Machine Learning for Facial Image Super-resolution

CHEUNG Tsun Hin (15083269D)

Supervisor: Prof. Kenneth LAM

Date: 23 April 2019

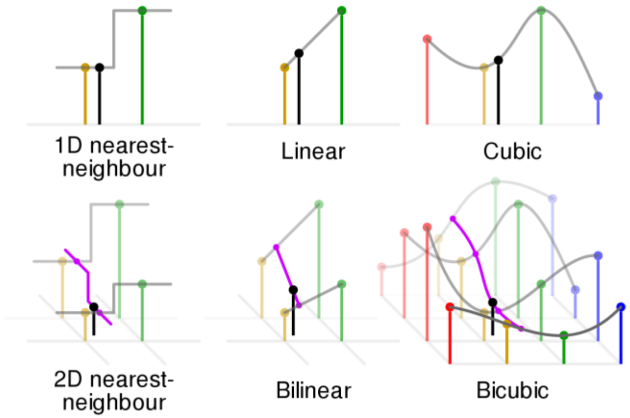
Image Super-resolution (SR)

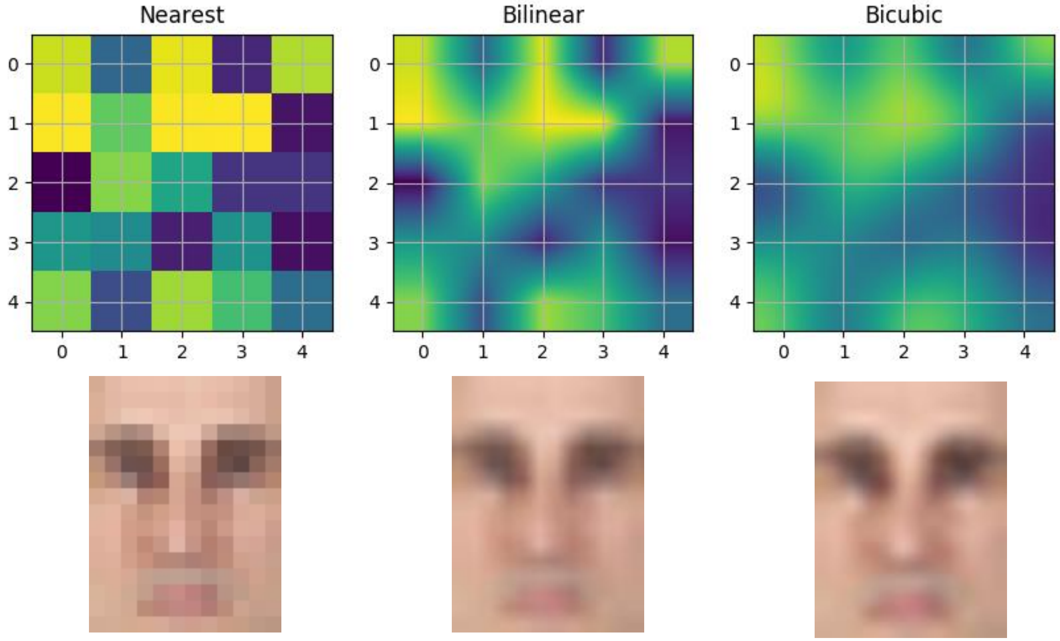
low-resolution image

upscale

high-resolution image

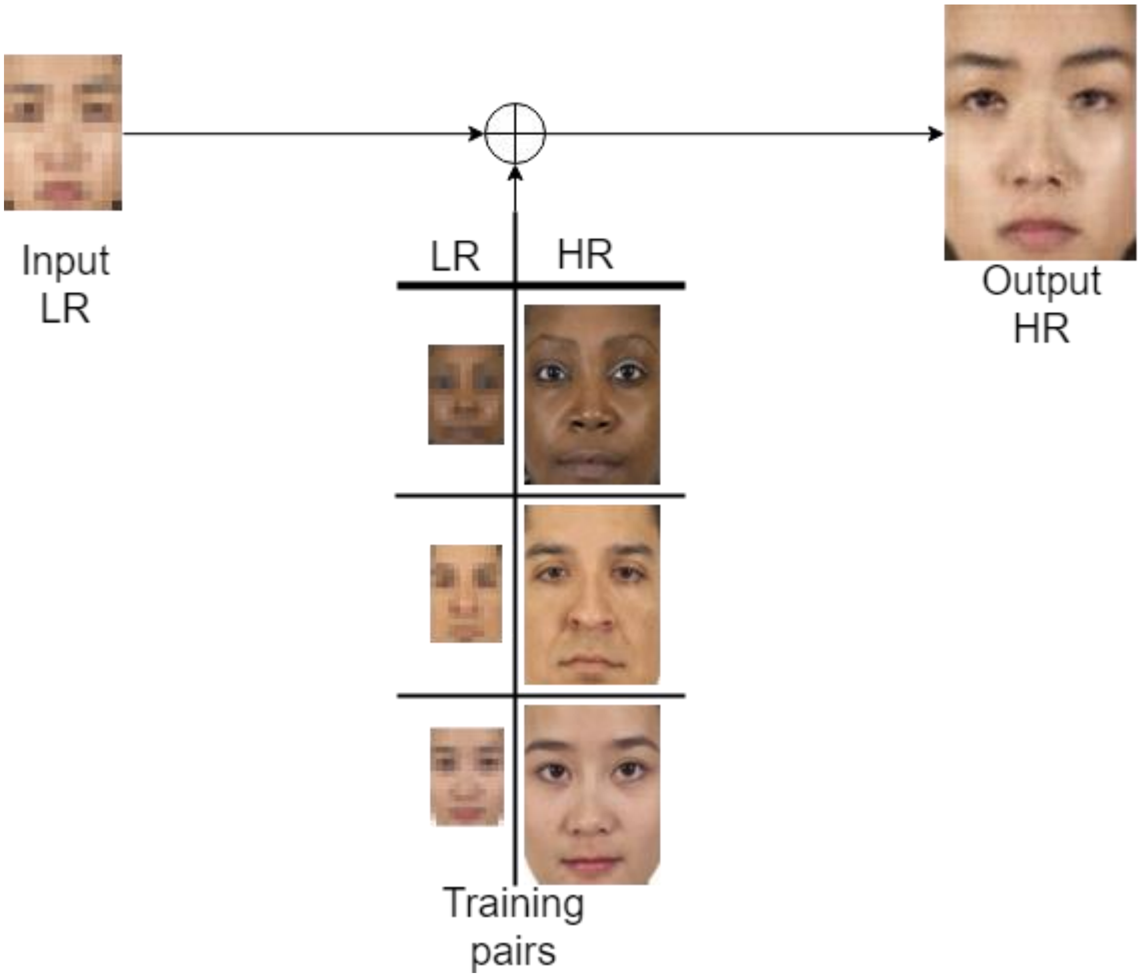
Polynomial-based interpolation





- Interpolation is the simplest method of upscaling an image using known data points
- Higher-order polynomial Interpolation maintains the continuity of adjacent pixels, but smooths the image

Example-based super-resolution



Contributions

- Implementation of machine learning and deep learning algorithms for facial image super-resolution using C++ with OpenCV library and Python with Pytorch library

- Modification and retraining the existing deep learning models for facial image super resolution

- Evaluation and comparison of the performance of different methods on different datasets, with different upscaling factors, down-sampling kernels, and noise levels

