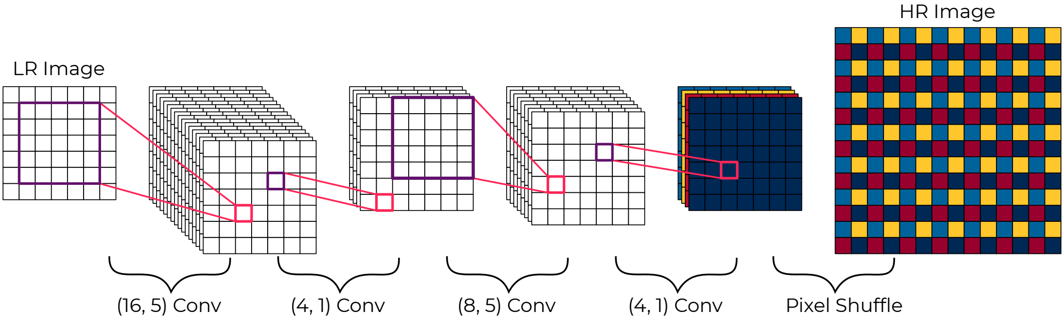


Figure: Model architecture of the **Tiny P**ixel **S**huffling **S**uper-**R**esolution (TinyPSSR) model.

Architecture: Parameter Breakdown

Layer	Parameters
(16,5) Convolution	416
PReLU	16
(4,1) Convolution	68
PReLU	4
(8,5) Convolution	808
PReLU	8
(4,1) Convolution	36
Pixel Shuffle	0
Total	**1,356**

Table: Parameters per layer and activation function of TinyPSSR.

Data

Train

- 800 images from DIV2K [1] training split. Sizes range from 648x2040 to 2040x2040.
- Pre-Processing:
  - Split into 64x64 patches
  - Downscale patches to 32x32
  - Split RGB channels
- Total of 522,939 training RGB images

Validation

- 1,000 images from ILSVRC [2] validation set of size 500x500
- Pre-Processing:
  - Downscale patches to 250x250
  - Split RGB channels

Test

- Mixed color and grayscale images
- Datasets:
  - Set5 [3] (five images)
  - Set14 [4] (14 images)
  - BSD100 [5] (100 images)
  - Urban100 [6] (100 images)
- Tested at 2x and 4x super-resolution

Training and Metrics

Training

- Loss Function
  - Adam Optimizer
  - Learning Rate: 2x10^{-4}
  - $\beta_1 = 0$
  - $\beta_2 = 0.9$
- Loss Metric
  - SSIM (calculated as $1-$SSIM)
- Training stopped after 50 epochs of no improvement in validation loss metric

Metrics

- Measure SSIM and PSNR
- Calculated on Y-channel (luminance) of the YCbCr color space

Hyperparameter Evaluation

Overview

- Activation Function
- Loss Metric
- RGB vs Grayscale
- 2x vs 4x scale training

Hyperparameter Evaluation
Activation Function

Activation	Set5		Set14		Urban100		BSD100	
	2x	4x	2x	4x	2x	4x	2x	4x
GCU+PReLU	**0.9492**	**0.8517**	**0.9015**	**0.7502**	**0.8897**	**0.7146**	**0.8814**	**0.7115**
GCU	<u>0.9486</u>	<u>0.8505</u>	0.9002	<u>0.7479</u>	<u>0.8875</u>	<u>0.7116</u>	0.8798	<u>0.7093</u>
PReLU	0.9483	0.8485	<u>0.9004</u>	0.7474	0.8865	0.7091	<u>0.8802</u>	0.7092
LReLU $\alpha = 0.50$	0.9482	0.8468	0.8997	0.7460	0.8858	0.7077	0.8792	0.7078
ReLU	0.9479	0.8462	0.8993	0.7455	0.8853	0.7078	0.8794	0.7087
LReLU $\alpha = 0.10$	0.9478	0.8451	0.8997	0.7455	0.8856	0.7074	0.8796	0.7082
LReLU $\alpha = 0.20$	0.9472	0.8426	0.8995	0.7440	0.8849	0.7056	0.8790	0.7071
LReLU $\alpha = 0.75$	0.9472	0.8414	0.8995	0.7439	0.8839	0.7046	0.8797	0.7079
None	0.9411	0.8254	0.8938	0.7321	0.8723	0.6873	0.8739	0.6988

Table: SSIM results for various activation functions ordered by descending Set5 4x results. The highest result in each column is in bold, second highest is underlined.

