Computer Vision Project

Topic Image Super Resolution

TA assigned Gowri Lekshmy

Team Vision21

Members

Astitva Ayan

Trusha Aryamaan

Paper Implementation:

Image Super-Resolution Using Deep Convolutional Networks

Chao Dong, Chen Change Loy, Member, IEEE, Kaiming He, Member, IEEE and Xiaoou Tang, Fellow, IEEE

2. Objectives

- **Implement algorithm** given in paper and test it on images.
- **Compare algorithm** to **classic CV methods** including:
    - Yang, J., Wright, J., Huang, T.S., Ma, Y.:
      Image super-resolution via sparse representation. IEEE Transactions on Image Processing 19(11), 2861–2873 (2010)
    - Chang, H., Yeung, D.Y., Xiong, Y.:
      Super-resolution through neighbor embedding. In: IEEE Conference on Computer Vision and Pattern Recognition (2004)
    - Kim, K.I., Kwon, Y.:
      Single-image super-resolution using sparse regression and natural image prior. IEEE Transactions on Pattern Analysis and Machine Intelligence 32(6), 1127–1133 (2010)
    - Timofte, R., De Smet, V., Van Gool, L.:
      Anchored neighborhood regression for fast example-based super-resolution. In: IEEE International Conference on Computer Vision. pp. 1920–1927 (2013)

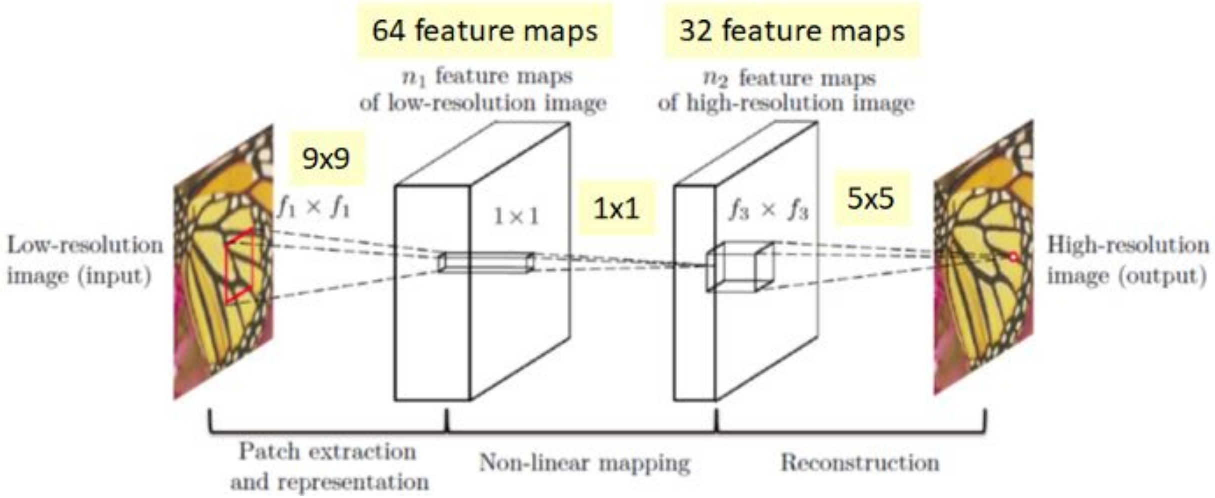
2. Objectives (cont'd)

- Use metrics such as PSNR, SSIM, MSSSIM for comparison with classic CV methods.

- Try variations of algorithm as given in paper by varying depth of network, number of channels and other hyperparameters. Compare results based on output quality and time taken.

- Visualize and interpret the channels learnt.

3. Method overview



3. Method overview (cont'd)

- This paper illustrates the use of CNN's for image super-resolution.
- The basic version of this algorithm uses a 3-layer CNN.
- Dataset to be used: Subset of ImageNet
- Input will be made as a part of preprocessing, in which the images of ImageNet will be converted to lower resolution images. This is sent to feed-forward network.
- Output will be original high resolution image.
- Loss function: MSE loss
- Iterations and other hyperparameters: to be tested as part of experiment.

4. Goals

- To build a network for image super-resolution.
- To survey classic CV methods for image super-resolution and compare it with SRCNN.
- To try variations of the proposed network by changing hyperparameters and analyzing the results.

5. Previous Timeline

- **Mid Evals:** Implement SRCNN and validate the results.

- **End Evals:** Review classical algorithms and present comparison results.

Datasets used

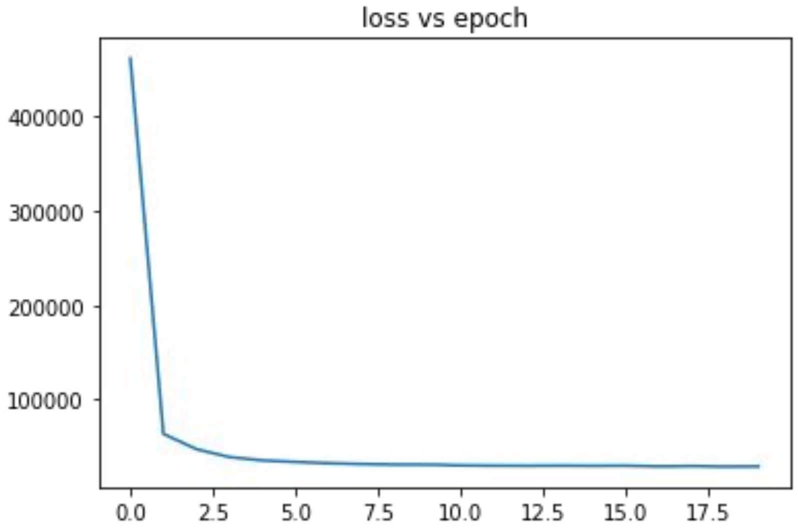
- 91-image train set mentioned in the paper is used.
- Set-5 and Set-14 test sets yield coherent results.
- While training, for each image several windows are also taken to increase the dataset size.