Problem formulation

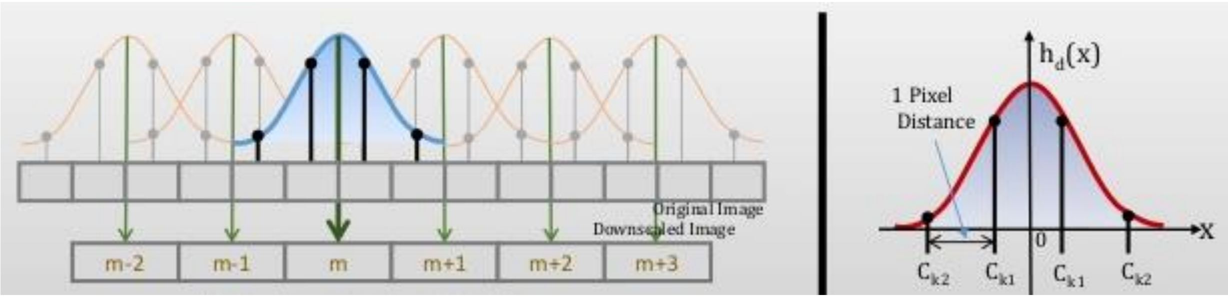
- General Approach:

Image $W \times H$ → LPF → Down-sampler $\downarrow$N → Image $\frac{W}{N} \times \frac{H}{N}$

- Practical Approach:

Image $W \times H$ → Down-sampling kernel → Image $\frac{W}{N} \times \frac{H}{N}$

- Kernels: nearest neighbour, bilinear & bicubic kernels



Datasets

- Chicago face database
  - 200 images for training, 20 images for testing
  - Face alignment is done using dlib and cropped into the size of 96x128

- LFW face database
  - 10, 000 images for training, 20 images for testing
  - Face alignment is done using dlib and cropped into the size of 128x128

Measurements

- Peak signal to noise Ratio (PSNR)

$$PSNR = 10log_{10}\left(\frac{MAX^2}{MSE}\right)$$

where $\mathrm{MSE} = \frac{1}{\mathrm{mn}}\sum_{\mathrm{i}=0}^{\mathrm{m}-1}\sum_{\mathrm{j}=0}^{\mathrm{n}-1}[\mathrm{I}(\mathrm{i},\mathrm{j}) - \mathrm{k}(\mathrm{i},\mathrm{j})]^2$

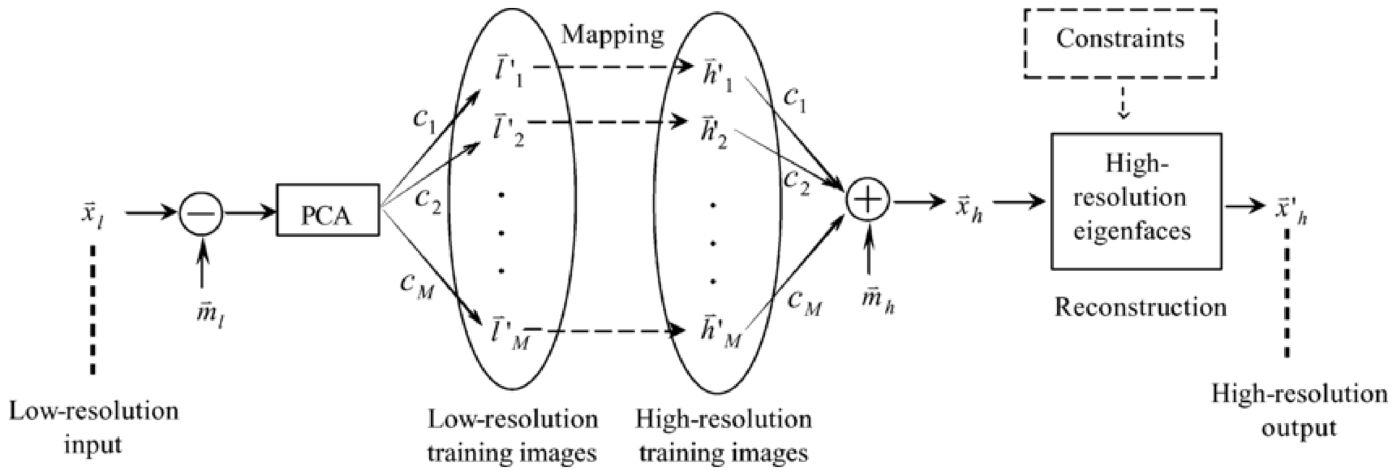
- Structural Similarity index (SSIM)

