Machine Learning for Facial Image Super-resolution

CHEUNG Tsun Hin

Supervisor: Prof. Kenneth LAM

Department of Electronic and Information Engineering

Date: 11 January 2019

What is Image Super-resolution?

- Image super resolution (SR) is the technique that estimates the high resolution (HR) image from a low resolution (LR) image.

25x25 Input (LR) | 200x200 Output (SR) | 200x200 Ground Truth (HR)

Why do we need Super-resolution?

- Medical usage
  - example: computed tomography (CT)

- Video surveillance
  - example: low resolution face recognition

Methods of Super-resolution

- Interpolation-based
  - Upscaling the image by using the known data points **without prior knowledge**.
  - Example: Bicubic interpolation

- Example-based
  - Estimating the HR image by **learning the training pairs** of LR and HR samples.

Problem Formulation

- The LR image $\vec{I}_l$ is generated from the HR $\vec{I}_h$ one by the linear model:

$$\vec{I}_l = H\vec{I}_h + \vec{n}$$

where *H* is the operation matrix involving blurring and downsampling. $\vec{n}$ is the random distribution added during image acquisition.

Equipment

- Windows 10 64bit
  - Microsoft Visual C++ 2017 with OPENCV library
  - MATLAB R2017b
- Ubuntu 18.04
  - Python 3.6 with PYTORCH library

Datasets

- Face image database
  - Chicago face database
  - This dataset consists of 597 facial images
  - 200 for training and 20 for testing
- Face alignment
  - All face images have been detected and aligned according to the eyes position using opencv and dlib before experiments. Then, the images are cropped into size of 168x200

Algorithms

- Conventional machine learning algorithms
  - Eigen transformation (Principal component analysis)
  - Neighbour embedding (Locally linear embedding)
  - Sparse representation
- Deep learning algorithms
  - Convolutional neutral network (CNN)
  - Generative adversarial network (GAN)

Progress

Work Done	Time
Implementation of Eigen transformation	Sep 2018
Implementation of Neighbour Embedding and Sparse Representation	Oct 2018
Measurements on the Eigen transformation, Neighbour Embedding and Sparse Representation	Nov 2018
Implementation of deep learning approaches	Dec 2018

Algorithm 1
Eigen Transformation

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

Algorithm - Eigen transformation

- Denote each training image as N-dimensional vector $\vec{l}_i$ and Group them into a $\mathrm{N} \times \mathrm{M}$ matrix

$$[\vec{l}_1, \vec{l}_2 \ldots \vec{l}_M]$$

  N is the number of pixels in the image

  M is the number of training samples

- Compute the mean face

$$\overrightarrow{m}_l = \frac{1}{M} \sum_{i=1}^{M} \vec{l}_i$$

- Subtract the mean from each of the training image

$$L = [\vec{l}_1 - \overrightarrow{m}_l, \vec{l}_2 - \overrightarrow{m}_l, \ldots \vec{l}_M - \overrightarrow{m}_l]$$

Algorithm - Eigen transformation

- Compute the eigenvectors of the covariance matrix LL^T
- Since N >> M and LL^T has at most M eigenvectors, the eigenvectors of the matrix $L^T L$ could be first computed

$$(L^T L)V = V\lambda$$

- Multiply both sides by L, we have

$$(LL^T)LV = LV\lambda$$

- $LV\lambda^{-\frac{1}{2}}$ are the orthonormal eigenvectors of the covariance matrix LL^T

Algorithm - Eigen transformation

- Compute the weight vector $\overrightarrow{w_l}$ by projecting the input LR image $\vec{x}_l$ onto the eigenvectors:

$$\overrightarrow{w_l} = (LV\lambda^{-\frac{1}{2}})^T(\vec{x}_l - \overrightarrow{m_l})$$

- The reconstructed LR image is:

$$\vec{r}_l = \left(LV\lambda^{-\frac{1}{2}}\right)\overrightarrow{w_l} + \overrightarrow{m_l}$$

