The Hong Kong Polytechnic University

Department of Electronic and Information Engineering

Final Year Project Report

submitted in partial fulfilment of the requirements the Degree of
Bachelor of Engineering (Honours) in Electronic and Information Engineering

Project Title:

Machine Learning for Facial Image Super-resolution

Name:

CHEUNG Tsun Hin

Student ID:

15083269D

Supervised by:

Prof. Kenneth Kin-Man LAM

Abstract

In this project, conventional machine learning and recent deep learning methods for facial image super-resolution are investigated and evaluated. Three state-of-the-art conventional machine learning approaches, including eigentransformation, neighbour embedding and sparse representation, are studied and implemented. Recently, deep learning (DL) methods have become popular for solving both low-level image processing and high-level computer vision problems due to the power and effectiveness of convolutional neural networks (CNNs). The deep learning algorithms considered in our study include the convolutional neural network (CNN) and the generative adversarial network (GAN). The conventional machine learning algorithms are implemented using C++ with the OpenCV library, while the deep learning methods are implemented using Python, with the Pytorch library. All the conventional machine learning methods and the deep learning methods implemented for facial image-resolution are trained using the same training set, and evaluated with the same testing set. Therefore, the performance of the different face super-resolution algorithms can be compared in a fair manner. Furthermore, the algorithms implemented in this project are evaluated on different datasets, with different down-sampling kernels, upscaling factors, and different noise levels. Experiment results show that the deep learning methods achieve the best performance in terms of the peak signal to noise ratio (PSNR) and structural similarity index (SSIM). Visual results based on the different methods are also illustrated.

Acknowledgement

I would like to express my special thanks to my supervisor **Prof. Kenneth LAM** for his guidance and support in completing the project for this year. During the whole project period, he taught me the technical knowledge required for this project and guided me to complete the entire work. He also helped me when I had questions and difficulties in this project.

CHEUNG Tsun Hin

April 2019

Table of Contents

1. Introduction

1.1 Image Super-resolution

Image super-resolution (SR) is to generate a high-resolution image given one or more low-resolution inputs of the same scene [1]. Depending on the number of inputs, it could be classified as single image super-resolution (SISR) or multi-image super-resolution (MISR). Only SISR is considered in this project. 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 [2]. E. Bilgazyev et al. showed that the performance of face recognition of a low-resolution image could be improved by performing image super-resolution [3].

LR Input (16 x 16) SR Output (128 x 128) HR Original (128 x 128)

Figure 1 Comparsion of the original high-resoluton (right) and super-resolved image (middle) given a low-resolution image (left)

Super-resolution is an ill-posed task since the down-sampling operation is non-invertible and the high-frequency information is lost. Thus, one of the assumptions of image-super-resolution is that the high-frequency components are recovered by its low-frequency components. Moreover, if the original high-resolution image is unknown, multiple solutions could be obtained and determining the best result among the solutions is not reasonable.

Face Hallucination is a special case of image super-resolution where the input image is a facial image. The super-resolution algorithms could be categorized in many ways, such as interpolation-based, reconstruction-based and example-based methods. 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 [4]. The example-based method is to generate the high-resolution image by learning the training pairs of low and high-resolution images. In this project, the majority of example-based methods of face hallucination is investigated. The interpolation-based methods will also be considered for comparison.

1.2 Machine Learning

Machine Learning (ML) is the algorithm that simulates the statistical model by programming the computers to use the training data to solve a given testing data [5]. Machine Learning algorithms could be classified as three types including supervised learning, semi-supervised learning and unsupervised learning. The main difference between supervised learning and unsupervised learning is that supervised learning is to learn from the training data with known labels. In contrast, unsupervised learning is to learn the pattern of the input data with unknown label or class. Semi-supervised learning algorithms also make use of the training data during training with a small portion of labelled data but large unlabelled data. In this project, the supervised learning algorithms are considered since the sample pairs of both low-resolution and high-resolution images, where the input is low-resolution and output is high-resolution in this case, will be used during training.

Deep learning (DL) is one of the most popular supervised learning algorithms that grow rapidly in the recent year. The deep neural network (DNN), which is a kind of artificial neural network (ANN), has many layers after the input layers and before the output layers [6]. Neural Network is a highly non-linear mathematical model due to the activation layer. The non-linear property of the activation function is helpful for solving many problems including prediction and classification. Neural Network is built using many connecting units called neurons, which simulate the neurons in the human brain.

Convolutional neural network (CNN) is one of the important deep neural networks that developed in recent years. The main property of CNN is the shift invariant, which is the most important property for computer vision problem [6]. For example, in image-super-resolution, the generated high-resolution image should be invariants to the spectral domain. No matter how the image is shifted in either x or y directions should give the same result. The typical CNN has convolutional, pooling and fully connected layers. However, pooling and fully connected layers may lower down the dimension of the features which is not suitable for the image-super-resolution problem that requires up-scaling. Therefore, CNN for super-resolution mainly focuses on the convolutional layer that extracts features in the images.

1.3 Contributions

Implementation of the machine learning and deep learning algorithms

In this project, the algorithms for facial image super-resolution are implemented on Windows 10 64-bit operating system. The conventional machine learning methods are implemented using C++ with OpenCV library while the deep learning methods are implemented using python with Pytorch library. Different training datasets and testing datasets are functional in all programs. The parameters for each algorithm could be adjusted in the programs.

Modification and retraining the existing models for facial image super resolution

Some algorithms are designed for image super-resolution with general images as the input, a modification is done so it fits the need for facial image super-resolution. The optimal values of the parameters for neighbour embedding and sparse representation are also investigated for facial images as the input. The values would be different from the general images as the input. The deep learning model are re-trained by facial images as the training datasets. The performance is better than the models trained by general images for facial image super-resolution.

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

In most of the original papers, the experiments are done using general images as the input. For the face hallucination ones, the authors only did the experiments on a constrained face dataset. Both unconstrained and constrained facial images are considered in this project. The evaluation and comparison have been done for different methods, with different down-sampling kernels and noise levels. Both training sets are testing sets are the same for fair comparison of different methods. Some experimental results are explained by signal processing theory.

2. Background

2.1 Interpolation-based

Polynomial-based interpolation is an up-sampling method estimating the unknown data of a signal by its known data. It does not require training samples. Bicubic interpolation is one of the cubic interpolations that up-sampling on a two-dimensional space. Bicubic interpolation is often used for upscaling of the images instead of bilinear or nearest-neighbour because more details could be obtained by bicubic interpolation. However, the computation of bicubic interpolation is higher.

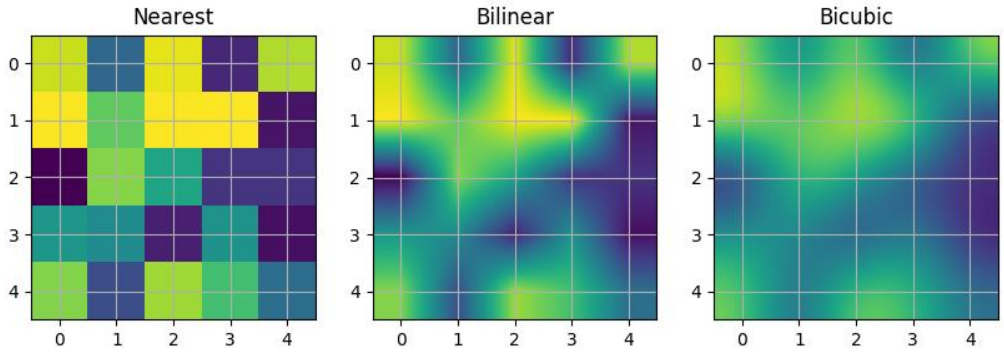

Figure 2 Illustration of different interpolation methods [7]

From figure 2, it is shown that the bicubic may generate a smoother image compared to nearest and bilinear methods.