$$\mathrm{SSIM}\ (\mathrm{x},\mathrm{y}) = \frac{2(m_x m_y + C_1)(2\sigma_{xy} + C_2)}{\left(m_x{}^2 + m_y{}^2 + C_1\right)\left(\sigma_x{}^2 + \sigma_y{}^2 + C_1\right)}$$

Algorithm 1 – Eigentransformation (PCA)

Algorithm 1 – Eigentransformation [1]

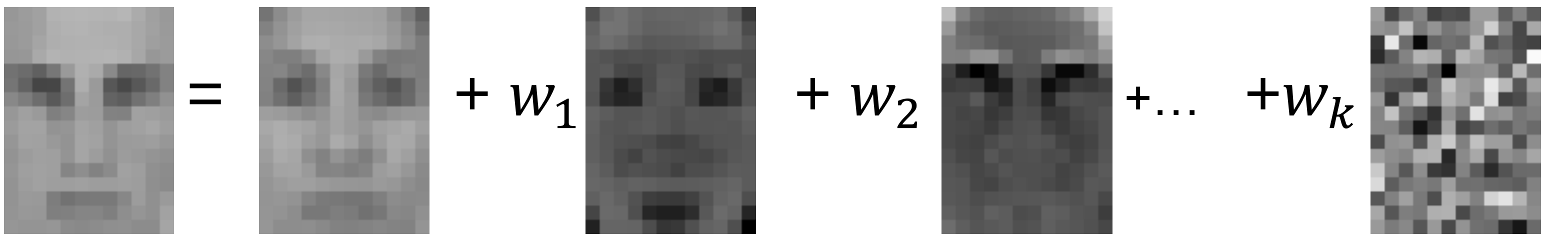
- This method is a transformation based on mapping between LR and HR groups of training samples
- The input LR image could be represented by a linear combination of LR training samples
- Keeping all the coefficients, the LR training samples are replaced by the corresponding HR ones



[1] X. Wang, etc., "Hallucinating face by eigentransformation, " *IEEE Transactions on Systems, Man, and Cybernetics*, 2005.

Algorithm 1 – Eigentransformation

- By principle component analysis (PCA), the training samples could be projected onto a subspace that the set of basis vectors is linearly uncorrelated
- The input LR image is projected onto that subspace



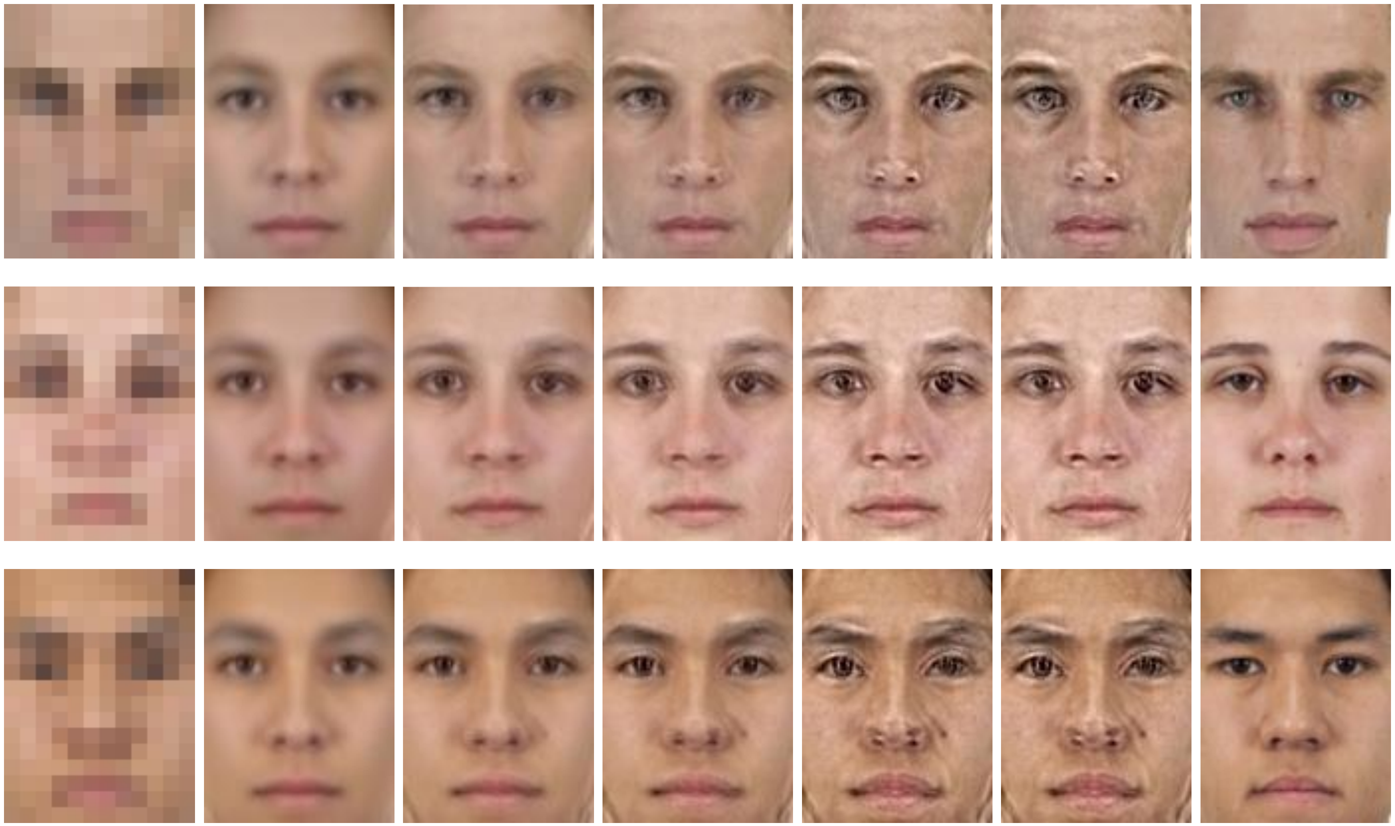
- Add the constraints to the weights w_i