Algorithm - Eigen transformation

- Denote $\vec{c}_l = V\lambda^{-\frac{1}{2}}\overrightarrow{w}_l$, we have

$$\vec{r}_l = L\vec{c}_l + \overrightarrow{m}_l$$

- Express $\vec{c}_l = [c_1, c_2, \ldots c_M]^T$, we have

$$\vec{r}_l = \sum_{i=1}^{M} c_i \vec{l}_l + \overrightarrow{m}_l$$

Algorithm - Eigen transformation

- Replace each LR sample $\vec{l}_l$ by HR sample $\vec{l}_h$ and mean face LR $\overrightarrow{m}_l$ by mean face HR $\overrightarrow{m}_h$

$$\vec{x}_h = \sum_{i=1}^{M} c_i \vec{l}_h + \overrightarrow{m}_h$$

- $\vec{x}_h$ is the output super resolution image

Algorithm 2
Neighbour Embedding

[2] H. Chang, D. Yeung& Y. Xiong, "Super-resolution through Neighbour Embedding, " *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2004.

Algorithm – Neighbour Embedding

- Denote the overlapping image patches of input image as $\boldsymbol{x}_t^q$ and that of training images as $\boldsymbol{x}_i^q$.

- Find the k nearest neighbours that have shortest square distance to the input image

- Find each weight w_i^q that best reconstruct the image patch by minimizing the error ε^q

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

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

Algorithm – Neighbour Embedding

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

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

- The objective function could be solved by

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

Algorithm – Neighbour Embedding

- The reconstructed LR image patch $\boldsymbol{r}_t^q$ is:

$$\boldsymbol{r}_t^q = \sum_{i=1}^{k} w_i^q \boldsymbol{x}_i^q$$

- Replace the LR training patches $\boldsymbol{x}_i^q$ by HR ones $\boldsymbol{y}_i^q$

$$\boldsymbol{y}_t^q = \sum_{i=1}^{k} w_i^q \boldsymbol{y}_i^q$$

- Construct the HR image by the patches $\boldsymbol{y}_t^q$ using averaging in the overlapping region

Algorithm 3
Sparse Representation

[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.

Algorithm – Sparse Representation

- Denote the overlapping image patch of input image as $\boldsymbol{x}_t^q$ and that of training image as $\boldsymbol{x}_i^q$.

- Find the weight w_i^q that best reconstructs the image patch by minimizing the error ε^q

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

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

Algorithm – 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 Fast Iterative Shrinkage-Thresholding Algorithm (FISTA) [5]

Algorithm – Sparse Representation

- The reconstructed LR image patch $\boldsymbol{r}_t^q$ is:

$$\boldsymbol{r}_t^q = \sum_{i=1}^{M} w_i^q \boldsymbol{x}_i^q$$

- Replace the LR training patches $\boldsymbol{x}_i^q$ by HR ones $\boldsymbol{y}_i^q$

$$\boldsymbol{y}_t^q = \sum_{i=1}^{M} w_i^q \boldsymbol{y}_i^q$$

- Construct the HR image by the patches $\boldsymbol{y}_t^q$ using averaging in the overlapping region

Result – Eigen transformation

M	**25**	**50**	**75**	**100**	**125**	**150**	**175**	**200**
PSNR (dB)	23.73	24.73	25.29	25.76	26.05	26.34	26.41	26.39

Table 1 the results using different number of training samples M

Too many training samples reduces the performance

Result – Neighbour Embedding

k	2	3	4
PSNR (dB)	26.73	26.73	26.68
SSIM	0.800	0.8027	0.8025

Table 2 the result using windows size of

5x5 and different values of k neighbours

k	2	3	4
PSNR (dB)	27.16	27.26	27.24
SSIM	0.798	0.806	0.809

Table 3 the result using windows size of

3x3 and different values of k neighbours

The values of k is optimal at 3 using 3x3 windows

Result – Sparse Representation

λ_1	**0.01**	**0.001**	**0.0001**	**0.00001**
PSNR (dB)	26.90	27.71	28.02	28.02
SSIM	0.793	0.828	0.838	0.838

Table 4 the result using values of $\boldsymbol{\lambda_1}$ and

$\boldsymbol{\lambda_2 = 0}$ for noiseless images

The values of λ_1 is optimal at 0.0001

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

Result – Sparse Representation

λ_2	**0.01**	**0.001**	**0.0001**	**0.00**
PSNR (dB)	26.99	27.78	27.98	28.02
SSIM	0.806	0.831	0.837	0.838

Table 5 the result $\boldsymbol{\lambda_1 = 0.0001}$ and different values $\boldsymbol{\lambda_2}$ for noiseless images