Hyperparameter Evaluation
Activation Function II

Activation	Set5		Set14		Urban100		BSD100	
	2x	4x	2x	4x	2x	4x	2x	4x
PReLU	**35.136**	**29.197**	<u>31.337</u>	<u>26.501</u>	<u>28.045</u>	**23.826**	<u>30.609</u>	<u>26.397</u>
ReLU	<u>35.045</u>	<u>29.107</u>	31.274	26.462	27.992	23.782	**30.628**	**26.418**
LReLU $\alpha = 0.20$	34.988	29.088	31.227	26.431	27.958	23.766	30.573	26.385
LReLU $\alpha = 0.10$	34.952	29.032	**31.352**	**26.520**	27.998	<u>23.798</u>	30.602	26.377
LReLU $\alpha = 0.50$	34.991	29.030	31.304	26.443	28.024	23.769	30.603	26.356
GCU	34.873	28.988	31.268	26.411	28.065	23.791	30.526	26.269
LReLU $\alpha = 0.75$	34.703	28.725	31.083	26.189	27.819	23.596	30.468	26.210
GCU+PReLU	34.768	28.720	31.159	26.192	**28.143**	23.740	30.539	26.150
None	34.327	28.672	30.729	26.047	27.377	23.427	30.232	26.177

Table: PSNR results for various activation functions ordered by descending Set5 4x results. The highest result in each column is in bold, second highest is underlined.

Hyperparameter Evaluation

Loss Metric

Activation	Set5		Set14		Urban100		BSD100	
	2x	4x	2x	4x	2x	4x	2x	4x
MAE	<u>0.9476</u>	**0.8488**	<u>0.8953</u>	<u>0.7402</u>	<u>0.8813</u>	<u>0.7032</u>	<u>0.8730</u>	<u>0.7006</u>
SSIM	**0.9486**	<u>0.8487</u>	**0.9002**	**0.7478**	**0.8873**	**0.7109**	**0.8800**	**0.7099**
MSE	0.9468	0.8457	0.8949	0.7392	0.8809	0.7012	0.8725	0.6994

Table: SSIM results for various loss functions ordered by descending Set5 4x results. The highest result in each column is in bold, second highest is underlined.

Activation	Set5		Set14		Urban100		BSD100	
	2x	4x	2x	4x	2x	4x	2x	4x
MAE	**35.568**	**29.813**	**31.449**	<u>26.753</u>	<u>28.148</u>	<u>23.982</u>	<u>30.664</u>	**26.618**
SSIM	35.163	29.199	31.412	26.538	28.118	23.859	**30.686**	26.457
MSE	<u>35.534</u>	<u>29.804</u>	<u>31.424</u>	**26.781**	**28.202**	**24.006**	30.651	<u>26.600</u>

Table: PSNR results for various loss functions ordered by descending Set5 4x results. The highest result in each column is in bold, second highest is underlined.

Hyperparameter Evaluation

Training Data

Color	Set5		Set14		Urban100		BSD100	
	2x	4x	2x	4x	2x	4x	2x	4x
RGB	**0.9489**	**0.8487**	**0.9006**	**0.7477**	**0.8876**	**0.7111**	**0.8807**	**0.7106**
Gray	0.9484	0.8473	0.9001	0.7468	**0.8876**	0.7106	0.8800	0.7095

Table: SSIM results compared between training on RGB and grayscale images. The highest result in each column is in bold.

Color	Set5		Set14		Urban100		BSD100	
	2x	4x	2x	4x	2x	4x	2x	4x
RGB	**35.124**	**29.125**	**31.336**	**26.472**	28.087	23.824	**30.687**	26.425
Gray	35.056	29.105	31.327	26.470	**28.118**	**23.837**	30.649	**26.434**

Table: PSNR results compared between training on RGB and grayscale images. The highest result in each column is in bold.

Hyperparameter Evaluation

Model Scale

Scale	Set5	Set14	Urban100	BSD100
2	0.8497	0.7481	0.7123	0.7106
4	**0.8606**	**0.7572**	**0.7254**	**0.7178**

Table: SSIM results at 4x upscaling comparing for models trained at 2x and 4x scales. The highest result in each column is in bold.