The idea of bicubic interpolation is to represent the unknown data points by a weighted average of its four pixels (4 x 4) nearby values. It is often used as a baseline for comparison in image super-resolution.

$$p(x,y) = \sum_{i=0}^{3}\sum_{j=0}^{3} a_{ij}x^i y^j \quad (1)$$

2.2 Eigen Transformation

Eigen transformation for face hallucination was proposed by Wang and Tang in 2005 [8]. This method is a direct application of the principal component analysis (PCA). This method relies on the fact that the training and testing facial 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 images could be recovered by the low-frequency component given a training set of facial images. This method tries to recover the high-frequency component of the input image by learning the training samples.

In PCA analysis, the LR images of training samples are first projected on to the subspace that maximises the variances. The basic vectors of this subspace will be uncorrelated. The input LR image is then projected on this subspace and the coefficients that best represents as the linear combination of the LR basic vectors are found. By replacing its LR basic vectors by its HR basic vectors, the image could be super-resolved.

Mathematical Analysis

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

$$[\vec{l}_1, \vec{l}_2 \dots \vec{l}_M] \tag{2}$$

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.

$$\vec{m}_l = \frac{1}{M}\sum_{i=1}^{M} \vec{l}_i \tag{3}$$

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

$$L = [\vec{l}_1 - \vec{m}_l, \vec{l}_2 - \vec{m}_l, \dots \vec{l}_M - \vec{m}_l] \tag{4}$$

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 eigenvector, the eigenvectors of. $L^T L$ could be first calculated.

$$(L^T L)V = V\lambda \tag{5}$$

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

$$(LL^T)LV = LV\lambda \tag{6}$$

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

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

$$\vec{w}_l = (LV\lambda^{-\frac{1}{2}})^T(\vec{x}_l - \vec{m}_l) \tag{7}$$

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

$$\vec{r}_l = \left(LV\lambda^{-\frac{1}{2}}\right)\vec{w}_l + \vec{m}_l \tag{8}$$

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

$$\vec{r}_l = L\vec{c}_l + \vec{m}_l \tag{9}$$

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

$$\vec{r}_l = \sum_{i=1}^{M} c_i\vec{l}_l + \vec{m}_l \tag{10}$$

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

$$\vec{x}_h = \sum_{i=1}^{M} c_i\vec{l}_h + \vec{m}_h \tag{11}$$

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

2.3 Neighbour Embedding

Neighbour embedding for image super-resolution was proposed by Chang, Yeung and Xiong in 2004 [9]. This method is based on one of the manifold learning algorithms called locally linear embedding (LLE). Especially, the method predicts the output by learning the linear relationship of the input with the training data that has a 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 the geometric distance.

Mathematical Analysis

First, denote the image patch of the input image as x_t^q and that of training image as x_i^q. For each image patch of the input image, find the k nearest neighbours x_i^q that is 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

$$\varepsilon^q = \left\| x_t^q - \sum_{i=1}^{k} w_i^q x_i^q \right\|_2^2 \quad \text{s.t. } \sum_{i=1}^{k} w_i^q = 1 \tag{12}$$

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:

$$G_q = (x_t^q \mathbf{1}^T - \boldsymbol{X})^T (x_t^q \mathbf{1}^T - \boldsymbol{X}) \tag{13}$$

The above objective function can be solved by

$$G_q w_q = \mathbf{1} \tag{14}$$

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

$$\sum_{i=1}^{k} w_i^q = 1 \tag{15}$$

The reconstructed LR image r_l is

$$r_l = w_q X \tag{16}$$

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

$$y_h = \sum_{i=1}^{k} y_i^q w_i^q \tag{17}$$

2.4 Sparse Representation

Sparse representation for image super-resolution was proposed by Yang in 2008 [10]. The method represents the input LR image as a sparse linear combination of the raw low-resolution images. Afterwards, the author further proposed that the over-completed sparse dictionary could be prepared instead of the raw images [11]. [12] 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. As mentioned in the original paper, a smooth sparse representation is more effective when there is 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

$$\varepsilon^q = \left\| x_t^q - \sum_{i=1}^{M} w_i^q x_i^q \right\|_2^2 \quad \text{s.t. } \sum_{i=1}^{M} w_i^q < \varepsilon_1 \text{ and } \sum_{i=2}^{M} w_i^q - w_{i-1}^q < \varepsilon_2 \tag{18}$$

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

$$\text{Min } \left\| x_t^q - \sum_{i=1}^{M} w_i^q x_i^q \right\|^2 + \lambda_1 \|w^q\| + \lambda_2 \left\| w_i^q - w_{i-1}^q \right\| \tag{19}$$

Where λ_1 and λ_2 are non-negative parameters.

It is noticed that the objective function and the constraints are both convex. The objective function is smooth, but the constraints are not. This means that the constraints function is not differentiable and thus it could not be solved directly by setting the derivative to zero.

The above is a constrained least square problem called Fused Lasso [13]. There are many existing algorithms that solve the optimization problem. This project uses the existing library and the algorithm used is called Fast Iterative shrinkage-thresholding algorithm (FISTA) [14].

2.5 Deep Learning

Convolutional Neural Network (CNN)

Deep learning is the current state-of-the-art method for image super-resolution. Most of the deep learning models for super-resolution is based on the convolutional neural network (CNN). The deep neural network is based on the back-propagation algorithm that learns the parameters in an end-to-end mapping function by minimizing the error at the output.

In 2014, the first deep learning method solving image super-resolution was invented using Convolutional Neural Network (SRCNN) [15]. It only has three layers, feature extraction, non-linear mapping and reconstruction layers. The authors further showed that these three layers are analogue to sparse coding methods. It also showed the performance of image super-resolution is comparable to the other conventional methods.

The SRCNN approach used the bicubic upscaling as the input of the neural network leads to two drawbacks. First, the computation time is higher since the dimension of the image is larger. Secondly, there may be some artifices in the generated high-resolution image especially when the down-sampling kernel is unknown. The network is not deep, and some studies showed that increasing the number of layers could have better performance of super-resolution images. The SRCNN minimises the mean square loss between the original high-resolution and the generated. However, the image is still blurred and smooth.

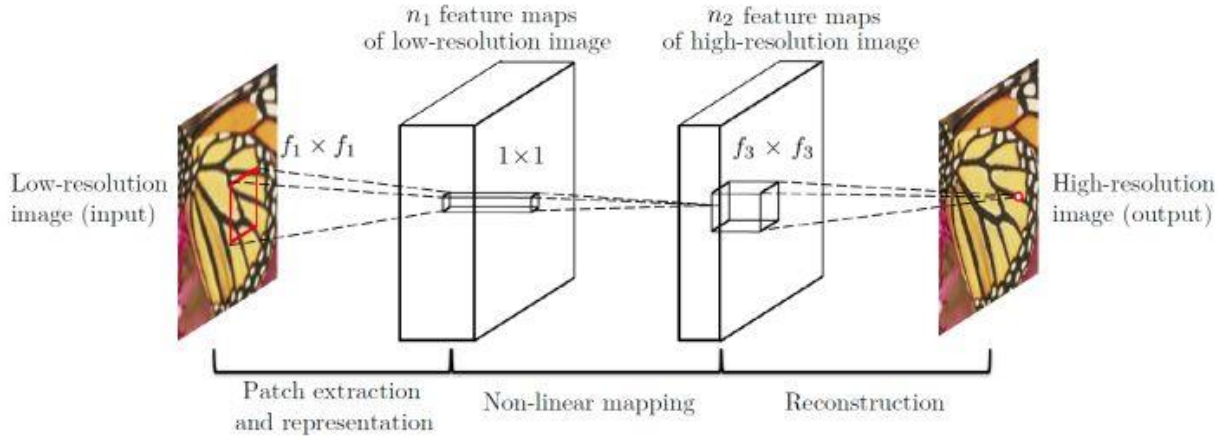

Figure 3 Diagram of SRCNN, the first CNN for image-super-resolution [15]