Mid-evals Outputs

Number of Epochs = 20

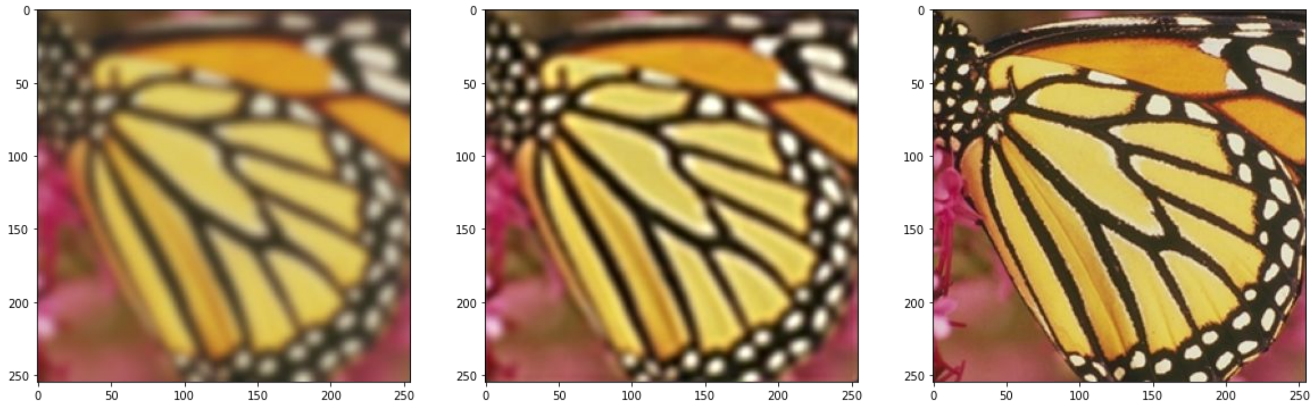
Learning rate = 1e-4

Loss Type = MSE



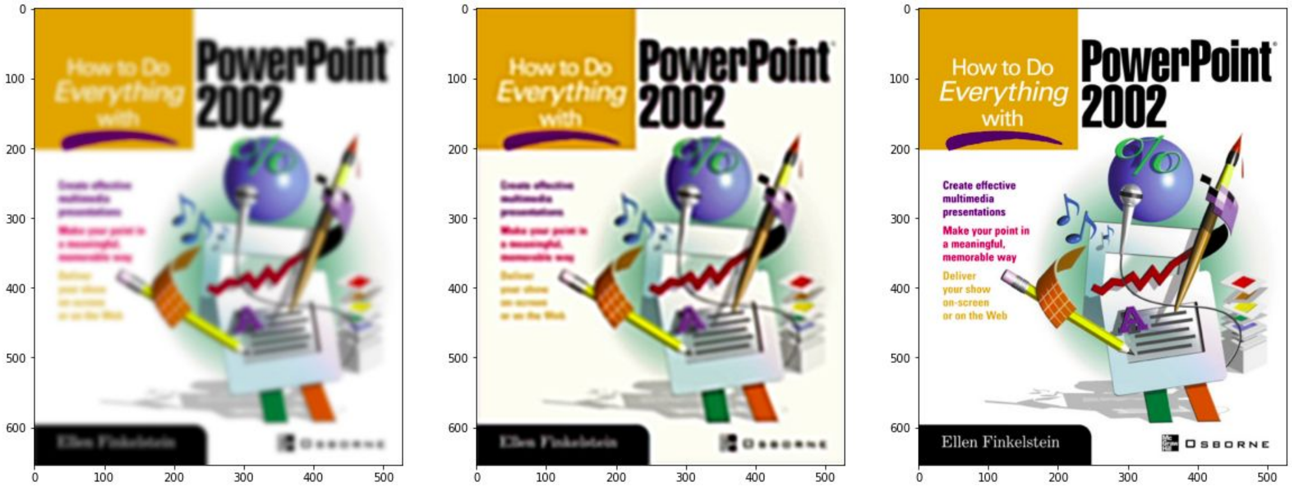
Test Image 1

SRCNN PSNR: 27.95 dB
Our PSNR: 29.19 dB



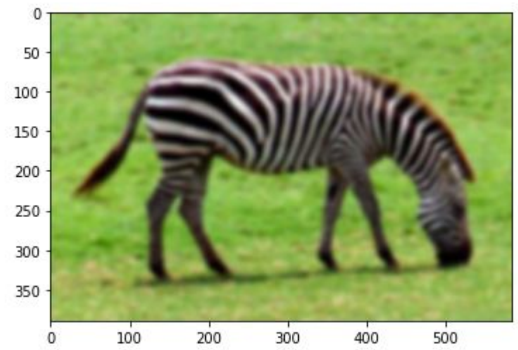
Test Image 2

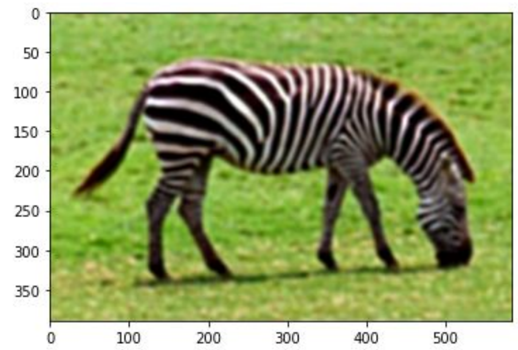
SRCNN PSNR: 27.04 dB
Our PSNR: 30.54 dB

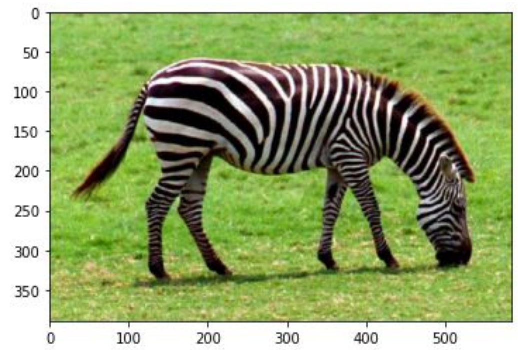


Test Image 3

SRCNN PSNR: 29.29dB
Our PSNR: 29.54dB

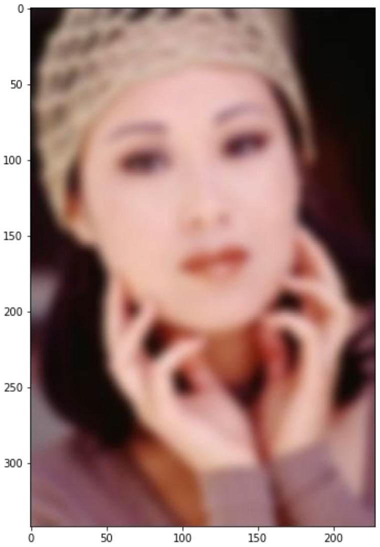


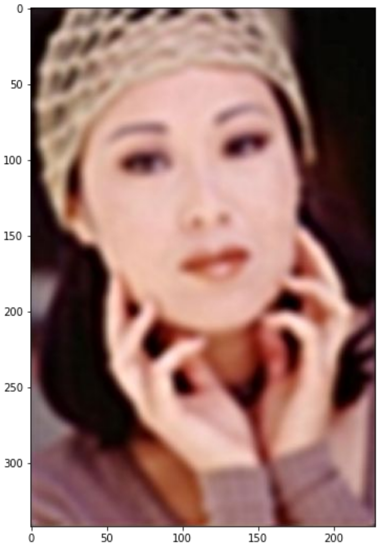




Test Image 4

SRCNN PSNR: 30.16dB
Our PSNR: 31.20dB





Metrics:

- Full-Reference metrics
- Absolute Error measurements
- Perception based models

Metrics (Cont'd):

1. Absolute error measurement:
    a. MSE: Mean Squared Error
    b. PSNR: Peak Signal to Noise Ratio

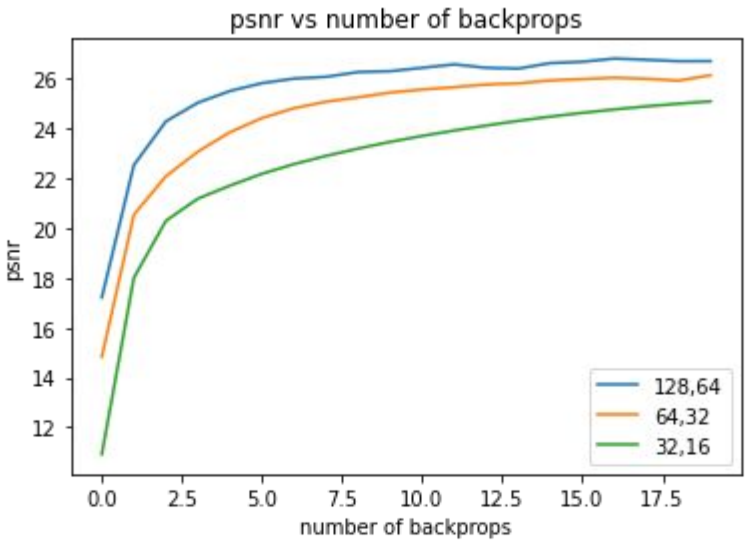
2. Perceptual Models - Structural Information
    a. SSIM: Structure Similarity Index
    b. MSSSIM : Multi-Scale SSIM

Experiments

List of experiments:

1. Number of filters per layer
2. Size of Filter
3. Number of layers
4. Comparison with classical models
5. Noise comparison with different metrics
6. Time comparison with different networks

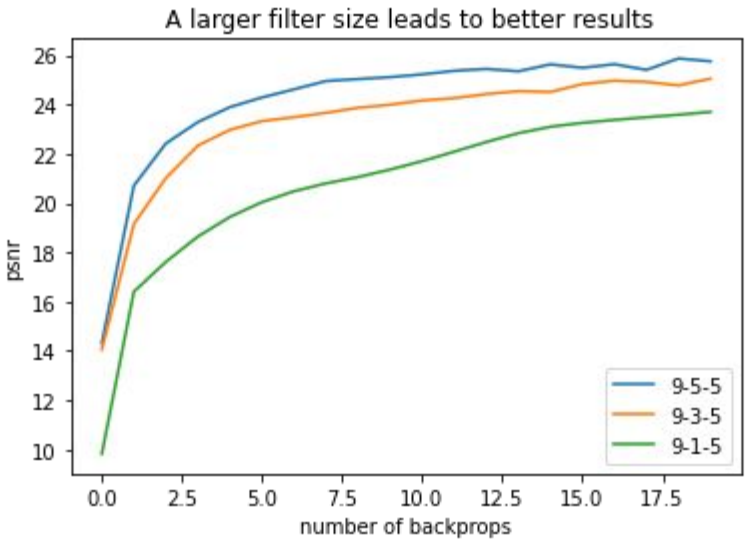
1. Number of filters per layer



Result:

Higher the number of filters per layer, Higher is the observed PSNR

2. Filter size



Result:

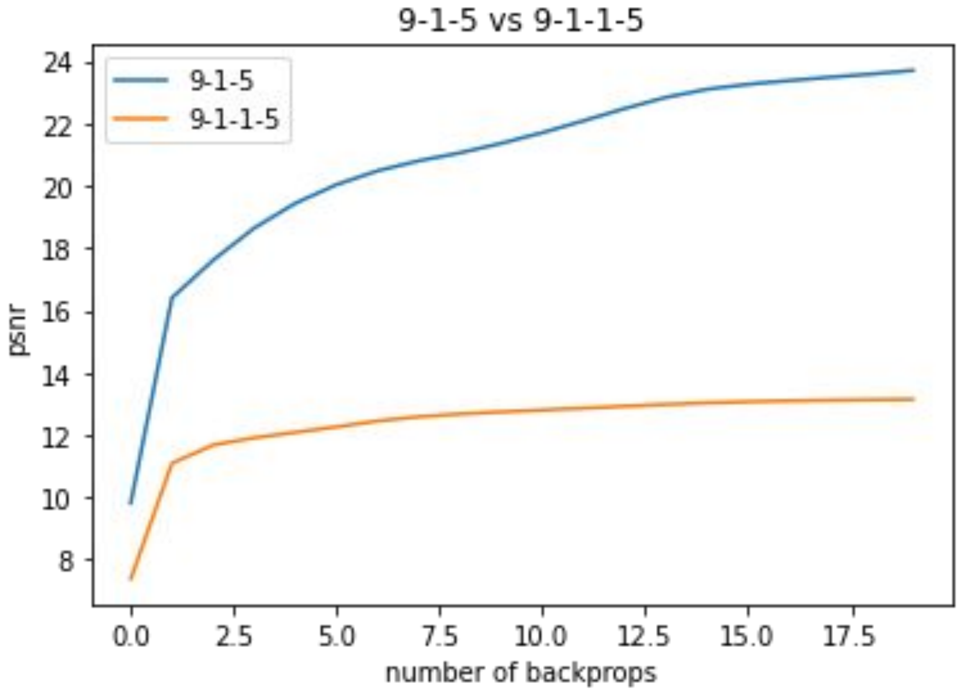
Increasing the filter size improves the performance. We can see the filter in the middle layer increasing proportionate to the performance.