Scale	Set5	Set14	Urban100	BSD100
2	28.933	26.338	23.789	26.346
4	**29.297**	**26.549**	**23.983**	**26.395**

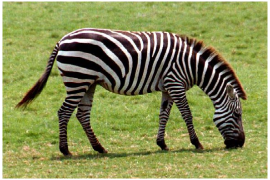
Table: PSNR results at 4x upscaling comparing for models trained at 2x and 4x scales. The highest result in each column is in bold.

Results – Metrics

Dataset	Scale	Bicubic	FSRCNN-s[7]	TinyPSSR-2	TinyPSSR-4
Set5	2	0.9226/31.964	0.9532/36.58	0.9488/34.95	–
Set14	2	0.8341/25.78	0.9052/32.28	0.9004/31.28	–
Urban100	2	0.7933/22.91	–	0.8879/28.10	–
BSD100	2	0.8287/27.233	–	0.8805/30.63	–
Set5	4	0.7890/26.07	0.8499/30.11	0.8497/28.93	0.8606/29.30
Set14	4	0.6774/23.13	0.7423/27.19	0.7481/26.34	0.7572/26.55
Urban100	4	0.6282/20.85	–	0.7123/23.79	0.7254/23.98
BSD100	4	0.6599/24.59	–	0.7106/26.35	0.7178/26.39

Table: SSIM/PSNR results for bicubic interpolation, FSRCNN-s (results are the maximum from their paper), TinyPSSR-2, and TinyPSSR-4 for various test datasets with 2x and 4x super-resolution.

Results – Set14 Zebra Reconstruction at 4x

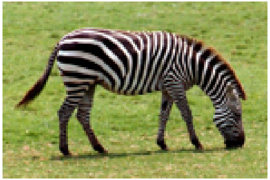


(a) HR

(b) TinyPSSR-4: 0.7600/25.54

(c) TinyPSSR-2: 0.7514/25.43

(d) Bicubic: 0.6290/17.69



(e) LR

Infrared Super-Resolution
Data

Datasets

- FLIR [8]
  - Split into 10,742 training and 1,144 validation
  - Images of size 512x640x1
  - Pre-Processing:
    - Split into 64x64 patches
    - Downscale patches to 32x32
    - 630,000 patches for training
    - 70,000 patches for validation
- OSU Thermal Pedestrian [9]
  - 10 sequences from 18 to 73 frames
  - Total of 284 frames
  - Sequences 9 and 10 (97 frames) used for testing
  - Images of size 360x240

Training

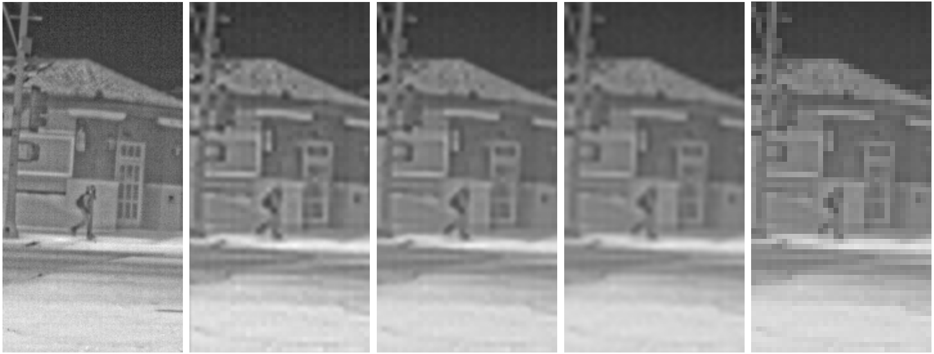
- Loss Function
  - Adam Optimizer
  - Learning Rate: 2x10^{-4}
  - $\beta_1 = 0$
  - $\beta_2 = 0.9$
  - Trained on SSIM and MAE
- Evaluation
  - SSIM and PSNR center cropped four-pixel border
  - Evaluated on Y-channel (luminance) of the YCbCr color space
- Training stopped after 50 epochs of no improvement in validation loss metric

Results – Infrared

Table: SSIM/PSNR results for bicubic and TinyPSSR-IR on the FLIR validation and OSU thermal pedestrian test datasets.