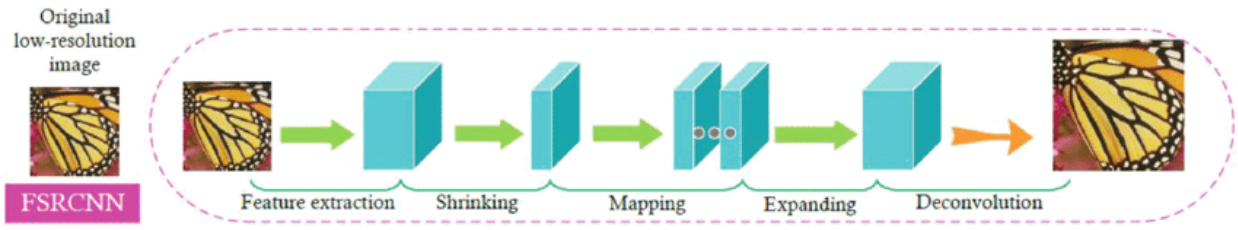

Figure 4 Diagram of FSRCNN with deconvolution layer [16]

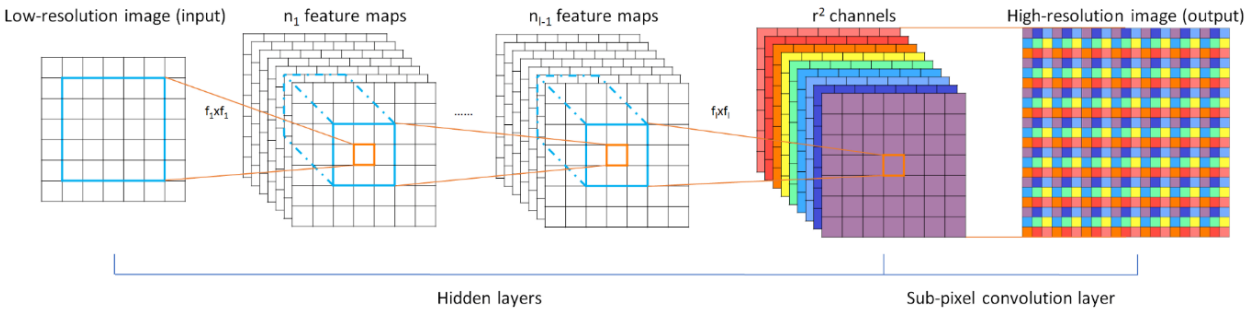

Figure 5 Diagram of ESPCN with sub-pixel convolution layer [17]

Afterwards, different deep learning methods are developed for solving single image super-resolution and modify to tackle these two issues. The FSRCNN [16] and ESPCN [17] are the two neural networks using the original size of the LR image as the input of the neural network. Both networks contain the upscaling layers at the end of the network but with different approaches. FSRCNN uses the deconvolution layer which could be viewed as the inverse operation of convolutional layer. The deconvolution layer acts as the upscaling layer as the final layer of the FSRCNN. In contrast, ESPCN adopts the sub-pixel layer as the upscaling layer.

Some studies showed that increasing the number of layers could improve the performance of super-resolution using neural networks. There are many approaches for increasing the number of layers, such as DRCN [18], VDSR [19], SRDenseNet [20], RDN [21] and SRResNet [22]. DRCN uses the recursive layers by repeating using the parameters in the consecutive layers [18]. VDSR adopts the skip connection layers that could avoid the gradient explosion problem [19]. SRDenseNet [20] and RDN [21] adopts densely connected layers, which was proposed in 2017. In the paper of SRResNet [22], the author added residual blocks and skip connections to increase the number of layers. This could be done because the skip connections are useful for training such a deep neural network. Later, EDSR [23] was proposed by removing the batch-normalization layer in SRResNet and adding more filters in the output layer. The performance of EDSR is better slightly.

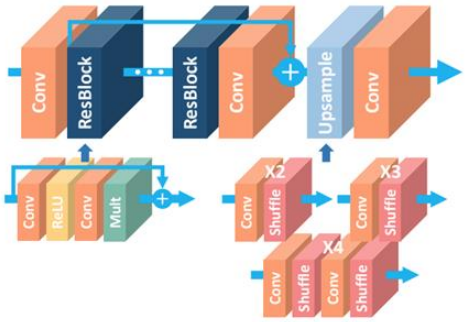

Figure 6 Diagram of Residual Block and skip connections in EDSR [23]

Generative Adversarial Network (GAN)

Apart from the previous works that aim to minimize the mean square error (MSE) between the high-resolution and the generated image, the SRGAN [22] used the perceptual loss in the training process of the deep neural network. Inspired by Li Fei-Fei's work [24], the perceptual loss is the mean square error difference in a high dimension space for the super-resolved and the high-resolution image. The images pass through the vgg19 network and the output of the high dimensional features is compared. This method could generate a photo-realistic image although the PSNR may not be optimised by estimating more high-frequency components in the generated image.

The SRGAN 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 neural high-resolution image. By training the pair of networks simultaneously, the generator network generates a photorealistic super-resolution image.

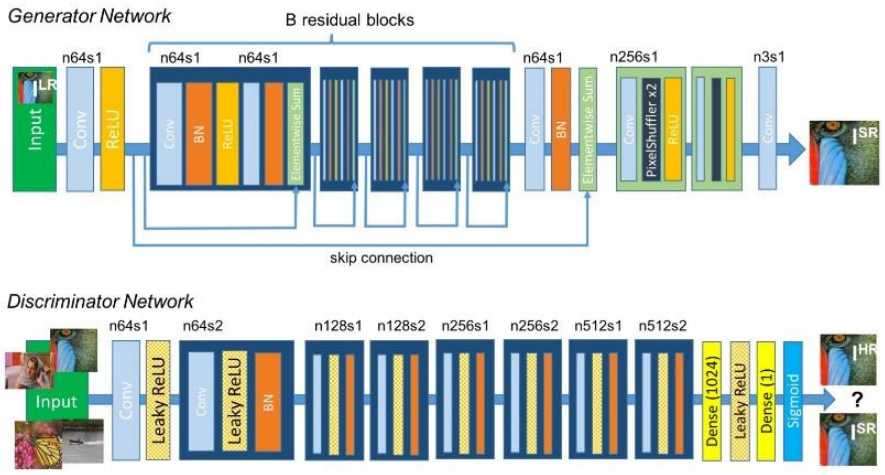

Figure 7 Diagram of the generative adversarial network [22]

In 2018, Y. Blau, T. Michaeli proposed the Perception-Distortion trade-off [25] which is a balance between the perception of human and super-resolved errors. It means that the super-resolution may have good PSNR objective quality but poor subject perception or vice versa. This is because the current algorithms tend to minimize the mean square losses between the generated and the original high-resolution may result in blurring effect [25].

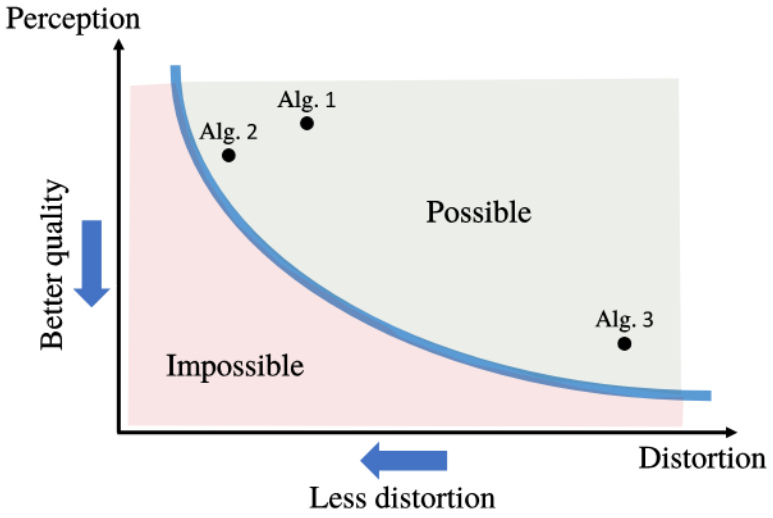

Figure 8 the perception and distortion trade-off [25]

3. Methodology

3.1 Problem Formulation

Down-sampling process

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

$$\vec{I}_l = H\vec{I}_h + \vec{n} \tag{20}$$

where H is the operation matrix involving blurring and down-sampling and $\vec{n}$ is the random distribution added during image acquisition. This assumption holds for all learning algorithms in the project.