3. Number of layers

1. Compared 3 Layer vs 4 Layer models
2. Added 1x1 Convolutions to increase layer depth
3. 3 layer models outperform 4 layer after adding 1x1 convolutions. This may be due to difficulty of training.

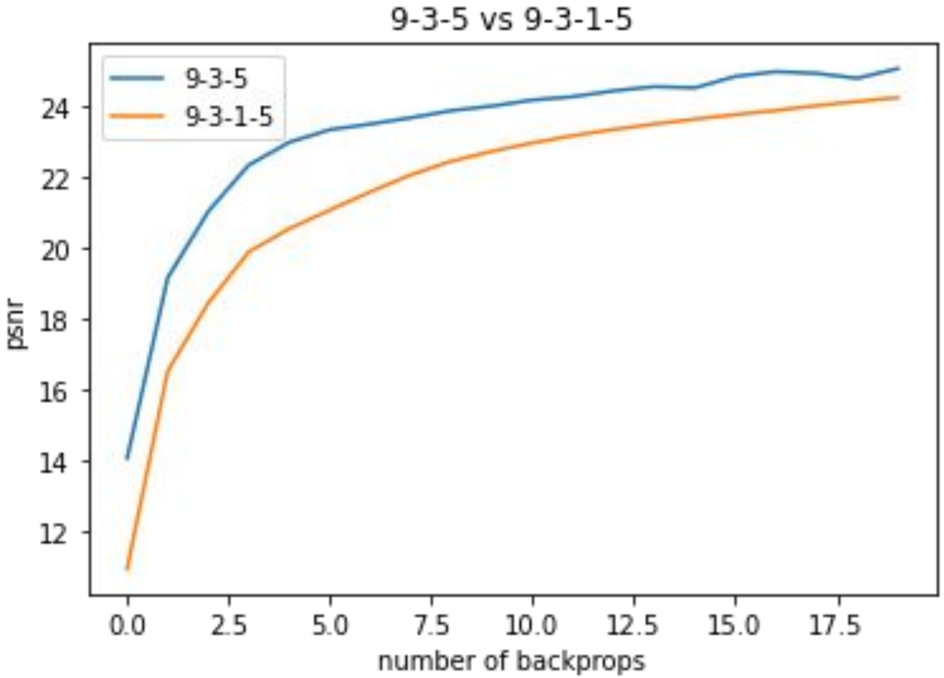
3.1 Number of layers

```
Layer1: 9x9
Layer2: 1x1
Layer3: 5x5
Vs
new 1x1 2nd last layer
```