Dataset	Scale	Bicubic	TinyPSSR-IR	TinyPSSR-IR-MAE
FLIR	2	0.8472/33.33	0.8657/35.26	0.8634/35.52
OSU	2	0.7210/32.69	0.7477/32.78	0.7405/32.90
FLIR	4	0.7409/31.08	0.7584/31.26	0.7566/31.68
OSU	4	0.6195/31.21	0.6338/31.08	0.6291/31.31

Results – FLIR 4x



(a) HR (b) TinyPSSR-IR: 0.6374/27.67 (c) TinyPSSR-IR-MAE: 0.6667/29.30 (d) Bicubic: 0.6416/29.00 (e) LR

Figure: SSIM/PSNR comparison 4x scale reconstruction from the FLIR validation set.

Deep Image Compression

Architecture
Overview

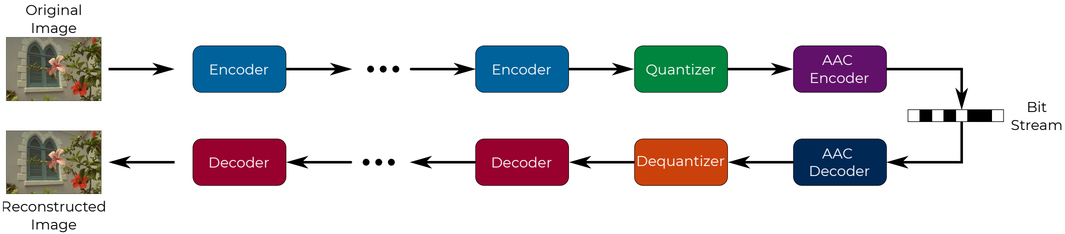


Figure: Iterative scheme for image compression.

Architecture
Encoder

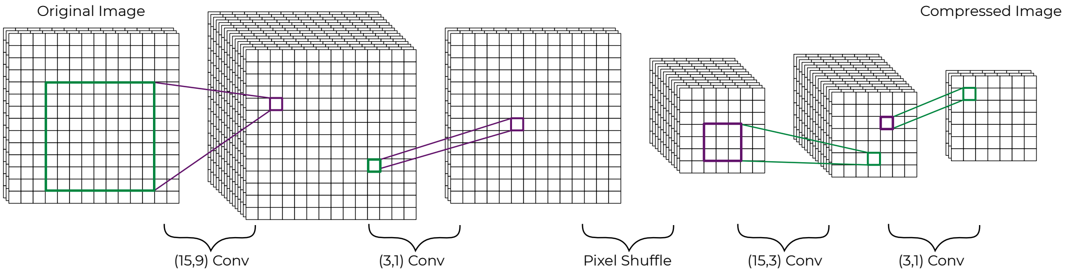


Figure: Encoding block.

Architecture
Decoder

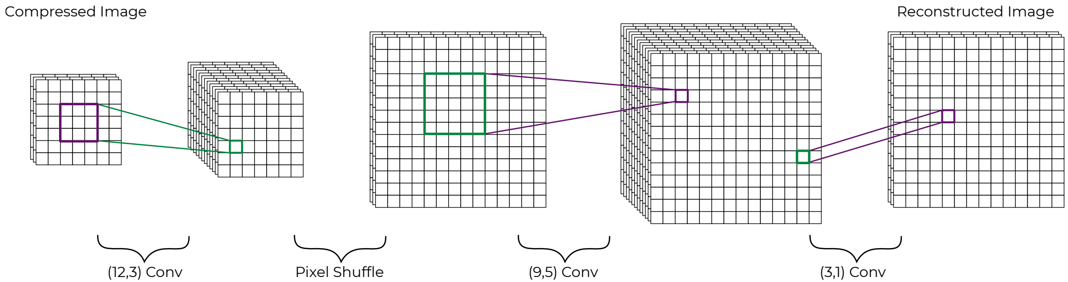


Figure: Decoding block.

Architecture: Parameter Breakdown

Encoding Block		Decoding Block	
Layer	**Parameters**	**Layer**	**Parameters**
(15,9) Convolution	3,660	(12,3) Convolution	336
PReLU	15	PReLU	12
(3,1) Convolution	48	Pixel Shuffle (up)	0
PReLU	3	(9,5) Convolution	684
Pixel Shuffle (down)	0	PReLU	9
(15,3) Convolution	1,635	(3,1) Convolution	30
PReLU	15	Sigmoid	0
(3,1) Convolution	48		
PReLU	3		
Total	**5,427**	**Total**	**1,071**

Table: Parameters per layer and activation function of TinyCompress.