The down-sampling kernel is a linear filter that contains both blurring and down-sampling processes. They could be the nearest neighbour, bilinear or bicubic interpolation kernels. The noise added is white noise. Our training samples and testing images are generated by this linear model.

Down-sampling kernels

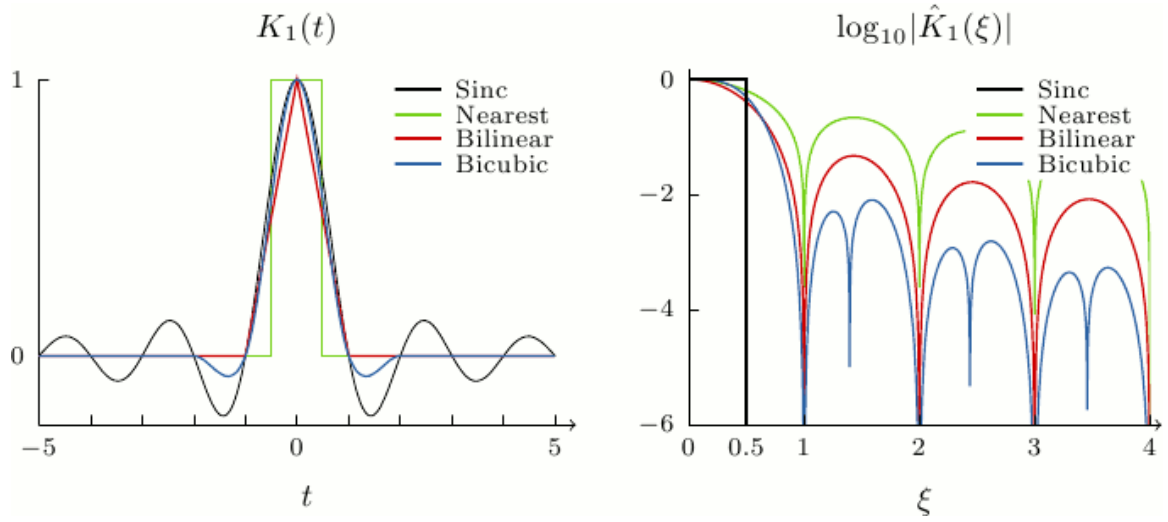

Figure 9 frequency response of the down-sampling kernels [26]

From figure 9, it is shown that the stopband ripple is the largest using nearest neighbour and the smallest using bicubic down-sampling kernels. The larger the stopband ripple may lead to more serious aliasing problem during the down-sampling process.

Sampling Theorem

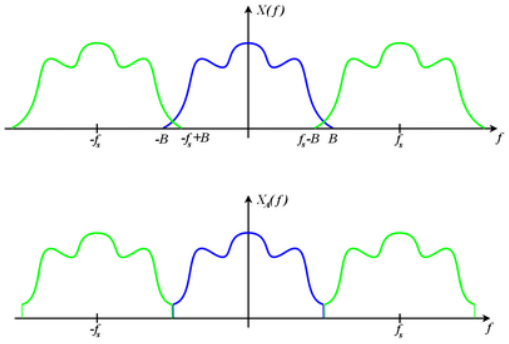

Figure 10 Illustration of sampling theorem [26]

Assume that the signal is band-limited at frequency *B*. If the signal is down-sampled at the sampling frequency *fs*, the frequency spectrum will have multiple samples of the original signal with period 1/ *fs*. If *fs* is smaller than 2 times *B*, aliasing problem occurs, and the frequency samples will be overlapped. The high-frequency components will be added by its low-frequency components as shown in figure 2. Although figure 2 shows a 1-D case, the idea could be generalized in a 2-D case for image analyses. This theorem is important for image super-resolution when the down-sampling kernel is unknown.

3.2 Environment

The eigentransformation, neighbour embedding, and sparse representation algorithms are implemented using c++ with the OpenCV library. With the help of MATLAB R2017b, the performance of the algorithms is measured. For the deep neural network part, the algorithms are implemented in python with the Pytorch library. Sparse Representation toolbox [27] is used for the sparse representation algorithms. All experiments are done in windows 10 64bit system.

3.3 Dataset

In the experiments, the constrained and unconstrained face datasets are examined.

- CHICAGO face database (constrained datasets)

This dataset [28] consists of 597 facial images. This dataset is a constrained dataset and all images are captured under a well-configured environment. 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 the existing library in dlib and opencv according to the 68 key points landmarks of the human face [29]. The images are further cropped into a size of 96×128.

- LFW face database (unconstrained datasets)

This dataset [30] consists of 13233 images which contain 5749 people. 10, 000 of them are selected as the training samples. This dataset is unconstrained face dataset and all images all captured from the internet. 20 of them is for testing. Similarly, the image has been aligned and then cropped into a size of 128x128.

3.4 Experiment Procedures

i. Down-sampling the training samples and testing images using different kernels

ii. Add the noise by varying the value of σ

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

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

v. Compute the average PSNR and SSIM for the testing images

3.5 Algorithms

Eigen transformation

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

$$\vec{x}_l - \vec{m}_l \tag{21}$$

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

$$\vec{c}_l = V\lambda^{-\frac{1}{2}}\vec{w}_l \tag{22}$$

Where

$$\vec{w}_i = \begin{cases} \vec{w}_i, & |\vec{w}_i| < a\sqrt{\lambda_i} \\ sign(\vec{w}_i) * a\sqrt{\lambda_i}, & |\vec{w}_i| \geq a\sqrt{\lambda_i} \end{cases} \tag{23}$$

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

$$\sum_{i=1}^{M} c_i \vec{l}_h \tag{24}$$

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

$$\vec{x}_h = \sum_{i=1}^{M} c_i \vec{l}_h + \vec{m}_h \tag{25}$$

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

$$G_q w_q = \mathbf{1} \tag{26}$$

   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

$$\text{Min } \left\| x_t^q - \sum_{i=1}^{M} w_i^q x_i^q \right\|^2 + \lambda_1 \|w^q\| + \lambda_2 \left\| w_i^q - w_{i-1}^q \right\| \tag{27}$$

   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.