$$\widehat{\vec{w}_i} = \begin{cases} \vec{w}_i, & |\vec{w}_i| < \alpha\sqrt{\lambda_i} \\ sign(\vec{w}_i) * \alpha\sqrt{\lambda_i}, & |\vec{w}_i| \geq \alpha\sqrt{\lambda_i} \end{cases}$$

where α is a positive parameter

- Compute the set of coefficients $\vec{c}_i$ from $\widehat{\vec{w}_i}$ and eigenfaces

Result 1 – Eigentransformation



	LR	α = 0.01	α = 0.05	α = 0.1	α = 0.5	α = INF	HR
PSNR (dB)		24.08	25.33	24.60	23.38	23.38	INF

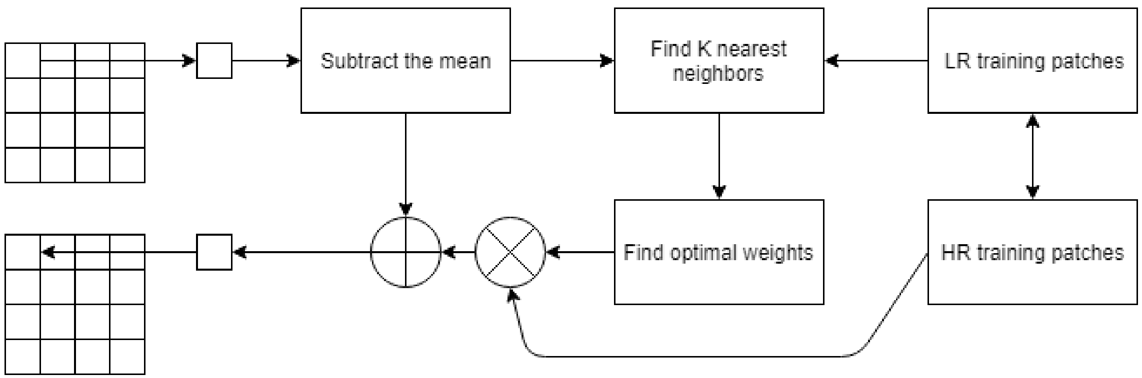
$$\widehat{\vec{w}_i} = \begin{cases} \vec{w}_i, & |\vec{w}_i| < \alpha\sqrt{\lambda_i} \\ sign(\vec{w}_i) * \alpha\sqrt{\lambda_i}, & |\vec{w}_i| \geq \alpha\sqrt{\lambda_i} \end{cases}$$

THE HONG KONG POLYTECHNIC UNIVERSITY
香港理工大學

Algorithm 2 – Neighbour Embedding (LLE)

Algorithm 2 – Neighbour Embedding [2]

- Locally linear embedding (LLE) is one of the manifold learning (nonlinear dimensionality reduction) methods
- By neighbourhood-preserving embeddings of high-dimensional inputs, the nonlinear structure is recovered from locally linear fits



- For each input image patch $\boldsymbol{x}_t^q$, find the optimal weights w_i^q for nearest neighbours $\boldsymbol{x}_i^q$ that minimize the reconstruction error ε^q

$$\varepsilon^q = \left\| \boldsymbol{x}_t^q - \sum_{i=1}^{k} w_i^q \boldsymbol{x}_i^q \right\|_2^2$$

$$\text{s.t. } \sum_{i=1}^{k} w_i^q = 1$$

[2] H. Chang, etc., "Super-resolution through Neighbour Embedding, " *IEEE Conference on Computer Vision and Pattern Recognition*, 2004.