Non-Learned Compression

Data Type Quantization

- Converting float32 to uint8
  1. Normalize values over [0,1]
  2. Scale to [0,255]
- Minimal loss in accuracy when normalizing instead of clipping

Bit Plane Quantization

- Separate 24-bit pixel values into 24 single bit-planes
- Remove from the least significant bit-plane
- Performed on compressed data post-quantization

Adaptive Arithmetic Coding

- Reduces average number of bits per pixel
- Adaptively encodes sequentially over the image
- Performs only one pass over the image
- Lossless

Data

Train

- 800 images from DIV2K [1] training split. Sizes range from 648x2040 to 2040x2040.
- Pre-Processing:
  - Split into 64x64 patches
- 100,000 of 522,939 patches used during training
- Trained at $\frac{1}{4}$, $\frac{1}{16}$, $\frac{1}{64}$, and $\frac{1}{256}$ scales
  - Static compression rates of 75%, 93.75%, 98.44%, and 99.61%

Validation & Test

- Mixed color and grayscale images
- Datasets:
  - Kodak PhotoCD [10] (24 images)
  - Set5 [3] (five images)
  - Set14 [4] (14 images)
- Validation on same four scales as training
- Testing only on $\frac{1}{4}$, $\frac{1}{16}$, and $\frac{1}{64}$

Training and Metrics

Training

- Loss Function
  - Adam Optimizer
  - Learning Rate: 5x10^{-4}
  - $\beta_1 = 0.9$
  - $\beta_2 = 0.999$
- Loss Metric
  - Mean Squared Error
- Training stopped after 20 epochs of no improvement in validation loss metric

Metrics

- Measure MS-SSIM, PSNR, and bits per pixel (bpp)
- PSNR calculated on Y-channel (luminance) of the YCbCr color space
- MS-SSIM calculated on original, $\frac{1}{2}$, $\frac{1}{4}$, $\frac{1}{8}$, and $\frac{1}{16}$ sizes

Results

Adaptive Arithmetic Coding

Table: Bits per pixel results on the Kodak set after implementing adaptive arithmetic coding.

Model	Static	With AAC	Improvement
TinyCompress-4	6.000	5.1043	14.93%
TinyCompress-16	1.500	1.2719	15.21%
TinyCompress-64	0.375	0.3216	14.24%

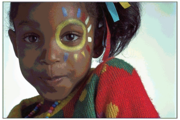
Results
Bit-Plane Quantization

(a) 8 bit planes. 0.9913/39.6596/ 5.2862

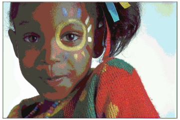
(b) 7 bit planes. 0.9957/39.0109/ 4.5383

(c) 6 bit planes. 0.9941/36.8899/ 3.7905

(d) 5 bit planes. 0.9877/32.6625/ 3.0464



(e) 4 bit planes. 0.9594/27.0897/ 2.3132



(f) 3 bit planes. 0.8666/20.9543/ 1.5867

(g) 2 bit planes. 0.7101/15.0303/ 1.0825

(h) 1 bit plane. 0.6353/11.2548/ 0.5684

Figure: Effects of bit-plane quantization on MS-SSIM/PSNR/bpp using TinyCompress-4.

Results
Kodak

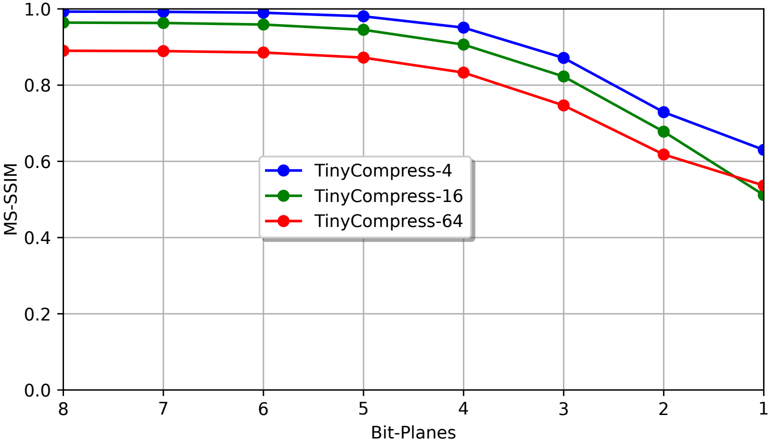


Figure: MS-SSIM results with respect to bit-planes kept.

