The Hong Kong Polytechnic University

Department of Electronic and Information Engineering

Final Year Project

Interim Report

Machine Learning for Facial Image Super-resolution

Name: CHEUNG Tsun Hin

Student ID: 15083269D

Supervisor: Prof. Kenneth LAM

Abstract

In this project, conventional machine learning and recent deep learning methods for facial image super resolution are examined. In particular, the three state of arts conventional machine learning approaches include eigen transformation, neighbour embedding and sparse representation. The recent deep learning algorithms being considered are convolutional neural network and generative adversarial network. The conventional machine learning algorithms are implemented using c++ with the opencv library while the deep learning ones are in python with the pytorch library. However, only the measurements of the conventional machine learning approaches have been done. In the current stage, it is shown that the sparse representation achieves the best performance among the three machine learning algorithms.

1. Introduction

Single image super-resolution is to generate the high-resolution image given a single low-resolution one. This technique has many applications in the areas of image processing and computer vision, such as video surveillance and medical usage. In video surveillance, the face recognition rate drops significantly when the size of the facial image becomes smaller [1]. E. Bilgazyev et al. showed that the performance of face recognition of a low-resolution image could be increased by performing image super resolution [2].

Face Hallucination is a special case of image super resolution where the input is a facial image. The algorithms could be categorized in many ways, such as interpolation-based, reconstruction-based and example-based methods. However, the performance of interpolation-based and reconstruction-based methods is usually worse because they do not make use of the similarity and the structure of a human face [3]. Example-based method is to generate the high-resolution image by learning the training pairs of low and high-resolution image. In this project, majority of example-based methods of face hallucination is investigated.

In 2014, the first deep learning method solving image super-resolution was invented using Convolutional Neutral Network (CNN) [4]. It showed the performance of image super resolution is comparable to the other conventional methods. Afterwards, different deep learning methods are developed for solving single image super-resolution. Moreover, the SRGAN used the perceptual loss in the training process instead of Mean square error (MSE). The property of the perceptual loss will be examined. In this project, different deep learning approaches will be compared to the traditional approaches.

2. Background

2.1 Problem Formulation

The low-resolution images $\vec{I}_l$ is generated from its high-resolution images $\vec{I}_h$ by the following linear model:

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

where *H* is the operation matrix involving blurring and down-sampling and $\vec{n}$ is the random distribution added during image acquisition. The burring matrix is Gaussian filter. This assumption holds for all learning algorithms in the project. The noise added is white noise in this project. Our training samples and testing images are generated by this linear model.

2.2 Eigen Transformation

Eigen transformation for face hallucination was proposed by Wang and Tang [5]. This method is a direct application of the principal component analysis (PCA). This method relies on the fact that the face images have similarity, so the feature of the face images could be used for performing super-resolution. Previous studies showed that the high frequency details of facial image could be recovered by the low frequency component by given a training sets of facial images. This method tries to recover the high frequency component of the input image by learning the training samples.

Mathematical Analysis

First, denote each LR training face image as N-dimensional vector $\vec{l}_i$ and Group them into a N×M matrix.

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

Here, N is the number of pixels in the image and M is the number of training samples. Then, compute the mean face $\vec{m}_l$ as the following.

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

Then, each training face image is subtracted by the mean face $\vec{m}_l$ and denote the matrix as L

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

By PCA, the demean training image is now projected to the subspace that has the largest variance. It can be achieved by finding the eigenvectors of the convenience matrix of the demean vector, i.e. LL^T. However, since N is much larger than M generally, LL^T has at most M eigen vector, the eigenvectors of. $L^T L$ could be first calculated.

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

Here, V is the eigenvector matrix and λ is the diagonal eigenvalue matrix of matrix of $L^T L$. Multiplying both sides by L, we have

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

It is clear that $LV\lambda^{-\frac{1}{2}}$ are the orthonormal eigenvectors of the covariance matrix $L^T L$.

For a given LR input image $\vec{x}_l$, the weight vector $\vec{w}_l$could be computed by projecting the input image onto the eigenvectors above, we have

$$\vec{w}_l = (LV\lambda^{-\frac{1}{2}})^T(\vec{x}_l - \vec{m}_l)$$

The reconstructed $\vec{r}_l$ LR image is

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

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

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

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

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

Replace each LR sample $\vec{l}_l$ by its HR image $\vec{l}_h$ and the LR mean face $\vec{m}_l$ by HR mean face $\vec{m}_h$, we have

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

$\vec{x}_h$ is the generated super resolution image.