Algorithm 2 – Neighbour Embedding

- Express matrix $\boldsymbol{X} = [x_1^q, \boldsymbol{x}_2^q, \ldots, \boldsymbol{x}_k^q]$, and the Gram matrix $G_q$as:

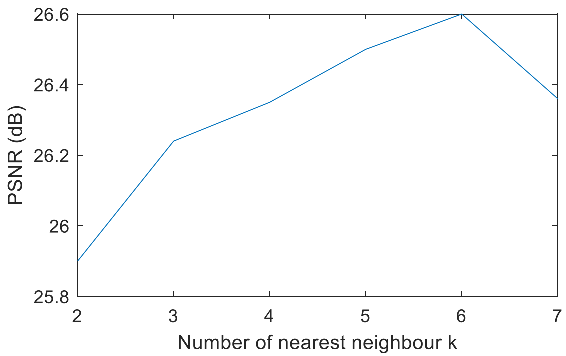
$$\boldsymbol{G}_q = (\boldsymbol{x}_t^q \boldsymbol{1}^{\boldsymbol{T}} - \boldsymbol{X})^T (\boldsymbol{x}_t^q \boldsymbol{1}^{\boldsymbol{T}} - \boldsymbol{X})$$

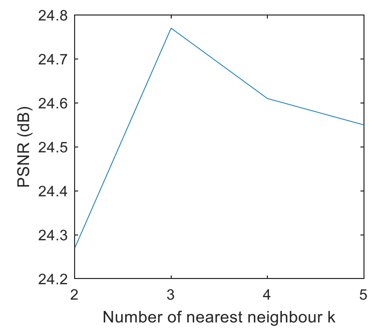
- The objective function could be solved by

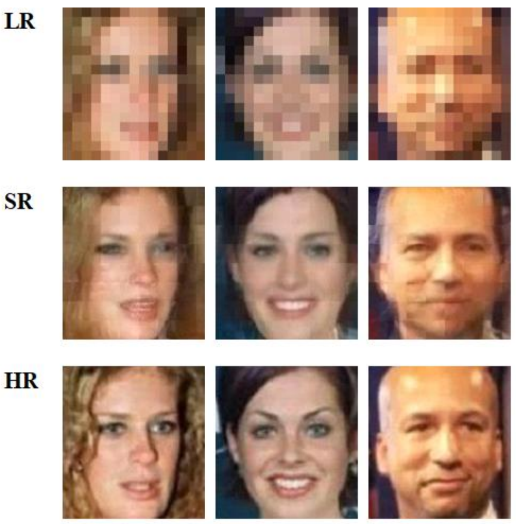
$$\boldsymbol{w}_q = \frac{\boldsymbol{G}_q{}^{-1}\boldsymbol{1}}{\boldsymbol{1}^{\boldsymbol{T}}\boldsymbol{G}_q{}^{-1}\boldsymbol{1}}$$

Result 2 – Neighbour Embedding

- Effect of the number of nearest neighbours k for different datasets



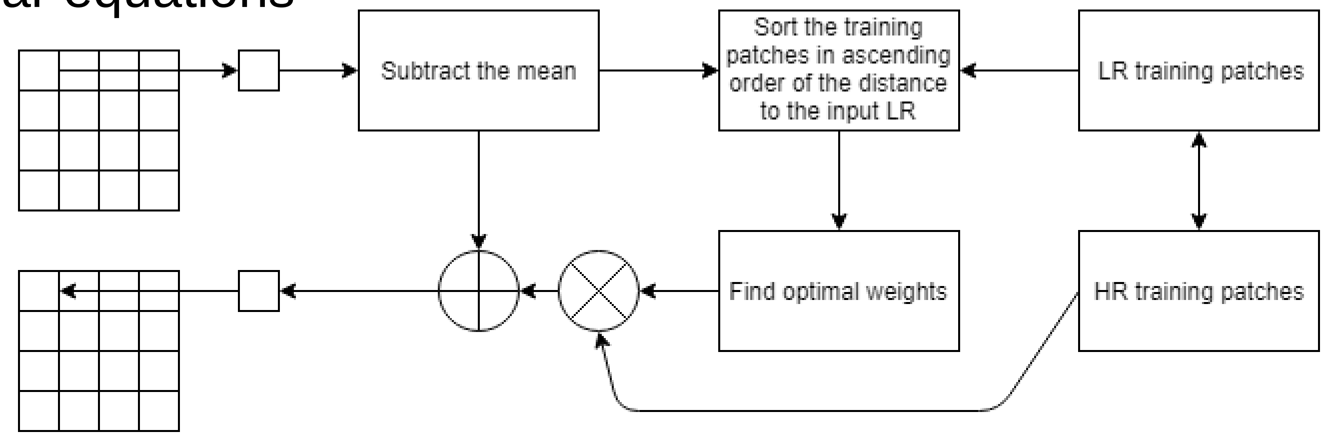




Algorithm 3 – Sparse Representation (SR)

Algorithm 3 – Sparse Representation [3] [4]

- Sparse representation deals with sparse solutions for systems of linear equations



- For each input image patch $\boldsymbol{x}_t^q$, find the optimal weights w_i^q for training image patches $\boldsymbol{x}_i^q$ that minimize the reconstruction error ε^q

$$\varepsilon^q = \left\| \boldsymbol{x}_t^q - \sum_{i=1}^{M} w_i^q \boldsymbol{x}_i^q \right\|_2^2$$

$$\text{s.t. } \sum_{i=1}^{M} w_i^q < \varepsilon_1 \text{ and } \sum_{i=2}^{M} w_i^q - w_{i-1}^q < \varepsilon_2$$

[3] J. Yang, etc., "Image Super-Resolution Via Sparse Representation, " *IEEE Transactions on Image Processing*, May 2010.