Results
Kodak

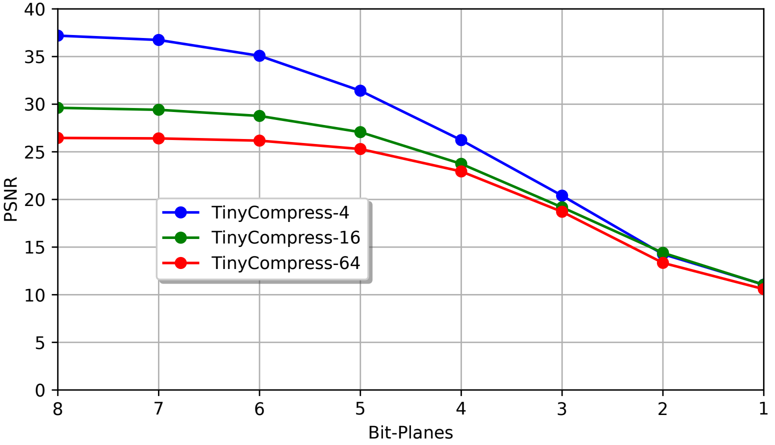


Figure: PSNR results with respect to bit-planes kept.

Results
Kodak

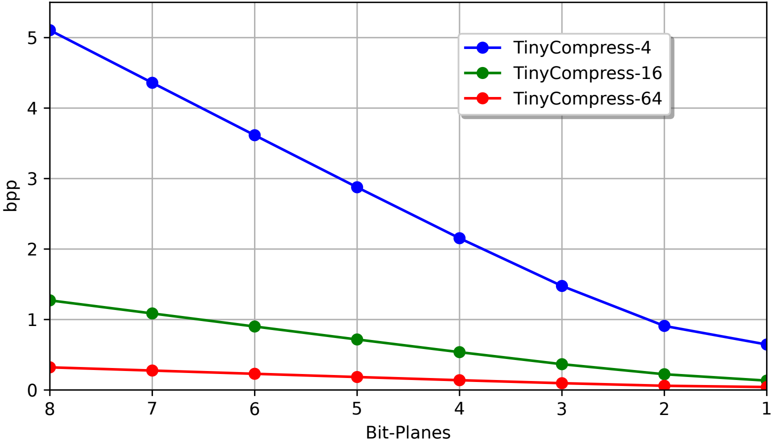


Figure: Bits per pixel results with respect to bit-planes kept.

Results
Kodak Image

(a) Original

(b) TinyCompress-4: 0.9961/39.6596/5.2862

(c) TinyCompress-16: 0.9819/31.9334/1.2758

(d) TinyCompress-64: 0.9387/28.5576/0.3350

Figure: MS-SSIM/PSNR/bpp results for an image from Kodak using TinyCompress-4, -16, and -64.

Results

Comparison with JPEG

(a) TinyCompress-4
0.9959/40.0778/4.8578

(b) JPEG
0.9975/51.8739/5.1224

(c) TinyCompress-16
0.9759/31.6855/0.9909

(d) JPEG:
0.9881/39.8930/1.0562

Figure: TinyCompress and JPEG comparisons with MS-SSIM/PSNR/bpp.

Results
Comparison with JPEG

(a) TinyCompress-64
0.9228/27.6528/0.3075

(b) JPEG
0.9277/32.6283/0.3145

(c) TinyCompress-256
0.8285/25.1362/0.0811

(d) JPEG
0.7834/25.7558/0.1643

Figure: TinyCompress and JPEG extreme compression comparisons with MS-SSIM/PSNR/bpp.

Conclusion

Conclusion

- Memory-Efficient Single-Image Super-Resolution
  - Extremely small – 1,356 parameters
  - Performs similar to or better than the nearest-sized models
- Deep Compression
  - Designed an iterative compression model
  - Low-memory requirements
    - Encoder – 5,427 parameters
    - Decoder – 1,071 parameters
  - Quality metrics similar to JPEG

Bibliography

Bibliography I

[1] Agustsson, E. and Timofte, R., “Ntire 2017 challenge on single image super-resolution: Dataset and study,” in [*The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*], (July 2017).

[2] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L., “ImageNet Large Scale Visual Recognition Challenge,” *International Journal of Computer Vision (IJCV)* **115**(3), 211–252 (2015).