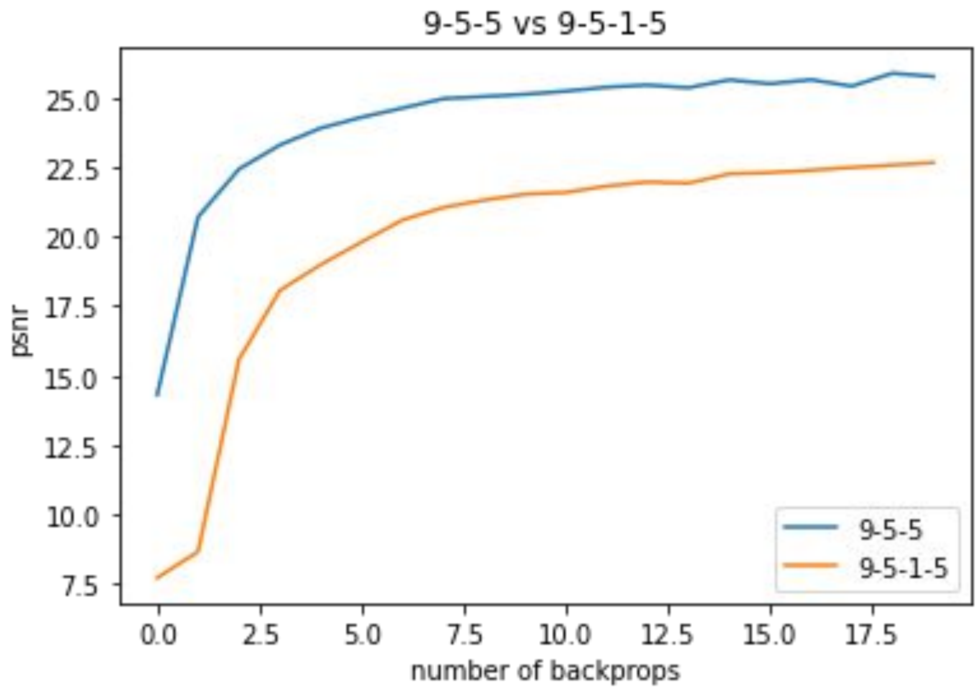
3.2 Number of layers

Layer1: 9x9
Layer2: 3x3
Layer3: 5x5
Vs
new 1x1 2nd last layer

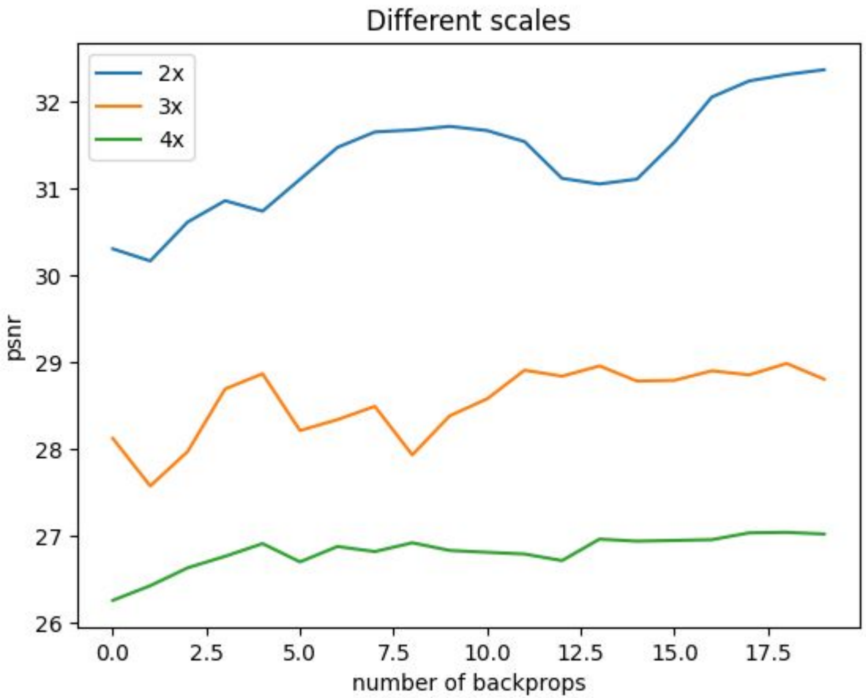


3.3 Number of layers

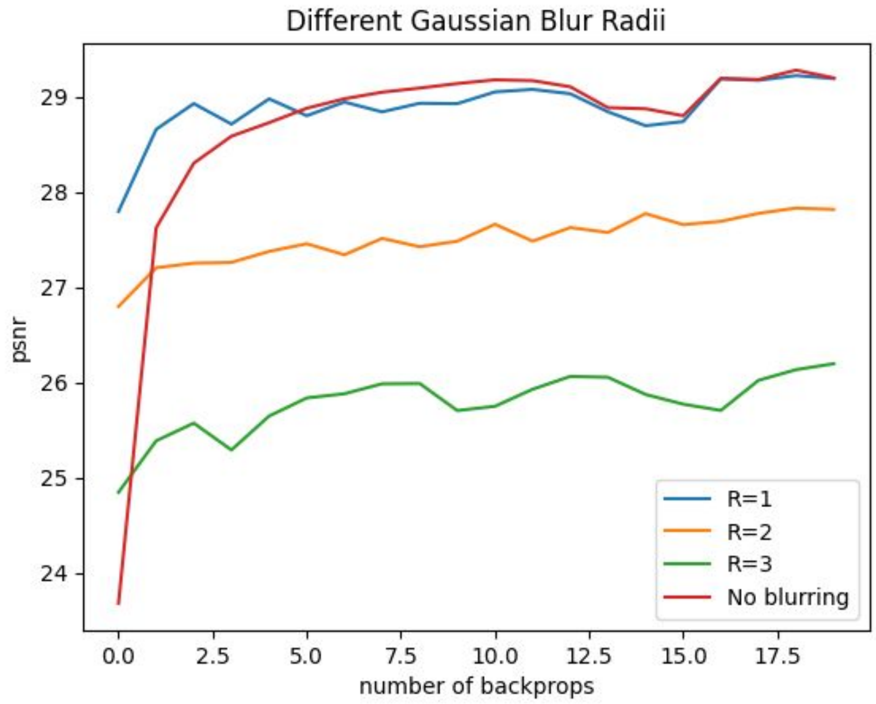
Layer1: 9x9
Layer2: 5x5
Layer3: 5x5
Vs
new 1x1 2nd last layer



4. Performance Across Scales

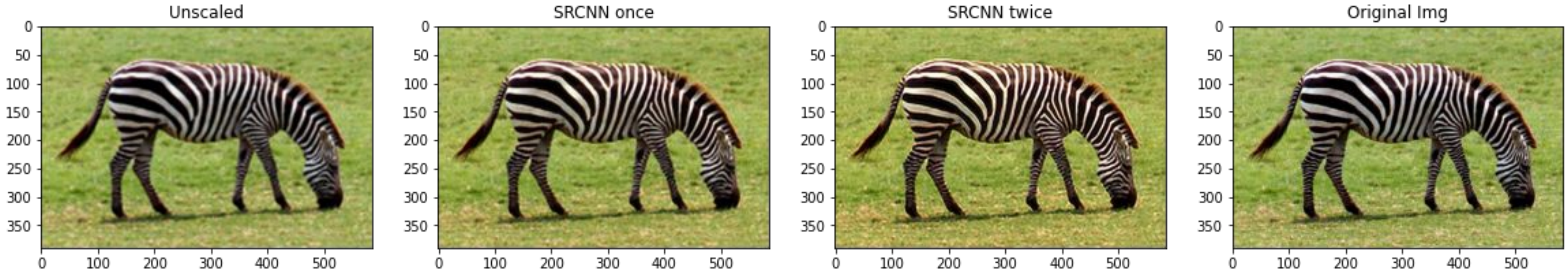


5. Changing gaussian blur radius



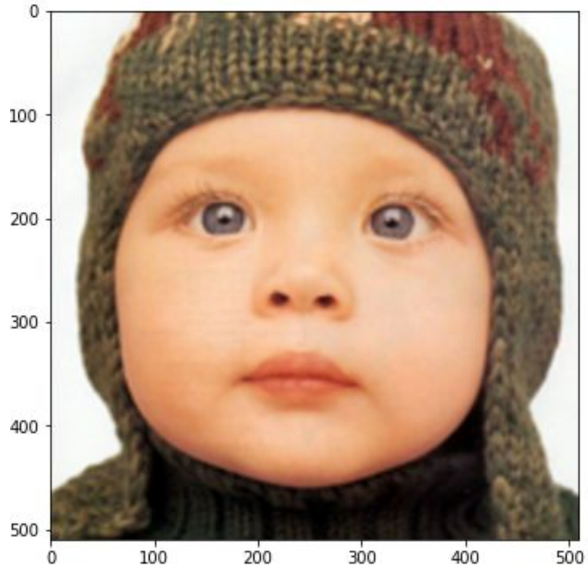
Additional Experiment #1

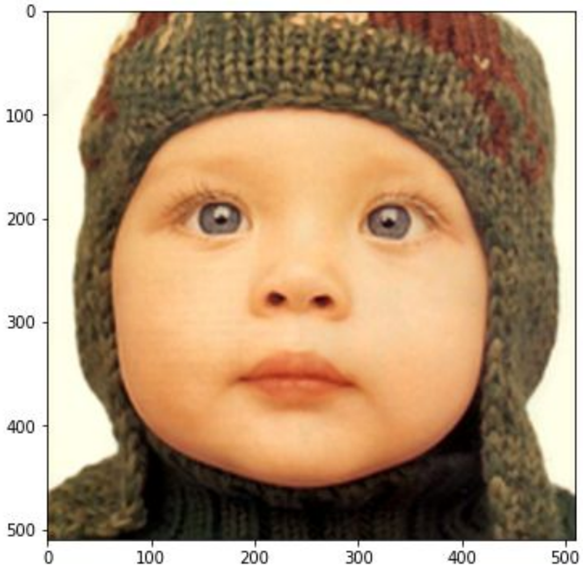
We also ran the model results on same image twice, which does not provide any improvement in quality or PNSR. Artifacts were observed.