[4] J. Jiang, etc., "Noise Robust Face Image Super-Resolution Through Smooth Sparse Representation, " *IEEE Transactions on Cybernetics*, Nov 2017.

Algorithm 3 – Sparse Representation

- Rewrite the optimization problem into the Lagrange multiplier form:

$$\min \left\| \boldsymbol{x}_t^q - \sum_{i=1}^{M} w_i^q \boldsymbol{x}_i^q \right\|^2 + \lambda_1 \|\boldsymbol{w}^q\|_1 + \lambda_2 \sum_{i=2}^{M} \left\| w_i^q - w_{i-1}^q \right\|_1$$

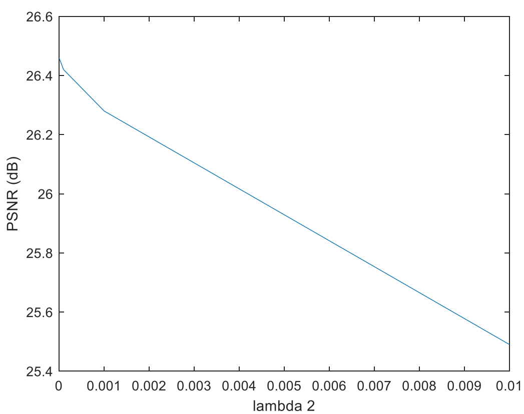
where λ_1 and λ_2 are non-negative parameters

- The least square is smooth while the regularized terms are non-smooth
- It can be solved by the Fast Iterative Shrinkage-Thresholding Algorithm (FISTA) [5]

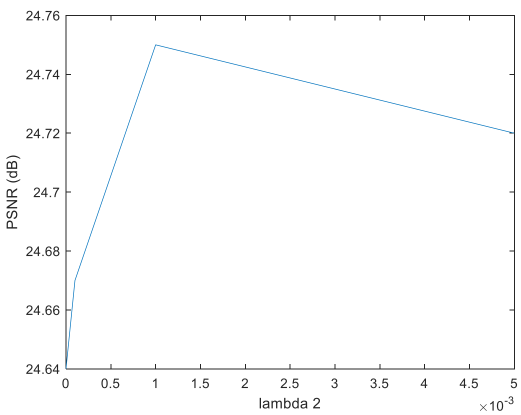
[5] A. Beck, "A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems, " *Imaging Sciences*, 2009.

Result 3 – Sparse Representation

- Effect of parameter λ_2



Noiseless images



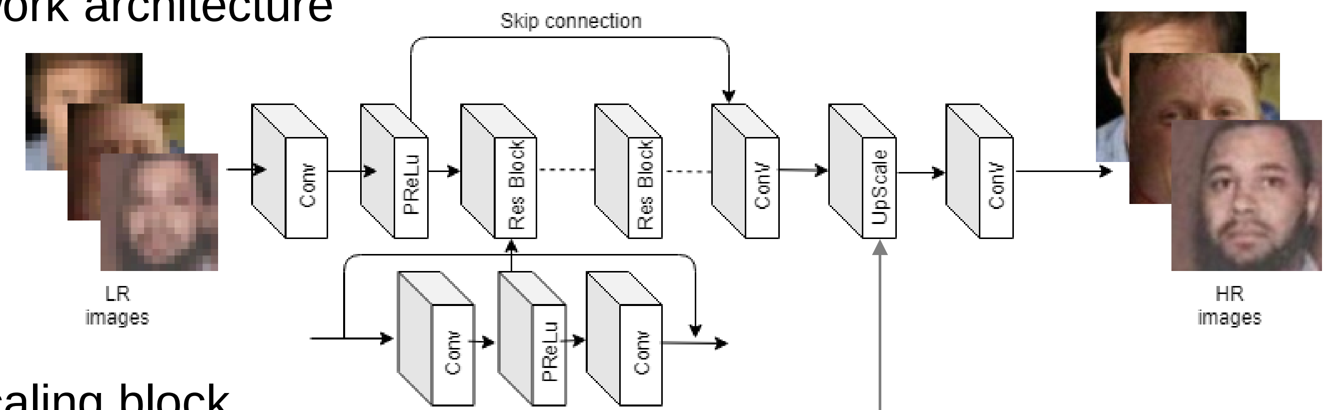
Noisy images
with white noise (σ = 0.05)

$$\min\left\|\boldsymbol{x}_t^q - \sum_{i=1}^{M} w_i^q \boldsymbol{x}_i^q\right\|^2 + \lambda_1 \|\boldsymbol{w}^q\|_1 + \lambda_2 \sum_{i=2}^{M} \left\|w_i^q - w_{i-1}^q\right\|_1$$

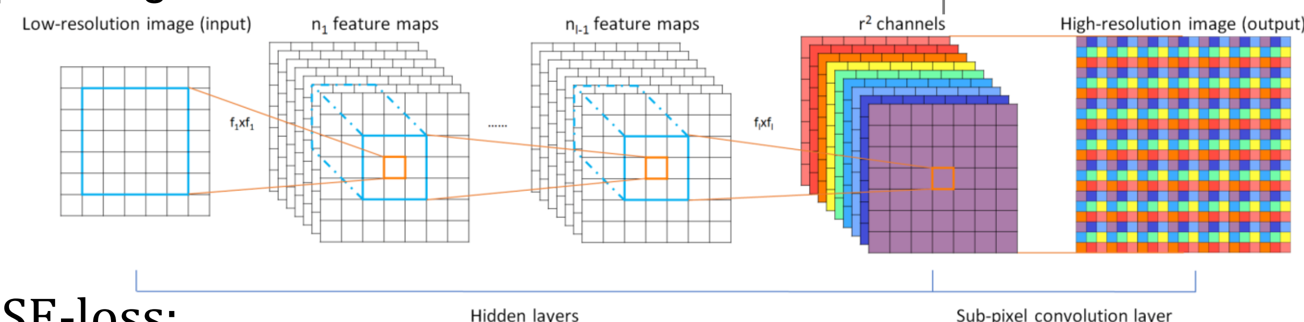
Algorithm 4 –
Convolutional Neural Network (CNN)