2.3 Neighbour Embedding

Neighbour embedding for image super resolution is proposed by Chang, Yeung and Xiong in 2004 [6]. This method is based on one of the manifold learning algorithms called locally linear embedding (LLE). Specially, the method predicts the output by learning the linear relationship of the input with the training data that has the closed geometric distance from the input. This algorithm relies on the fact that the low-resolution image has similar linear structure if the images are closed in geometric distance.

Mathematical Analysis

First, denote the image patch of input image as x_t^q and that of training image as x_i^q. For each image patch of input image, find the k nearest neighbours x_i^q that are closed to input patch x_t^q. Compute the weights w_i^q for each neighbour that best reconstruct the input patch and express the reconstruction error as ε^q, we have

$$\varepsilon^q = \left\| x_t^q - \sum_{i=1}^{k} w_i^q x_i^q \right\|^2$$

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

To solve the above object function, we first express matrix $X = [x_1^q, x_2^q, \dots, x_k^q]$, and the Gram matrix G_q as:

$$G_q = (x_t^q \mathbf{1}^T - \boldsymbol{X})^T (x_t^q \mathbf{1}^T - \boldsymbol{X})$$

The above constrained object function can be solved by

$$G_q w_q = \mathbf{1}$$

And then normalize the weights such that the sum of the weights is equal to one.

$$\sum_{i=1}^{k} w_i^q = 1$$

The reconstructed LR image r_l is

$$r_l = w_q X$$

Expand the above equation, we have

$$r_l = \sum_{i=1}^{k} x_i^q w_i^q$$

Replace the low-resolution image patch by its high-resolution patch, we have

$$y_h = \sum_{i=1}^{k} y_i^q w_i^q$$

2.4 Sparse Representation

Sparse representation for image super resolution was proposed by Yang in 2008 [67. The method represents the input LR image as a sparse linear combination of the raw low-7resolution images. Afterwards, the author further proposed that the over-completed sparse dictionary could be prepared instead of the raw images [8]. [9] Jiang proposed a smooth sparse representation for face hallucination based on sparse representation with the addition of the constraints. This project directly applies the smooth sparse representation. However, the smooth sparse is not effective when there is no noise.

Mathematical Analysis

The object function of this method is to find the weights for each raw image patch that minimize the reconstruct error subject to the constraints

$$\varepsilon^q = \left\| x_t^q - \sum_{i=1}^{M} w_i^q x_i^q \right\|^2$$

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

Rewrite the object function into the Lagrange multiplier form, we have

$$\text{Min } \left\| x_t^q - \sum_{i=1}^{M} w_i^q x_i^q \right\|^2 + \lambda_1 \|w^q\| + \lambda_2 \left\| w_i^q - w_{i-1}^q \right\|$$

Where λ_1 and λ_2 are non-negative parameters.

It is noticed that the object function and the constraints are both convex. The object function smooth but the constrains are not. This means that the constraints function is not differentiable for all the domain.

The above is a constrained least square problem called Fused Lasso [10]. There are many existing algorithms that solve the optimization problem. This project uses existing library and the algorithm used is called Fast Iterative shrinkage-thresholding algorithm FISTA [11].

2.5 Convolutional Neutral Network

Convolutional Neutral Network (CNN) is found to have many applications in image processing and computer vision in recent year. In 2014, the first CNN used for solving image super resolution was invented by Dong [4]. In the paper of SRCNN, the authors used 3 layers and the input image has been upscaling using bicubic interpolation. Later, the authors showed that the upscaling process could be neglected by adding a deconvolutional layer as the output layer and named the method FSRCNN [12]. Afterward, different variations of CNN have been proposed. In this project, the FSRCNN is considered and modifications are made on top of it.

2.6 Generative Adversarial Network

The second neutral network approach for this project is the generative adversarial network (GAN) [13]. This network consists of a pair of networks, a generator network and a discriminator network. The generator network generates the high-resolution image from the low-resolution input. The discriminator network classifies whether the image is generated super-resolution or a neutral high-resolution image. By training the pair of networks simultaneously, the generator network generates a photo realistic super-resolution image.