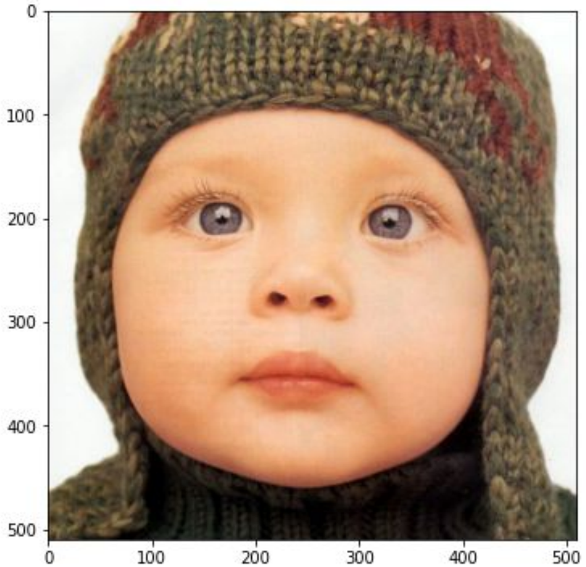


Additional Experiment #2

Inspiration: Need for colour correction







Additional Experiment #2 (Cont'd)

- Colour artifacts are perceptually discernable in some cases.
- Some method to regularize colour across model configurations and training states.
- To ensure colour control, a classical method might be better.

Additional Experiment #2 (Cont'd)

- Idea: Fit the distribution of colours of input image to image output by the model.
- Solution: Histogram Matching

Additional Experiment #2 - Results

For our best results in SRCNN, we observe the following-

- Set14 is slightly worse across all entries.
- Set5 is slightly better across all entries.

Average Values	PSNR	SSIM	MSSSIM	
Set 14	32.07818786	0.8438833053	0.9628345683	
Set 14 (Matched)	32.04619691	0.8427170627	0.9621960341	Matched performs worse
Set 5	33.31653126	0.9196947115	0.9828535978	
Set 5 (Matched)	33.36905171	0.9207541561	0.9829994907	Matched performs better