Algorithm 4 – Convolutional Neural Network (CNN) [6]

- Network architecture



- Upscaling block



- MSE-loss:

$$l_{mse} = \frac{1}{WH}\sum_{x=1}^{W}\sum_{y=1}^{H}\left(I_{x,y}^{HR} - G\left(I_{x,y}^{LR}\right)\right)^2$$

learning rate: 0.0001
batch size: 2
number of epochs: 200

[6] B. Lim, etc., "Enhanced Deep Residual Networks for Single Image Super-Resolution, " IEEE Conference on Computer Vision and Pattern Recognition Workshop, 2017.

Results 4 – CNN with different down-sampling kernels

down-sampling kernels of training images

down-sampling kernels of testing images

	Bicubic	Bilinear	Nearest	Mix
Bicubic	28.49 / 0.821	26.80 / 0.800	27.46 / 0.863	28.31 / 0.820
Bilinear	27.39 / 0.797	28.45 / 0.818	25.04 / 0.738	28.44 / 0.821
Nearest	21.75 / 0.677	19.04 / 0.570	26.39 / 0.801	25.54 / 0.788
Average	**25.88 / 0.765**	**25.06 / 0.729**	**26.30 / 0.800**	**27.43 / 0.810**

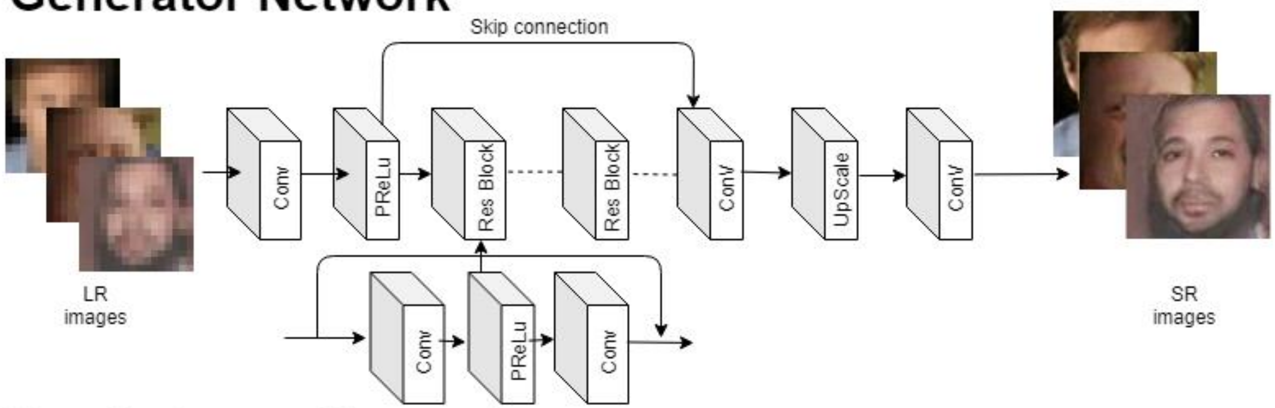
number of training images: 10, 000

number of testing images: 20

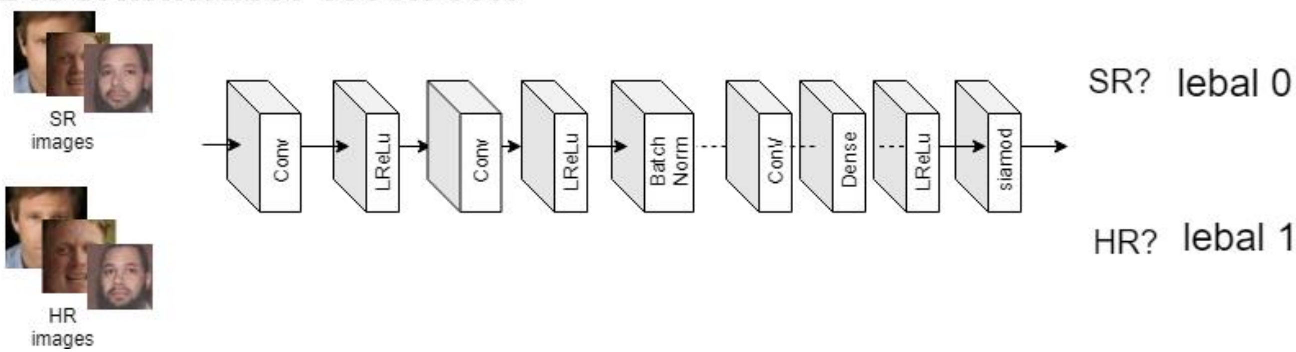
Algorithm 5 –
Generative Adversarial Network (GAN)

Algorithm 5 – Generative Adversarial Network (GAN) [7]

Generator Network



Discriminator Network



learning rates: 0.0001
batch size: 2
number of epochs: 200

- The generator network fools the discriminator that the generated image is natural
- The discriminator network distinguishes that the image is a natural or generated image

[7] C. Ledig, etc., "Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network, " IEEE Conference on Computer Vision and Pattern Recognition, Aug 2017.