3. Methodology

3.1 Environment

The eigen transformation, neighbour embedding, and sparse representation algorithms are implemented using c++ with the opencv library. All experiments are done in windows 10 64bit system. With the help of MATLAB R2017b, the performance of the algorithms is measured. For the deep neutral network part, the algorithms are implemented in python with pytorch library. The codes are run in linux ubuntu 18.04.

3.2 Dataset

- CHICAGO face database

This dataset consists of 597 facial images. 200 of them are randomly chosen for training samples and 20 of them are randomly selected for testing. Before conducting image super resolution experiments, the face images have been aligned using existing library in dlib and opencv according to the 68 key points landmarks of the human face. The images are further cropped into size of 168×200.

3.3 Experiment Procedures

i. Low pass filter the training samples and testing images

ii. Down-sample the training samples and testing images

iii. Add the noise by varying the value of σ**.**

iv. Super resolute the testing images using the algorithms by learning the training samples

v. Measure the performance of algorithms by measuring the peak signal to noise ratio (PSNR) and structural similarity index (SSIM) between the generated super-resolution image and the ground truth high resolution image

3.4 Algorithms

Eigen transformation

1. Subtract the input LR image from the mean face of the LR training samples

$$\vec{x}_l - \vec{m}_l$$

2. Find the coefficients that best project the image onto the subspace of demean training samples using PCA

$$\vec{c}_l = V\lambda^{-\frac{1}{2}}\vec{w}_l$$

3. Map the LR training sample with the HR training sample
4. Multiply and add the coefficients and the corresponding high-resolution images

$$\sum_{i=1}^{M} c_i \vec{l}_h$$

5. Add the result with the mean face of the HR training samples

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

Neighbour Embedding

1. Divide the input image and training samples into overlapping patches
2. For each input patch

   2.1 Subtract the mean from each patch to obtain the feature vectors

   2.2 Sort the feature vectors of the training samples in ascending order of the square distance to the input feature vector

2.3 Find the k nearest neighbour of training patches that closed to the input patch in terms of square distance of the feature

2.4 Compute the weights of those training patches that best reconstruct the input patch

$$G_q w_q = \mathbf{1}$$

Where $G_q = (x_t^q \mathbf{1}^T - \boldsymbol{X})^T (x_t^q \mathbf{1}^T - \boldsymbol{X})$ and $\sum_{i=1}^{k} w_i^q = 1$

2.5 Add the reconstructed patch with the input mean

2 Reconstruct the SR image using the image patches. Averaging the overlapping pixels.

Sparse Representation

1. Divide the input image and training samples into overlapping patches
2. For each input patch

2.1 Subtract the mean from each patch to obtain the feature vectors

2.2 Sort the feature vectors of the training samples in ascending order of the square distance to the input feature vector

2.3 Compute the weights of training patches that best reconstruct the input patch

$$\text{Min } \left\| x_t^q - \sum_{i=1}^{M} w_i^q x_i^q \right\|^2 + \lambda_1 \|w^q\| + \lambda_2 \left\| w_i^q - w_{i-1}^q \right\|$$

2.4 subtract the mean from each patch to obtain the feature vectors

3. Reconstruct the SR image using the image patches. Averaging the overlapping pixels.

3.5 Parameters setting

Neighbour embedding:

k – number of the training features selected that are closed to the input features

s - size of the patch

l - size of the overlapping region

without specified, the default setting is k = 4, s = 3, l = 1

Sparse Representation:

λ_1 – positive scalar for constraint $\|w^q\|$

λ_2 – positive scalar for constraint $\left\|w_i^q - w_{i-1}^q\right\|$

s - size of the patch

l - size of the overlapping region2

without specified, the default setting is λ_1 = 0.0001, λ_1 = 0.0, s = 3, l = 1

4. Result

4.1 Eigen transformation

k	**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
SSIM	0.6914	0.7077	0.7175	0.7251	0.7319	0.7398	0.7398	0.7390

Table 1 the results using different number of training samples k

4.2 Neighbour Embedding

k	**2**	**3**	**4**	**5**
PSNR (dB)	26.73	26.73	26.68	26.57
SSIM	0.7997	0.8027	0.8025	0.8015

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

4.3 Sparse Representation

Windows Size	**3x3**	**5x5**
PSNR (dB)	27.92	27.71
SSIM	0.835	0.828

Table 4 the result using different windows size

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 5 the result using values of λ_1 and $\lambda_2 = 0$ for noiseless images