Convolution Neural Network (CNN)

- **Network Architecture**

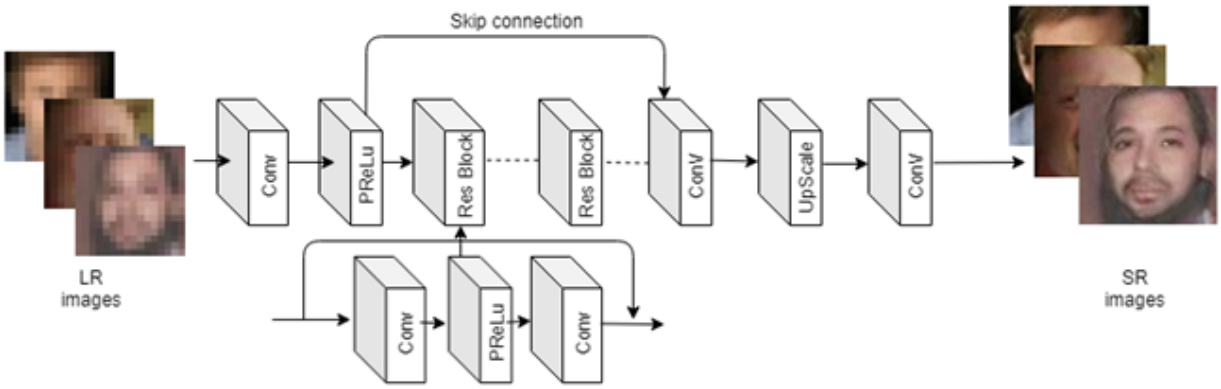

Figure 11 the network architecture of the CNN

Following the most common manner of CNN for image-super-resolution, the generator network consists of three main parts, feature extraction layer as the input layer, non-linear mapping layer as the middle layer, and reconstruction layer for upscaling as the output layer. A single convolutional layer will be used as the input layer. A number of residual blocks and skip connections would be used as the middle layer. Finally, following the ESPCN, sub-pixel suffer layers would be used as the output layer.

- **Feature extraction Part**

The first layer $F_{int}(Y)$ is to extract features by first separating the image into different small patches with overlapping then performing multiplication and addition with the filters. This is exactly doing the convolution the image followed by the parametric rectified linear activation function.

$$F_{int}(Y) = PReLU(W_{in} * Y + B_{in}) \tag{28}$$

- **Non-linear mapping Part**

The second part is the non-linear mapping part. The output from the first layer containing the low-resolution feature is now non-linearly mapped to the high-resolution feature in the second part of the network.

The residual block is the convolution layer followed by the parametric activation function and a convolutional layer as follows.

$$F_{res}(Y) = PReLU(W_{res} * Y + B_{res1}) * Y + B_{res2} \tag{29}$$

The skipping connection $F_{middle}(Y)$ is used to link the output of the first layer and the consecutive residual blocks.

$$F_{middle}(Y) = F_{int}(Y) + F_{res1}(F_{res2}(F_{res\_}(F_{resk}(F_{int}(Y))))) \quad (30)$$

- **Reconstruction Part**

The third part is the reconstruction layers, where the features obtained in the previous are now upscaled by the sub-pixel layers $F_{up}(Y)$. After upscaling to the desired factor, the last output layer $F_{out}(Y)$ is to be convoluted to generate the super-resolved image.

$$F_{up}(Y) = Pixcelshuffer\_2x(Y) \quad (31)$$

$$F_{out}(Y) = W_{out} * F_{up}\left(F_{up}\left(F_{up}(Y)\right)\right) + B_{out} \quad (32)$$

- **Loss function**

The mean square error between the generated SR and the ground-truth HR image is used for the loss function.

$$l_{mse} = \frac{1}{WH}\sum_{x=1}^{W}\sum_{y=1}^{H}(I_{x,y}^{HR} - G(I_{x,y}^{LR}))^2 \quad (33)$$

Generative Adversarial Network (GAN)

Generator Network

The generator network uses the network in the above CNN. The trained CNN will be used as the pre-trained generator in the GAN. Some studies showed the generator network pre-trained using MSE-loss could have better performance in training the GAN [22].

Discriminator Network

Following the same discriminator network used in SRGAN, the network consists of several convolutional layers followed by batch normalization and non-linear activation function PRLU. After passing the convolutional layers, a fully connected layer followed by non-linear activation function sigmoid. This network is trained to identify the images are generated by the generator or a ground truth original image.

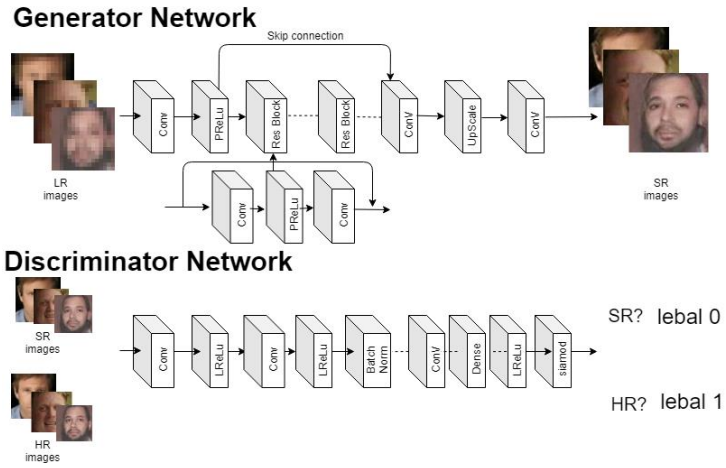

Figure 12 the network architecture of the GAN

Loss functions

The total network loss is a linear combination of content loss and adversarial loss.

$$l_{total} = l_{content} + 0.006\, l_{adversairal} \tag{34}$$

- **Content loss**

There are two content losses considered in this project. They are MSE-loss and VGG-loss.

MSE-loss is the mean square error between the generated and the original high-resolution image.

$$l_{mse} = \frac{1}{WH}\sum_{x=1}^{W}\sum_{y=1}^{H}(I_{x,y}^{HR} - G(I_{x,y}^{LR}))^2 \tag{35}$$

VGG-loss (also called) the feature loss is the mean square error between the high feature space between the generated and the original high-resolution image, where the high feature space projected is the middle layer in the VGG-19 network.

$$l_{vgg} = \frac{1}{WH}\sum_{x=1}^{W}\sum_{y=1}^{H}(V(I_{x,y}^{HR}) - V\left(G(I_{x,y}^{LR})\right))^2 \tag{36}$$

- **Adversarial loss**

The adversarial loss is the loss from the discriminator network. This is to minimize negative log-likelihood that the generator could generate an image that has label 1 in the discriminator. In other words, the generator network tries to fake the discriminator network that the image generated is real.

$$I_{adversairal} = \sum_{n=1}^{N} -\log D(G(I^{LR})) \tag{37}$$

3.6 Training settings

Eigen transformation

a – scalar for bounding the small eigenvectors with smaller eigenvalues

Neighbour embedding

k – the number of the nearest neighbour

s – the size of the patch = 4

l – the size of the overlapping region = 2

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 – the size of the patch = 4

l – the size of the overlapping region = 2

Deep Neural Network

All the deep learning model are trained with learning rate of 0.0001, batch size of 2 and 200 epochs.

3.7 Measurements

Peak signal to noise ratio (PSNR)

$$PSNR = 10log_{10}\left(\frac{MAX^2}{MSE}\right) \tag{38}$$

Where

$$\mathrm{MSE} = \frac{1}{\mathrm{mn}}\sum_{\mathrm{i}=0}^{\mathrm{m}-1}\sum_{\mathrm{j}=0}^{\mathrm{n}-1}[\mathrm{I}(\mathrm{i},\mathrm{j}) - \mathrm{k}(\mathrm{i},\mathrm{j})]^2 \tag{39}$$

Peak signal-to-noise ratio (PSNR) is the most common measurement for super-resolution. It is defined as the ratio between the maximum possible power of the signal and the power of the noise that is corrupted in the generated high-resolution image. The noise is calculated as the mean square error between the original high-resolution image and the generated super-resolved image. In this project, we evaluate the performance of the obtained images by measurement the luminance value in the YCbCr space of the image. The boundary of the image will be ignored so only ten pixels in the boundary will not be calculated for evaluating the performance.

Structural Similarity index (SSIM)

$$\mathrm{SSIM}\ (\mathrm{x},\mathrm{y}) = \frac{2(m_x m_y + C_1)(2\sigma_{xy} + C_2)}{\left({m_x}^2 + {m_y}^2 + C_1\right)\left({\sigma_x}^2 + {\sigma_y}^2 + C_1\right)} \tag{40}$$

Where, where m_x, m_y, σ_x, σ_y, and σ_{xy} are the local means, standard deviations, and cross-covariance for images x, y. C_1 and C_2 are to make the division stabilized.