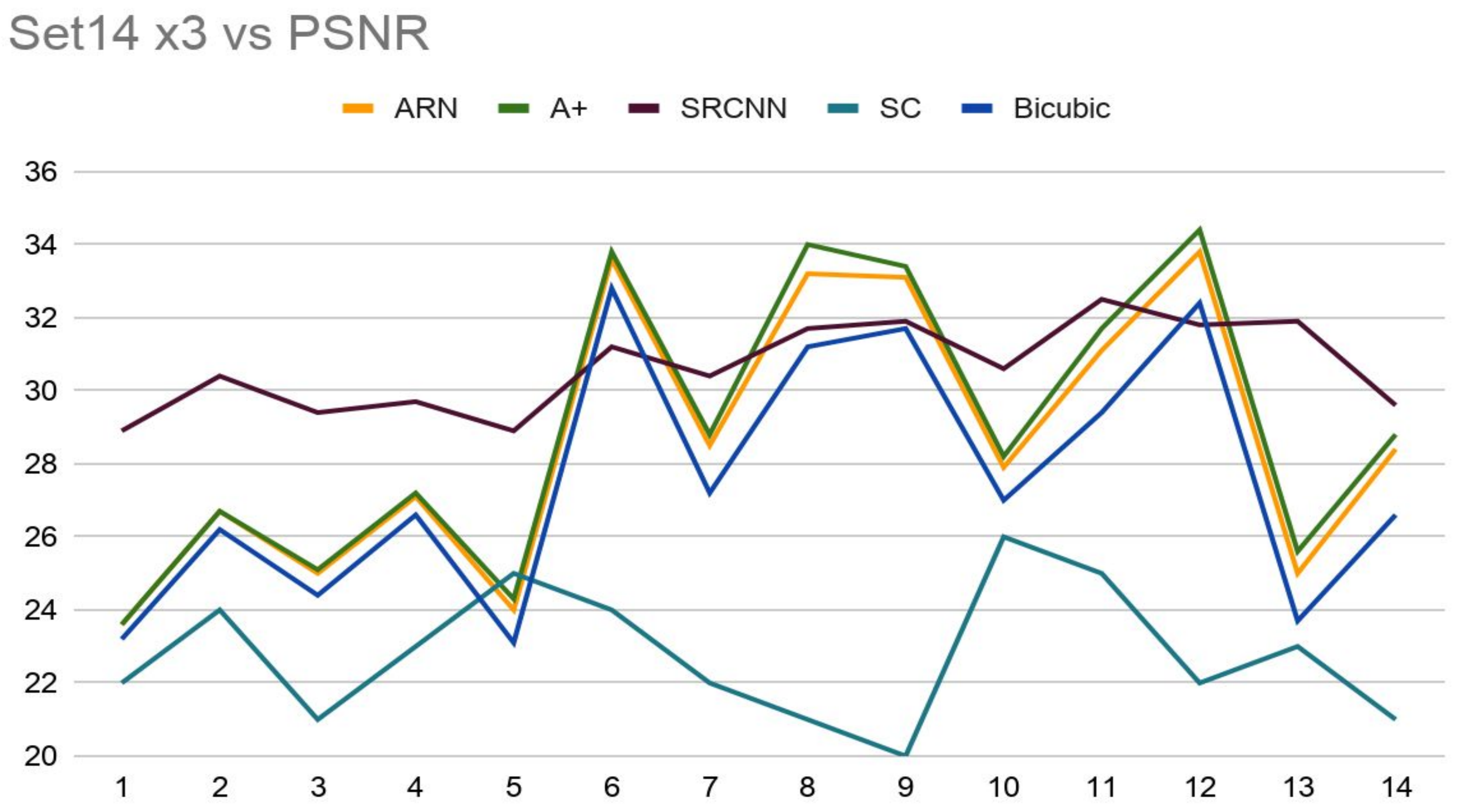
Additional Experiment #2 - Conclusion

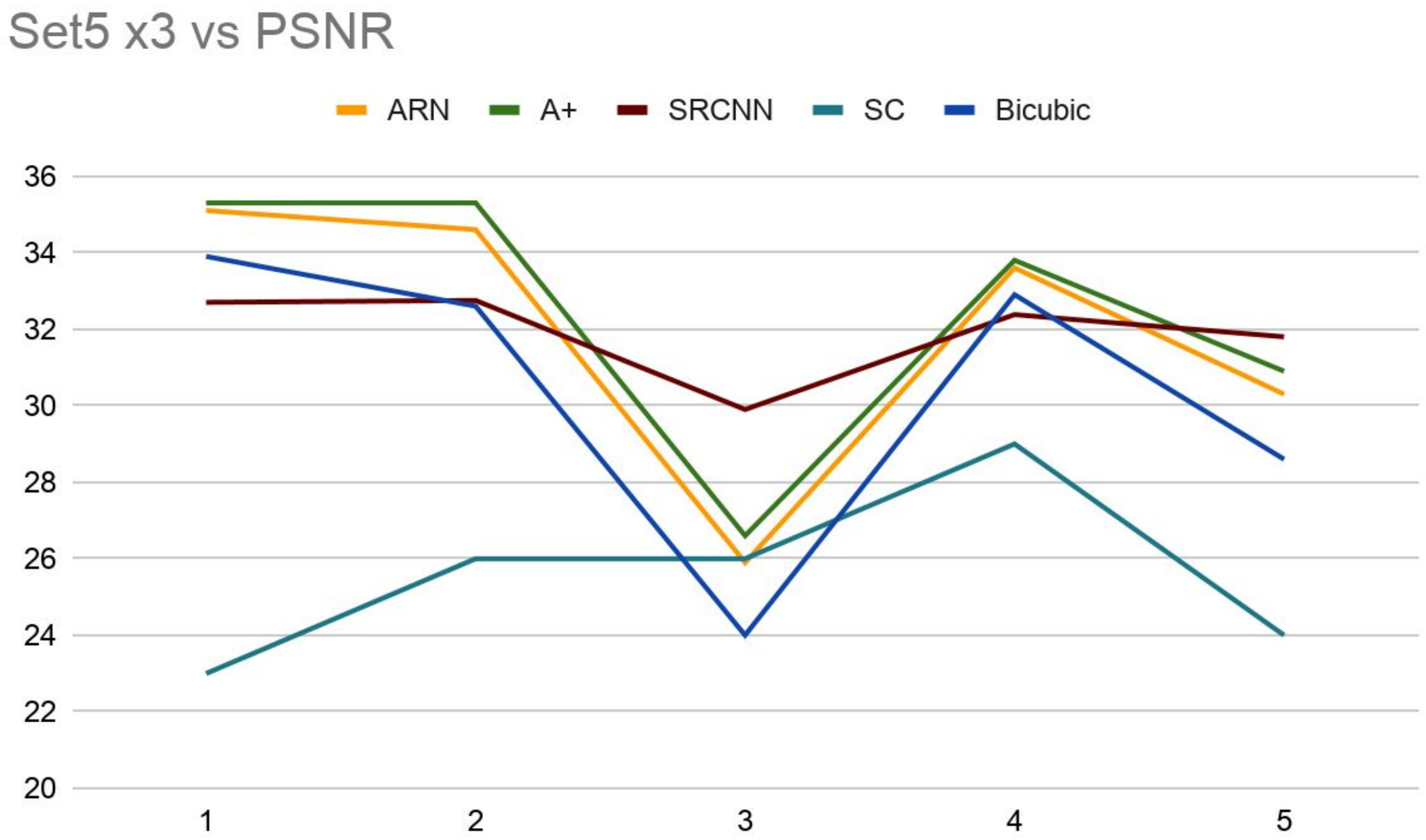
- Observable improvement in colour is observed even for best models.
- Quality of image being noticeably lowered is NOT observed so far.
- Significant difference for models with noticeable colour issues.
- Future work: Let it be done in a way that errors can be back-propagated.