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 6 the result $\lambda_1 = 0.0001$ and different values λ_2 for noiseless images

λ_2	**0.001**	**0.005**	**0.0001**	**0.00**
PSNR (dB)	26.25	26.22	26.17	26.14
SSIM	0.777	0.771	0.767	0.766

Table 7 the result $\lambda_1 = 0.0001$ and different values λ_2 for noisy testing ($\sigma = 0.05$)

4.4 Comparison of all algorithms

Table 8 – 10 shows the results using upscaling factor of 8 and different values σ

	Bicubic	**Eigen trasnformation**	**Neighbour Embedding**	**Sparse Representation**
PSNR (dB)	25.29	25.38	25.76	26.60
SSIM	0.729	0.706	0.763	0.779

Table 8 σ = 0.02

	Bicubic	**Eigen trasnformation**	**Neighbour Embedding**	**Sparse Representation**
PSNR (dB)	23.86	24.94	25.04	26.46
SSIM	0.665	0.707	0.747	0.778

Table 9 σ = 0.05

	Bicubic	**Eigen trasnformation**	**Neighbour Embedding**	**Sparse Representation**
PSNR (dB)	20.00	23.90	22.85	24.78
SSIM	0.500	0.708	0.691	0.750

Table 10 σ = 0.1

	Eigen trasnformation	**Neighbour Embedding**	**Sparse Representation**
PSNR (dB)	26.41	27.26	28.02
SSIM	0.74	0.806	0.838

Table 11 the results using upscaling factor of 4 for noiseless image

5. Discussion

5.1 Eigen transformation

For eigen transformation, the size of the training samples alternates the performance of the super resolution. Increasing the size of the training samples does not necessary increase the performance of the super resolution. From table 1, it is shown that the performance of the super resolution image first increases with the size of the training samples. However, it drops when the number of training samples increases from 175 to 200. This may be due to the noise is added when there are too much eigen faces.

This algorithm also relies on the fact that the images of the training samples should be similar to the input resolution image. Moreover, before applying this algorithm, the facial image must be first aligned according to the eyes position of the image. Otherwise, the generated high-resolution image contains a lot of noise.

The noise performance of the algorithm is satisfactory once the face is aligned to the same position. From table8 to 10, compared to the bicubic interpolation, it is shown that the performance does not drop significantly when random noise is added to the input image. The generated high-resolution image still preserves the global structure of the facial image even the noise is added to the input image.

5.2 Neighbour Embedding

In the experiment, it is found that the k neighbours are the optimal at 2-3 for 5x5 windows and 3-4 for 3x3 windows. The performance is the best for 3x3 windows. This is because larger value of k tends to add more artificial noise in to the training samples. Smaller value of k will overfit the results to that few training samples.

It is shown that this method performs quite well compared to the eigen transformation and the bicubic interpolation. This method is better than bicubic interpolation because more high frequency details could be recovered by the training samples. However, this method is not robust to noise. In table 8 – 10, when the noise is larger, the performance of the neighbour embedding is poor. The PSNR and SSIM drop significantly when the noise added to the input is large. One of the reasons is that the pixels intensity is no longer the same in the original input. The approximation of k neighbours does not hold true because the distance from the input and training samples vary with noise.

Although this method achieves a better PSNR to the eigen transformation when there is little or no noise, the visual effects of the super resolution vary very much. This is because artificial noise is added when the high-resolution image is generated.

5.3 Sparse Representation

The sparse representation is best among all three learning algorithms. In the experiment, the size of the windows is the best at 3x3 with 1 overlapping pixel. This method is the best whenever the image is noisy or not.

There are two parameters could be tuned in this algorithm, λ_1 and λ_2. The physical meaning of the parameters is the penalty of the constraint in the optimization problem. By using a suitable value of λ_1 and λ_2, the PSNR could be improved around 20 – 30db.

For a noiseless image, the performance is the optimal when $\lambda_1 = 0.0001$ and $\lambda_2 = 0.0$. This is meant that the penalty $\|w_i^q - w_{i-1}^q\|$ could increase the performance only when there is noise. This is because this penalty tends to make the image smoother. In table 7, it is shown that the performance is optimal when $\lambda_2 = 0.001$ in our experiments.