Structural Similarity index (SSIM) is another method for predicting the quality of the image. It was originally used for evaluating the performance in digital TV broadcasting. This will be used as the second measurement of the performance for the resolved image in this project. Similar to PSNR, only the luminance values in YCbCr space of the image will be used and the boundary will be ignored for evaluating the performance of the super-resolved images.

3.8 Graphical User Interface (GUI)

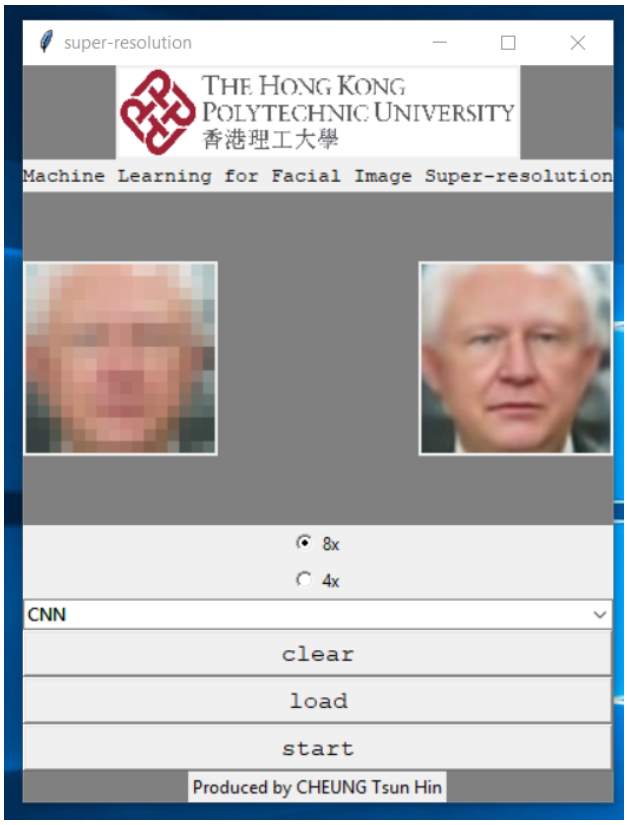

Figure 13 GUI for facial image super-resolution

A GUI has been developed for practical usage of facial image-resolution. It is implemented using python. It could be run on windows operating system both 32bit and 64bit computer installed with python. The user can load their own image and resolving the image. The user can choose the specific upscaling factor 8x or 4x. Moreover, the user can choose the algorithms to be used for super-resolution. For tradition machine learning algorithms, the user can replace the training pairs of LR and HR images. For the deep learning method, a trained model could be changed in the folder. After loading and starting the program, the high-resolution image will be generated.

4. Result

Eigen transformation (PCA)

- Dataset: CHICAGO face database
- Upscaling factor: 8
- Down-sampling kernel: bicubic

α	**0.05**	**0.1**	**0.2**	**0.5**	**1**	**INF**
PSNR (dB)	24.08	25.33	24.60	23.38	23.38	23.38
SSIM	0.766	0.744	0.687	0.625	0.625	0.625

Table 1 results of PCA using different values of α using an upscaling factor of 8

- Dataset: CHICAGO face database
- Upscaling factor: 4
- Down-sampling kernel: bicubic

α	**0.05**	**0.1**	**0.2**	**0.5**	**1**	**INF**
PSNR (dB)	25.94	27.96	27.95	27.85	27.84	27.84
SSIM	0.812	0.808	0.789	0.783	0.783	0.783

Table 2 results of PCA using different values of α using an upscaling factor of 4

Neighbour Embedding (LLE)

- Dataset: CHICAGO face database
- Upscaling factor: 8
- Down-sampling kernel: bicubic

k	**2**	**3**	**4**	**5**	**6**	**7**
PSNR (dB)	25.90	26.24	26.35	26.50	26.60	26.36
SSIM	0.775	0.784	0.784	0.785	0.785	0.776

Table 3 results of LLE using different values of k with constrained dataset

- Dataset: LFW face database
- Upscaling factor: 8
- Down-sampling kernel: bicubic

k	2	3	4	5
PSNR (dB)	24.27	24.77	24.61	24.55
SSIM	0.669	0.691	0.681	0.675

Table 4 results of LLE using different values of k with unconstrained dataset

Sparse Representation (SR)

- Dataset: CHICAGO face database
- Upscaling factor: 8
- Down-sampling kernel: bicubic

λ_1	**0.01**	**0.001**	**0.0001**	**0.00001**
PSNR (dB)	25.40	26.21	26.46	26.46
SSIM	0.782	0.783	0.794	0.794

Table 5 results of SR using values of $\boldsymbol{\lambda_1}$ and $\boldsymbol{\lambda_2 = 0}$ for noiseless images

λ_2	**0.01**	**0.001**	**0.0001**	**0.00**
PSNR (dB)	25.49	26.28	26.42	26.46
SSIM	0.762	0.787	0.793	0.794

Table 6 results of SR $\boldsymbol{\lambda_1 = 0.0001}$ and different values $\boldsymbol{\lambda_2}$ for noiseless images

λ_2	**0.005**	**0.001**	**0.0001**	**0.00**
PSNR (dB)	24.72	24.75	24.67	24.64
SSIM	0.685	0.691	0.681	0.680

Table 7 results of SR $\boldsymbol{\lambda_1 = 0.0001}$ and different values $\boldsymbol{\lambda_2}$ for noisy testing (σ = 0.05)

Convolutional Neural Network (CNN)

- Dataset: LFW face datasets
- Upscaling factor: 8

Down-sampling kernels for training samples

Down-sampling kernels for 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

Table 8 results of CNN of PSNR (dB) / SSIM with different down-sampling kernels in training and testing images.

Overall Performance

Bicubic down-sampling, Upscaling factor 4

- Dataset: CHICAGO face database

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

Table 9 results using different methods using bicubic down-sampling with constrained dataset

- Dataset: LFW face database

	BC	PCA	LLE	SR	CNN	GAN
PSNR (dB)	28.85	N/A	28.10	29.08	32.69	30.75
SSIM	0.843	N/A	0.794	0.823	0.911	0.880

Table 10 results using different methods using bicubic down-sampling with unconstrained dataset

Unknown down-sampling, Upscaling factor 4

- Dataset: CHICAGO face database

	BC	PCA	LLE	SR	CNN	GAN
PSNR (dB)	27.74	26.67	28.16	28.32	31.20	30.10
SSIM	0.842	0.788	0.840	0.849	0.904	0.887

Table 11 results using different methods using unknown down-sampling with constrained dataset

- Dataset: LFW face database

	BC	PCA	LLE	SR	CNN	GAN
PSNR (dB)	27.46	N/A	27.01	27.95	31.89	30.01
SSIM	0.804	N/A	0.765	0.798	0.899	0.865

Table 12 results using different methods using mix down-sampling with unconstrained dataset

Overall Performance

Bicubic down-sampling, Upscaling factor 8

- Dataset: CHICAGO face database

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

Table 13 results using different methods using bicubic down-sampling with constrained dataset

- Dataset: LFW face database

	BC	PCA	LLE	SR	CNN	GAN
PSNR (dB)	24.75	N/A	24.77	25.22	28.32	27.06
SSIM	0.689	N/A	0.691	0.711	0.820	0.773

Table 14 results using different methods using bicubic down-sampling with unconstrained dataset

Unknown down-sampling, Upscaling factor 8

- Dataset: CHICAGO face database

	BC	PCA	LLE	SR	CNN	GAN
PSNR (dB)	23.94	24.31	25.43	25.69	27.48	27.01
SSIM	0.694	0.697	0.766	0.785	0.822	0.797

Table 15 results using different methods using unknown down-sampling with constrained dataset