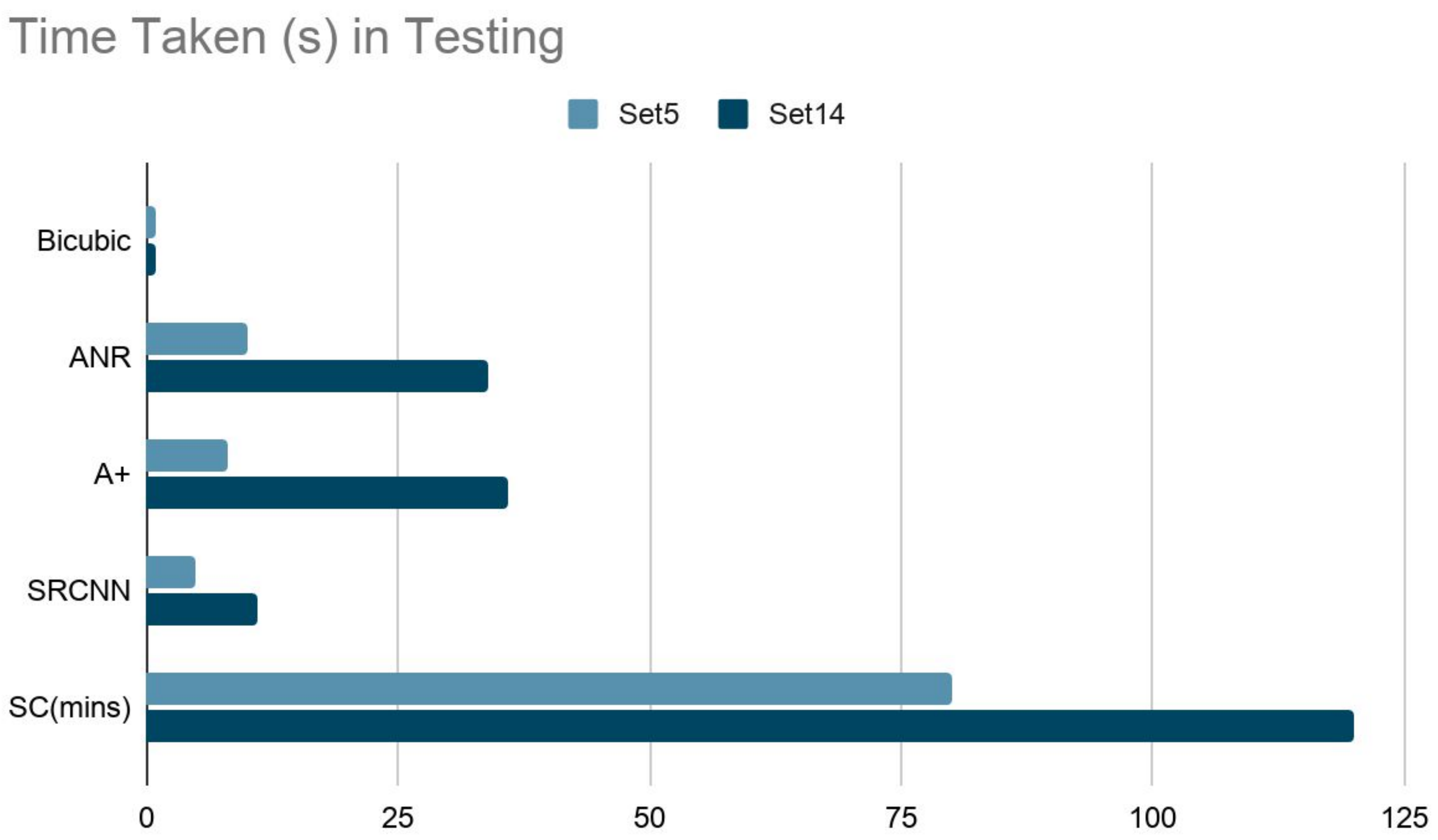
Visualizations

- Time and accuracy comparison for different classical models

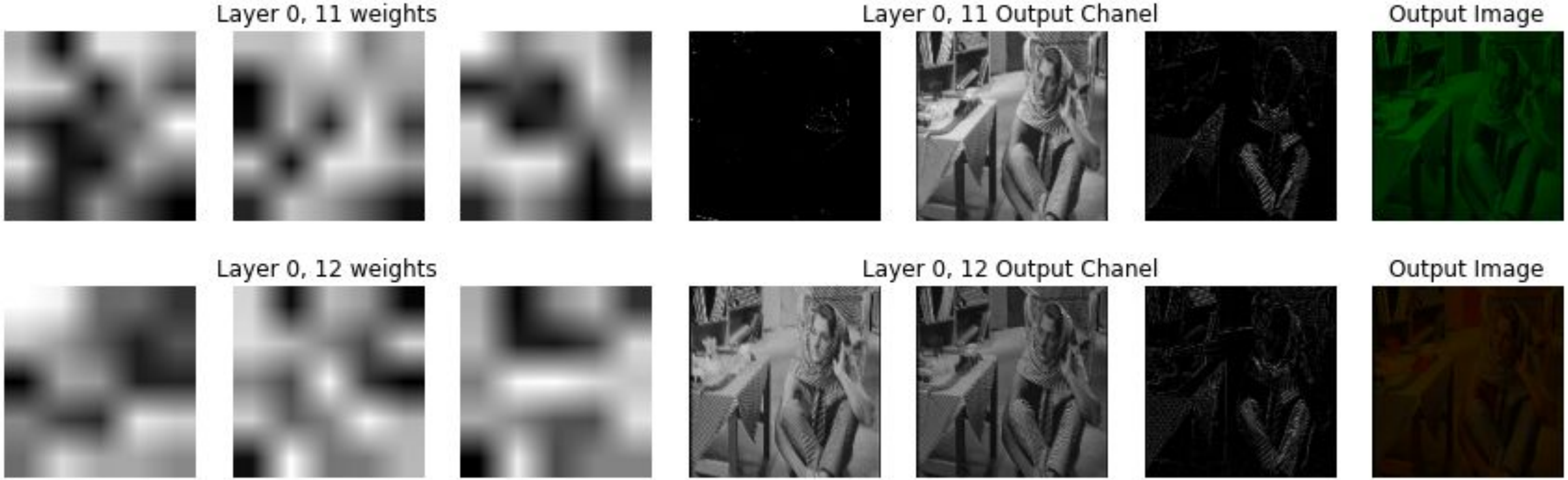
- Feature maps obtained from layer outputs


Set14 x3 vs PSNR
ARN
A+
SRCNN
SC
Bicubic
36
34
32
30
28
26
24
22
20
1
2
3
4
5
6
7
8
9
10
11
12
13
14


Set5 x3 vs PSNR
ARN
A+
SRCNN
SC
Bicubic
36
34
32
30
28
26
24
22
20
1
2
3
4
5


Time Taken (s) in Testing
Set5
Set14
Bicubic
ANR
A+
SRCNN
SC(mins)
0
25
50
75
100
125

Convolution layers And feature maps visualized



Classic model: Super-Resolution via Sparse Representation

Single-image super-resolution, based on sparse signal representation. Research on image statistics suggests that image patches can be well represented as a sparse linear combination of elements from an appropriately chosen over-complete dictionary. Inspired by this observation, this paper seek a sparse representation for each patch of the low-resolution input, and then use the coefficients of this representation to generate the high-resolution output. Jointly training two dictionaries for low and high resolutions gives ability to bring about coherent superresolution.

Classical model Explanation

Basic Idea: $\boldsymbol{y} \doteq L\boldsymbol{x} = L\boldsymbol{D}\boldsymbol{\alpha}_0,$

Reconstruction Constrain: $\boldsymbol{Y} = SH\boldsymbol{X}$

Sparsity prior: $\boldsymbol{x} \approx \boldsymbol{D}_h\boldsymbol{\alpha}$ for some $\boldsymbol{\alpha} \in \mathbb{R}^K$ with $\|\boldsymbol{\alpha}\|_0 \ll K.$

Final Optimization formulation: $$\min \|\boldsymbol{\alpha}\|_1 \quad \text{s.t.} \quad \|F\boldsymbol{D}_l\boldsymbol{\alpha} - F\boldsymbol{y}\|_2^2 \le \epsilon_1, \quad \|P\boldsymbol{D}_h\boldsymbol{\alpha} - \boldsymbol{w}\|_2^2 \le \epsilon_2,$$

Dictionary learning:

$$\boldsymbol{D}_h = \arg \min_{\{\boldsymbol{D}_h, Z\}} \|X^h - \boldsymbol{D}_h Z\|_2^2 + \lambda\|Z\|_1,$$

$$\boldsymbol{D}_l = \arg \min_{\{\boldsymbol{D}_l, Z\}} \|Y^l - \boldsymbol{D}_l Z\|_2^2 + \lambda\|Z\|_1,$$

Thank You.