[3] Bevilacqua, M., Roumy, A., Guillemot, C., and Alberi-Morel, M.-L., “Low-complexity single image super-resolution based on nonnegative neighbor embedding,” (09 2012).

[4] Zeyde, R., Elad, M., and Protter, M., “On single image scale-up using sparse-representations,” in [*Curves and Surfaces*], Boissonnat, J.-D., Chenin, P., Cohen, A., Gout, C., Lyche, T., Mazure, M.-L., and Schumaker, L., eds., 711–730, Springer Berlin Heidelberg, Berlin, Heidelberg (2012).

[5] Martin, D., Fowlkes, C., Tal, D., and Malik, J., “A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics,” in [*Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*], **2**, 416–423 vol.2 (2001).

[6] Huang, J.-B., Singh, A., and Ahuja, N., “Single image super-resolution from transformed self-exemplars,” in [*2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*], 5197–5206 (2015).

Bibliography II

[7] Dong, C., Loy, C. C., and Tang, X., "Accelerating the super-resolution convolutional neural network," *CoRR* **abs/1608.00367** (2016).

[8] "Free teledyne flir thermal dataset for algorithm training," *FREE - FLIR Thermal Dataset for Algorithm Training | Teledyne FLIR* .

[9] Davis, J. W., "Otcbvs benchmark dataset collection," *OTCBVS* .

[10] Franzen, R. W., "Kodak lossless true color image suite," *True Color Kodak Images* .

[11] Dong, C., Loy, C. C., He, K., and Tang, X., "Learning a deep convolutional network for image super-resolution," in [*Computer Vision – ECCV 2014*], Fleet, D., Pajdla, T., Schiele, B., and Tuytelaars, T., eds., 184–199, Springer International Publishing, Cham (2014).

[12] Shi, W., Caballero, J., Huszár, F., Totz, J., Aitken, A. P., Bishop, R., Rueckert, D., and Wang, Z., "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network," in [*2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*], 1874–1883 (2016).

[13] Rippel, O. and Bourdev, L., "Real-time adaptive image compression," (2017).

Q&A

Appendix

Hyperparameter Evaluation
Activation Function

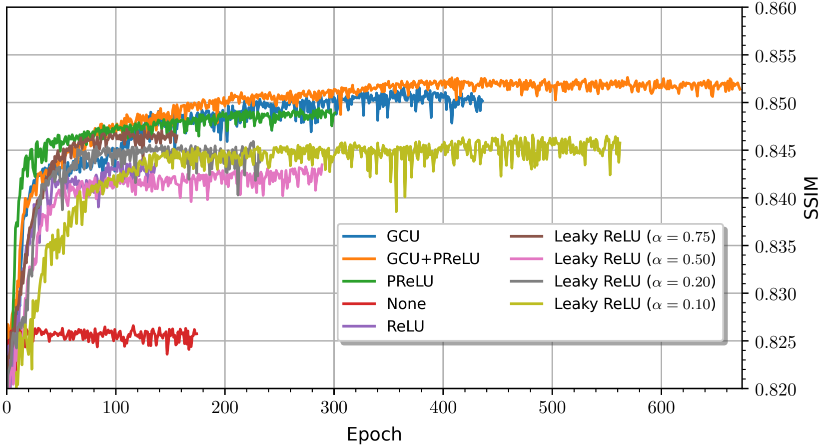


Figure: Training history SSIM comparison on Set5 4x between various activation functions.

Hyperparameter Evaluation
Loss Metric

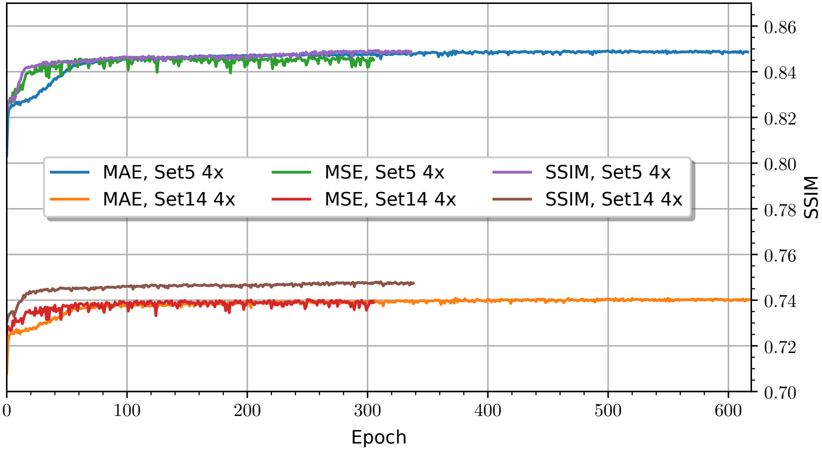


Figure: Training history SSIM comparison between various loss metrics.