- Dataset: LFW face database

	BC	PCA	LLE	SR	CNN	GAN
PSNR (dB)	24.22	N/A	23.27	24.37	28.16	27.55
SSIM	0.676	N/A	0.624	0.686	0.819	0.788

Table 16 results using different methods using unknown down-sampling with unconstrained dataset

Bicubic down-sampling, Upscaling factor 8 with noise

- Dataset: CHICAGO face database, sigma = 0.01

	BC	PCA	LLE	SR	CNN	GAN
PSNR (dB)	24.51	24.59	25.77	26.17	27.94	27.30
SSIM	0.710	0.709	0.772	0.790	0.836	0.794

Table 17 results using different methods using noisy images (sigma = 0.01)

- Dataset: CHICAGO face database, sigma = 0.02

	BC	PCA	LLE	SR	CNN	GAN
PSNR (dB)	24.39	24.46	25.74	26.05	27.53	26.94
SSIM	0.703	0.706	0.774	0.790	0.822	0.784

Table 18 results using different methods using noisy images (sigma = 0.02)

- Dataset: CHICAGO face database, sigma = 0.05

	BC	PCA	LLE	SR	CNN	GAN
PSNR (dB)	23.65	24.02	25.18	25.77	25.83	25.56
SSIM	0.665	0.706	0.754	0.785	0.771	0.740

Table 19 results using different methods using noisy images (sigma = 0.05)

5. Discussion

5.1 Eigentransformation

- **Effect of parameter *a* for eigentransformation**

Figure 14 results using different values of α in eigentransformation

The parameter *a* used for bounding the eigenvectors affects the performance of the face hallucination. It is shown that suitable values of *a* could increase the peak signal to noise ratio (PSNR). This is because the value of *a* is to constraint the eigenvectors with small eigenvalues. The larger the eigenvalue means that the eigenvectors contain more correlation in the training sets. This means the eigenvectors with larger eigenvalues are face-like images. For those smaller eigenvalues, their corresponding eigenvectors are non-face elements which may be noise.

In figure 14, it is shown that the value of *a* equals to 0.05 – 0.1 has the best results in terms of PSNR and image quality. A smaller value of *a* may result in less noise contained in the image. However, the image may no longer look like the original input image, and it looks like a more general face image. This is because only small and major eigenfaces are considered. If a larger value of *a* is used, more eigenfaces are considered but some noise may appear in the images as shown in figure 14. If the value of *a* is infinity, it means that there is no boundary for the eigenvectors, the performance will be poor because there is a lot of noise inside the generated image.

Characteristics of Eigentransformation

Eigentransformation is a global face hallucination algorithm. This method could generate a face-like image. Although the image generated has more high-frequency details, the PSNR is the worst among all the machine learning algorithms. This algorithm tends to maintain a good global structure of the facial image. However, the local features are lost. There are two limitations to the eigentransformation method. First, it requires that the input LR image has similar faces inside the training samples, otherwise the results will be poor. Secondly, both training samples and the testing images should be well captured under a similar environment and conditions and the face images must be aligned.

5.2 Neighbour Embedding

- **Effect of the number of nearest neighbours k for Neighbour Embedding**

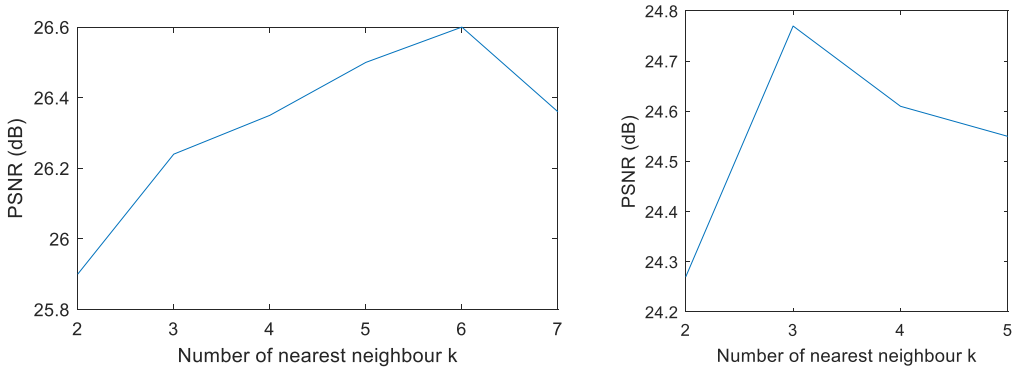

Figure 15 Results of using different values of k with the constrained dataset (left) and unconstrained dataset (right) using neighbour embedding

In the experiment, it is found that the k neighbours are optimal at 6 for constrained datasets and 3 for unconstrained datasets. This is because a larger value of k tends to add more artificial noise into the training samples. A smaller value of k will overfit the results to that few training samples. Moreover, for the constrained datasets, more high-frequency details could be reflected from the training samples of the image because the environment of capturing the image is the same. Therefore, the optimal of k is high at 6, which is a double for the unconstrained datasets. In contrast, for the unconstrained datasets, the training samples would be quite different from the test image.

Figure 16 results of a constrained dataset using neighbour embedding

Figure 17 results of an unconstrained dataset using neighbour embedding

Characteristics of Neighbour Embedding

Neighbour Embedding has a better performance in terms of PSNR than the eigentransformation but poorer performance than the sparse representation and deep learning method. This method preserves the local features of the facial images as compared to the eigentransformation one. However, there may be distortion generated especially in the boundaries between the overlapping regions. This is because the image is proceeded in a window by window basis, it could not preserve the continuity between the boundaries of the overlapping windows.

The computation time of this algorithm is high since the optimization problem is solved for each patch of the image when there is an input image. Moreover, the computation of the feature extraction is high and the distances between the input image patch and patches in the training samples is also time-consuming. This method needs to find the best k nearest patches which are a time-consuming task.

5.3 Sparse Representation

- **Effect of lambda 2 for Sparse Representation**

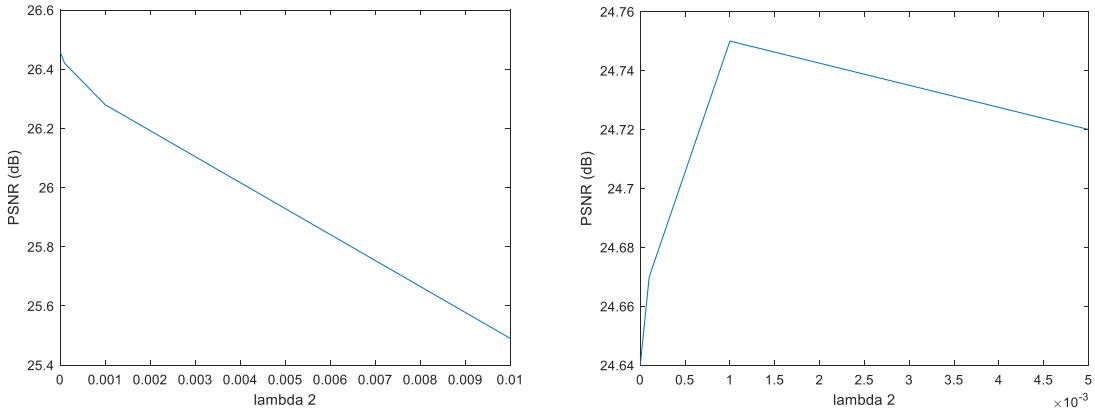

Figure 18 results of using different lambda 2 using the sparse representation for noiseless images (left) and noisy images (right)

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 0.2 – 0.3db.

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

Figure 19 results of a constrained dataset using the sparse representation

Figure 20 results of an unconstrained dataset using the sparse representation

Characteristics of Sparse Representation

The sparse representation is an enhanced version of neighbour embedding algorithms. The sparsity selects features automatically by enforcing most of the coefficients to be zero. This eliminates the effect of noise added during the down-sampling process. The results generated have a high PSNR than the neighbour embedding and eigentransformation algorithms. However, the image generated tends to be smooth and blurring because it minimizes the square error and some high-frequency components would be lost.