Algorithm 5 – Generative Adversarial Network (GAN)

- Perceptual loss
  - $l_{total} = l_{vgg} + 0.006\, l_{adversairal}$

$$l_{vgg} = \frac{1}{WH} \sum_{x=1}^{W} \sum_{y=1}^{H} (V\left(I_{x,y}^{HR}\right) - V\left(G\left(I_{x,y}^{LR}\right)\right))^2$$

where V(I) is the output of the middle of the VGG-network

$$I_{adversairal} = \sum_{n=1}^{N} -\log D(G(I^{LR}))$$

where $D(G(I^{LR}))$ is the probability that the discriminator predicts the generated image is a natural image

Overall Results – upscaling factor 8

	BC	PCA	LLE	SR	CNN	GAN	HR
PSNR (dB)	24.57	25.33	26.24	26.34	27.96	27.36	INF
SSIM	0.712	0.744	0.784	0.794	0.829	0.803	1

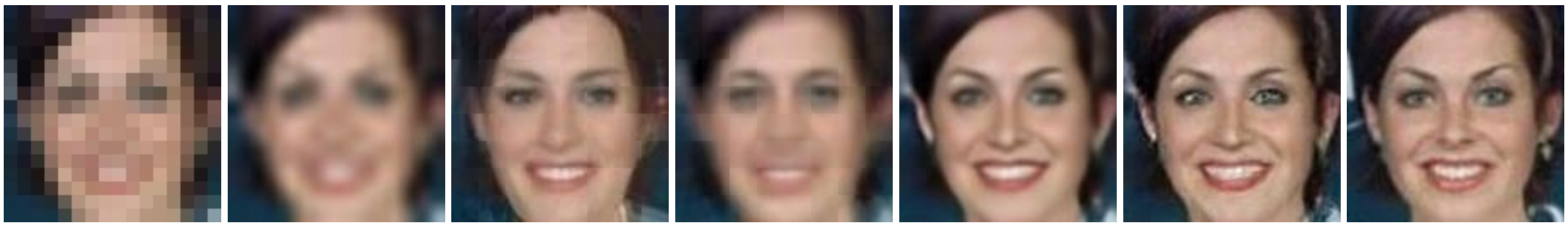
number of training images: 200
number of testing images: 20

Overall Results – upscaling factor 4

	BC	PCA	LLE	SR	CNN	GAN	HR
PSNR (dB)	28.49	27.96	29.28	29.37	32.07	30.73	INF
SSIM	0.853	0.808	0.856	0.860	0.908	0.890	1

number of training images: 200
number of testing images: 20

Overall Results – unconstrainted dataset



	BC	LLE	SR	CNN	GAN	HR
PSNR (dB)	24.75	24.77	25.22	28.32	27.06	INF
SSIM	0.689	0.691	0.711	0.820	0.773	1

number of training images: 10, 000
number of testing images: 20

Overall Results – noise performance

σ = 0.01

σ = 0.05

	BC	PCA	LLE	SR	CNN	GAN
PSNR (dB) ($\sigma = 0.01$)	24.51	24.59	25.77	26.10	27.94	27.30
PSNR (dB) ($\sigma = 0.05$)	23.65	24.02	25.18	25.77	25.83	25.56
dB drops	0.86	0.57	0.59	0.33	2.11	1.74

number of training images: 200
number of testing images: 20

Noisy Image *LR* → Image denoising& super-resolution → Noiseless image HR

Conclusion

- Convolutional Neural Networks (CNNs) achieve the best performance in terms of PSNR and SSIM.
- CNNs are very sensitive to the down-sampling kernels used to generate the input LR images.
- Sparse representation has less dB drops in terms PSNR when noise increases.

Future work

1. Image super-resolution for noisy inputs

- My current approach:

Noisy Image LR → Image denoising& super-resolution → Noiseless image HR

- Other possible approaches:

Noisy Image LR → Image denoising → Image super-resolution → Noiseless image HR

Noisy Image LR → Image super-resolution → Image denoising → Noiseless image HR

Future work

2. Image degradation in real case

References

- [1] X. Wang& X. Tang, "Hallucinating face by eigentransformation, " *IEEE Transactions on Systems, Man, and Cybernetics*, Jul 2005.
- [2] H. Chang, D. Yeung& Y. Xiong, "Super-resolution through Neighbour Embedding, " *IEEE Conference on Computer Vision and Pattern Recognition*, 2004.
- [3] J. Yang, J. Wright& T. S. Huang, "Image Super-Resolution Via Sparse Representation, “ *IEEE Transactions on Image Processing*, May 2010.
- [4] J. Jiang, J. Ma& C. Chen, "Noise Robust Face Image Super-Resolution Through Smooth Sparse Representation, “ *IEEE Transactions on Cybernetics*, Nov 2017.
- [5] A. Beck and M. Teboulle, "A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems, " *Imaging Sciences*, 2009.

References

- [6] B. Lim, S. Son, H. Kim, S. Nah, K. Mu Lee, "Enhanced Deep Residual Networks for Single Image Super-Resolution, " IEEE Conference on Computer Vision and Pattern Recognition Workshop, 2017.
- [7] C. Ledig, L. Theis, F. Huszar, "Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network, " IEEE Conference on Computer Vision and Pattern Recognition, Aug 2017.