Hyperparameter Evaluation
Training Data

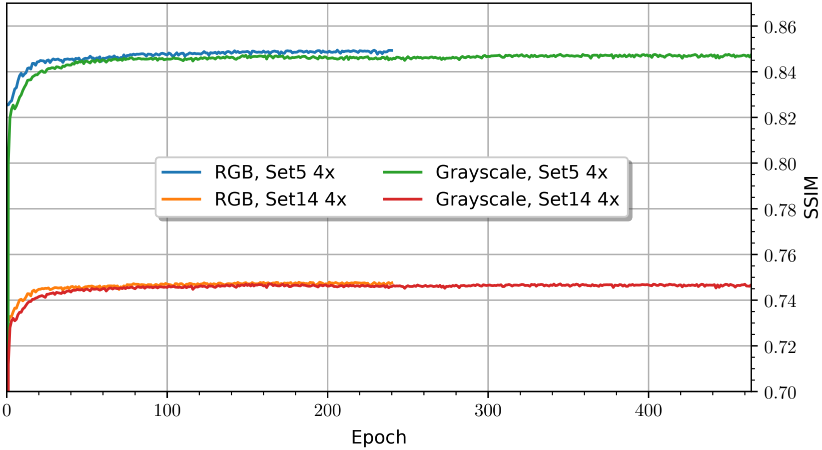


Figure: Training history SSIM comparison between training on RGB and grayscale images.

Hyperparameter Evaluation
Model Scale

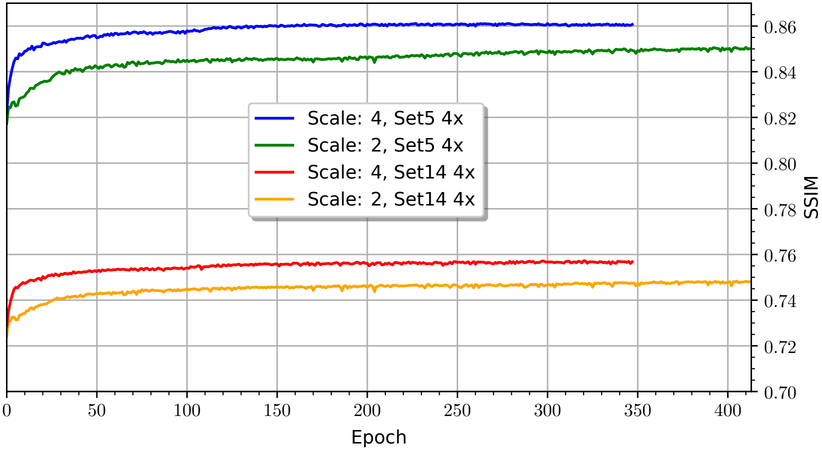


Figure: Training history SSIM comparison between 2x and 4x scale models.

Model Comparison

Dataset	FSRCNN-s[7]	FSRCNN[7]	SRCNN[11]	ESPCN[12]	TinyPSSR-4
Set5	0.8499/30.11	0.8660/30.71	0.8628/30.48	0.8646/30.66	0.8606/29.30
Set14	0.7423/27.19	0.7550/27.59	0.7513/27.50	0.7562/27.71	0.7572/26.55
Urban100	–	0.7280/24.62	0.7221/24.52	0.7360/24.60	0.7254/23.98
Parameters	3,937	12,464	57,184	24,384	2,716

Table: SSIM/PSNR results for 4x upscale using TinyPSSR-4 compared to other works.

Execution Time

Table: Execution time in milliseconds for encoding and decoding only using GPU and CPU.

Model	Encoding-GPU	Encoding-CPU	Decoding-GPU	Decoding-CPU
TinyCompress-4	2.7	29.7	1.5	18.4
TinyCompress-16	4.4	39.2	2.7	23.5
TinyCompress-64	6.0	42.4	3.6	26.3