$\lambda_2 = 0$ is optimal when there is no noise

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

Result – Sparse Representation

$\boldsymbol{\lambda_2}$	**0.01**	**0.001**	**0.0001**	**0.00**
PSNR (dB)	26.25	26.22	26.17	26.14
SSIM	0.777	0.771	0.767	0.766

Table 6 the result $\boldsymbol{\lambda_1 = 0.0001}$ and different values $\boldsymbol{\lambda_2}$ for noisy testing (σ = 0.05)

A suitable values for λ_2 improves the performance when there is noise

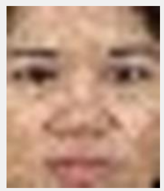
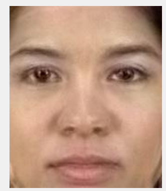
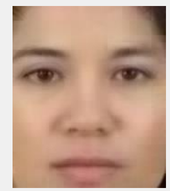
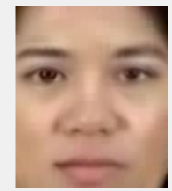
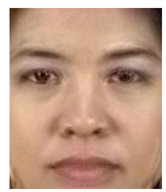
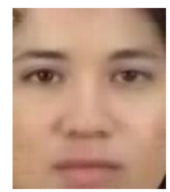
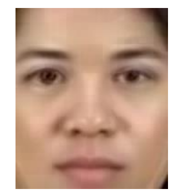
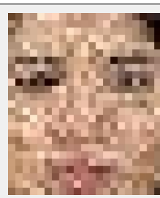
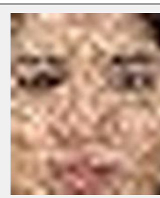
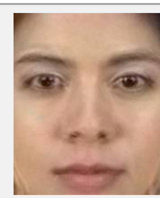
$$\min\left\|\boldsymbol{x}_t^q - \sum_{i=1}^{M} w_i^q \boldsymbol{x}_i^q\right\|^2 + \lambda_1\|\boldsymbol{w}^q\|_1 + \lambda_2 \sum_{i=2}^{M}\left\|w_i^q - w_{i-1}^q\right\|_1$$

Result

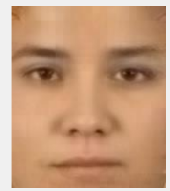
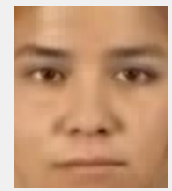
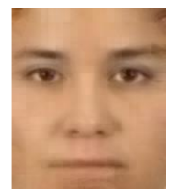
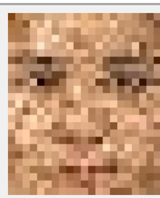
PSNR (dB) / SSIM Using Different algorithms with varying values of σ

σ	Bicubic	Eigen Transformation	Neighbour Embedding	Sparse Representation
0.02	25.29 / 0.729	25.38 / 0.706	25.76 / 0.763	26.60 / 0.779
0.05	23.86 / 0.665	24.94 / 0.707	25.04 / 0.747	26.46 / 0.778
0.1	20.00 / 0.500	23.90 / 0.708	22.85 / 0.691	24.78 / 0.750

Result

σ	LR	BC	ET	NE	SR	GT
0.02						
0.05						
0.1						

Result

σ	LR	BC	ET	NE	SR	GT
0.02						
0.05						
0.1						

Result

- Sparse representation achieves the best result whenever there is noise.

- Eigen transformation is more robust to noise than neighbour embedding

Cconclusion

- Interpolation is not robust to noise and makes the image unclear

- Examples-based algorithm recovers more details of the input image by learning the training samples

- Sparse representation achieves the best performance

Future Plan

Work	Time
Measurement of deep learning approaches	Jan 2019
Implementation of GUI system	Feb 2019
Discovering the effects of super resolution to face recognition	Mar 2019
Finalizing the project and report writing	Apr 2019

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 Computer Society 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.