This method could achieve better performance than the other twos because of the sparse coefficients. The sparsity selects features automatically and set most of the coefficients to be zero. This eliminates the effect of noise added during down-sampling process. By introducing the second penalty, the effect of noise could be further eliminated, and the generated image is smoother.

6. Future Plan

Work	Time
Modification and measurement of Deep learning approaches	Jan 2019
Implementation of facial image super resolution GUI system	Feb 2019
Discovering the effect of image super resolution to face recognition	Mar 2019
Finalizing the project and report writing	April 2019

7. Conclusion

Image super resolution is an ill-posed problem. There are many ways to estimate the high-resolution image given the low resolution one. Increasing the resolution by interpolation is not robust to noise and tends to make the image unclear. Using examples-based learning algorithms, more details of the input image could be recovered by learning the training samples. In the current process, it is shown that the sparse representation achieves the best performance.

8. Appendices

LR: Low Resolution Input, BC: Bicubic Interpolation, PCA: Eigen transformation,

LLE: Neighbour Embedding, SR: Sparse Representation GT: Ground Truth

upscaling factor = 8 and σ = 0.02

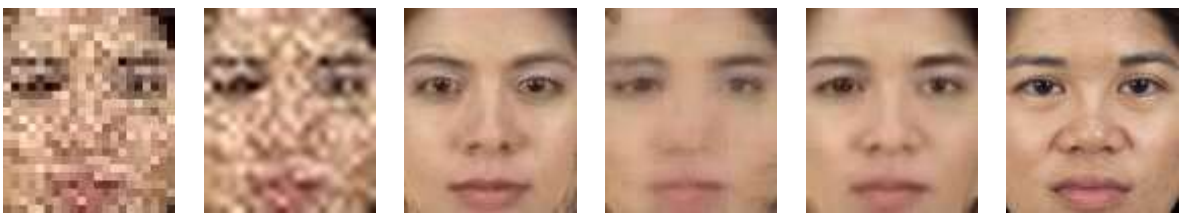
LR **BC** **PCA** **LLE** **SR** **GT**

upscaling factor = 8 and σ = 0.05

LR **BC** **PCA** **LLE** **SR** **GT**

upscaling factor = 8 and σ = 0.1

LR **BC** **PCA** **LLE** **SR** **GT**



upscaling factor = 8 and σ = 0.02

upscaling factor = 8 and σ = 0.05

upscaling factor = 8 and σ = 0.1

9. References

[1] B.J. Boom, G.M. Beumer, L.J. Spreeuwers& R. N. J. Veldhuis, "The Effect of Image Resolution on the Performance of a Face Recognition System," *International Conference on Control, Automation, Robotics and Vision*, 5-8 Dec. 2006.

[2] E. Bilgazyev, B. Efraty, S. K. Shah& I.s A. Kakadiaris, "Improved face recognition using super-resolution," *IEEE International Joint Conference on Biometrics*, 11-13 Oct. 2011.

[3] S. Moitra, "Single-Image Super-Resolution Techniques: A Review," *International Journal for Science and Advance Research in Technology*, Apr. 2017.

[4] C. Dong, C. C. Loy, K. He& X. Tang, "Learning a Deep Convolutional Network for Image Super-Resolution, " *European Conference on Computer Vision*, 6-12 Sep 2014.

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

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

[7] J.Yang, J. Wright, T. Huang& Y. Ma, "Image super-resolution as sparse representation of raw image patches, " *IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[8] J. Yang, J. Wright& T. S. Huang, "Image Super-Resolution Via Sparse Representation, "*IEEE Transactions on Image Processing*, May 2010.

[9] J. Jiang, J.Ma& C. Chen, "Noise Robust Face Image Super-Resolution Through Smooth Sparse Representation, "*IEEE Transactions on Cybernetics*, Nov 2017.

[10] R. Tibshirani "Sparsity and smoothness via the fused lasso, "*Journal of the Royal Statistical Society Series B*, 2015.

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

[12] C. Dong, C. C. Loy, X. Tang, "Accelerating the Super-Resolution Convolutional Neural Network, " *European Conference on Computer Vision*, 2016.

[13] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang& Wenzhe Shi, "Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network, " *Computer Vision and Pattern Recognition,* Aug 2017.