For a noisy image, the second penalty could avoid the over-fitting of the training samples. The image has a high PSNR, but the image becomes blurring and smooth. That means the high-frequency components could not be estimated.

Similar to the neighbour embedding algorithm, an optimized is solved when there is an input image, which may require a high computation time. Also, the image is cropped into windows by windows basis which is a time-consuming task. This method also preserves the local features of the input LR image, but the distortion of the image is not high compared to the neighbour embedding algorithm because of the sparsity of the coefficients. Therefore, sparse representation achieves the best result among the three traditional machine learning algorithms, but the performance is still worse than deep learning method.

5.4 deep learning

The convolutional neural network achieves the best result in terms of both PSNR and SSIM. For a constrained dataset, CNN method has the performance PSNR of 1 dB larger than the traditional machine learning algorithms. If a constrained dataset is used, the performance is 3 dB larger than the traditional machine learning algorithms. This shows that the deep learning method could work in a non-constrained environment due to the time-invariant property of the deep neural network.

Although GAN has a smaller PSNR and SSIM than CNN, it still outperforms the traditional machine learning for both constrained and unconstrained datasets. As shown in the below figure, CNN tends to produce a smooth image while GAN could produce more high-frequency components over the textures of the image which makes the image more photo-realistic.

Figure 21 results of a constrained dataset using CNN and GAN

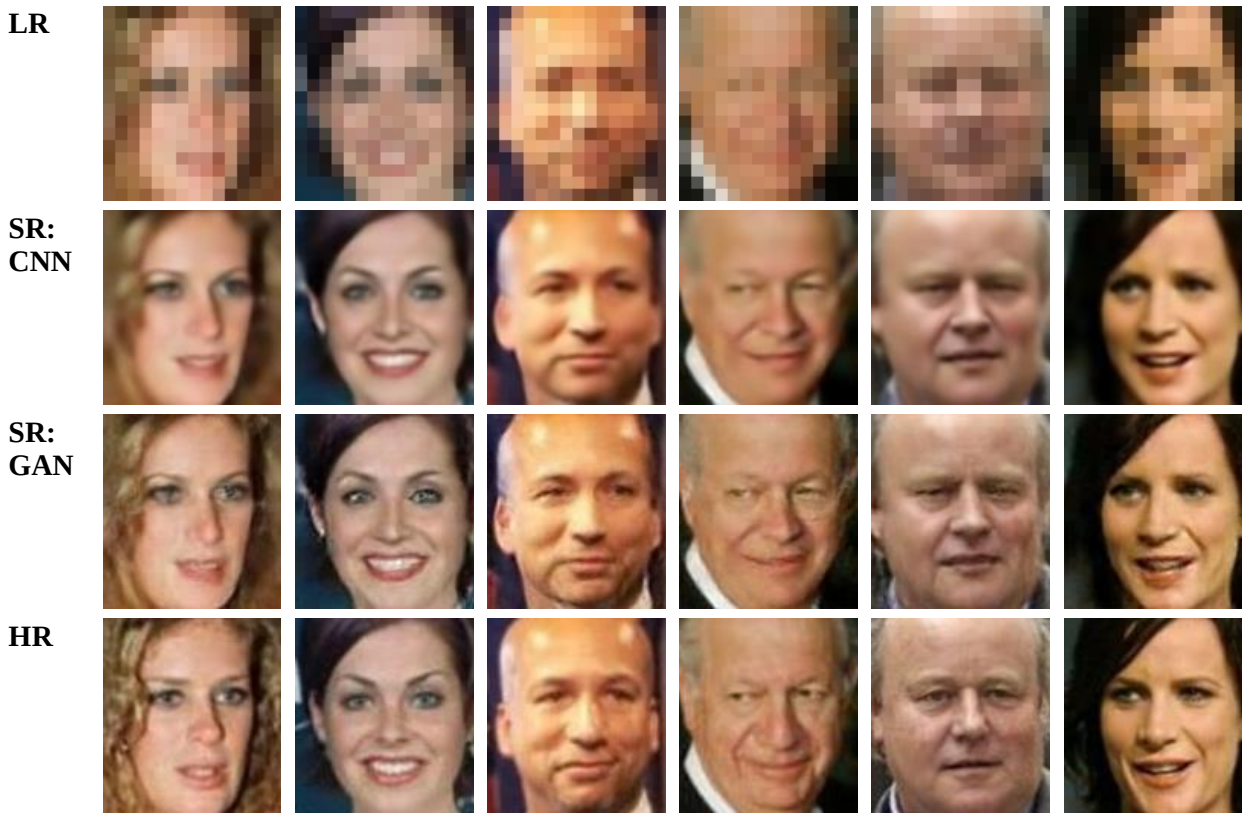

Figure 22 results of an unconstrained dataset using CNN and GAN

The computation time is the fastest for solving a new image. This is because the optimization is done during the training process. Having obtained the parameters for the deep neural network, a new input image is passed the feed-forward network and no optimization process is done. This could lower the computation time for solving a new image.

The number of training samples is large for training a deep neural network. In the experiments, if 200 images are used for training samples, the results will be poor. This may due to the over-fitting of the training samples. In practice, a 10, 000 training images are required to train a deep neural network. It takes almost 24 hours for a GPU to run 200 epochs using a batch size of 2 to obtain satisfactory performance.

Dependency on the training and testing degradation process

The neural network is dependent and sensitive to the down-sampling kernels applied in the training samples and the testing images. In the experiment, if a bicubic down-sampling kernel is used for training samples, the result drops 6 dB for nearest neighbour down-sampling kernel as the test images. However, if the nearest neighbour kernel is used for training samples, the result only drops 1 dB for a bicubic down-sampling kernel as the testing images.

5.5 Effect of down-sampling kernels

Experiments show that the different down-sampling kernels would have different performances of super-resolved images. LR image using bicubic down-sampling kernels could obtain the best super-resolved image. If the nearest neighbour is used as the down-sampling kernel, the result will be the worst, and its PSNR drops 3 dB compared to the bicubic down-sampling kernels one. This means that the bicubic down-sampling kernel preserves most of the information in the LR image.

This effect could be explained by sampling theorem. The frequency response of the bicubic down-sampling has a smaller stopband ripple than that of nearest neighbour. When down-sampling is done, few high-frequency components are overlapped and distorted in the frequency spectrum using bicubic down-sampling. Since the effect of bicubic interpolation could be viewed as a low pass filter, the aliasing becomes less serious. Therefore, more information is preserved using bicubic down-sampling. During the super-resolution stage, more correlation could be obtained in the training image, the performance of the super-resolution using bicubic down-sampling is better.

Figure 23 Results of using different down-sampling kernels using CNN

5.6 Noise Performance

	BC	PCA	LLE	SR	CNN	GAN
PSNR (dB) (sigma = 0.01)	24.51	24.59	25.77	26.17	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	0.40	2.11	1.74

Table 20 results of noise performance using constrained dataset

Sparse representation has the best noise performance. This is because the sparse coefficient plays an important role to avoid over-fitting to the training samples. It also selects features automatically by enforcing most of the coefficients to be zero. When the standard deviation of Gaussian noise increases from 0.01 to 0.05, there are only 0.4 dB drops in the PSNR. The second-best noise performance is eigentransformation. There will be a 0.57dB drops in the PSNR.

The deep learning methods are not robust to strong noise. The PSNR drops 2.11dB when the standard deviation of Gaussian noise increases from 0.01 to 0.05 when CNN is used. It drops is at least twice as the bicubic interpolation does Although the dB drop is significant, CNN still outperforms the other methods.

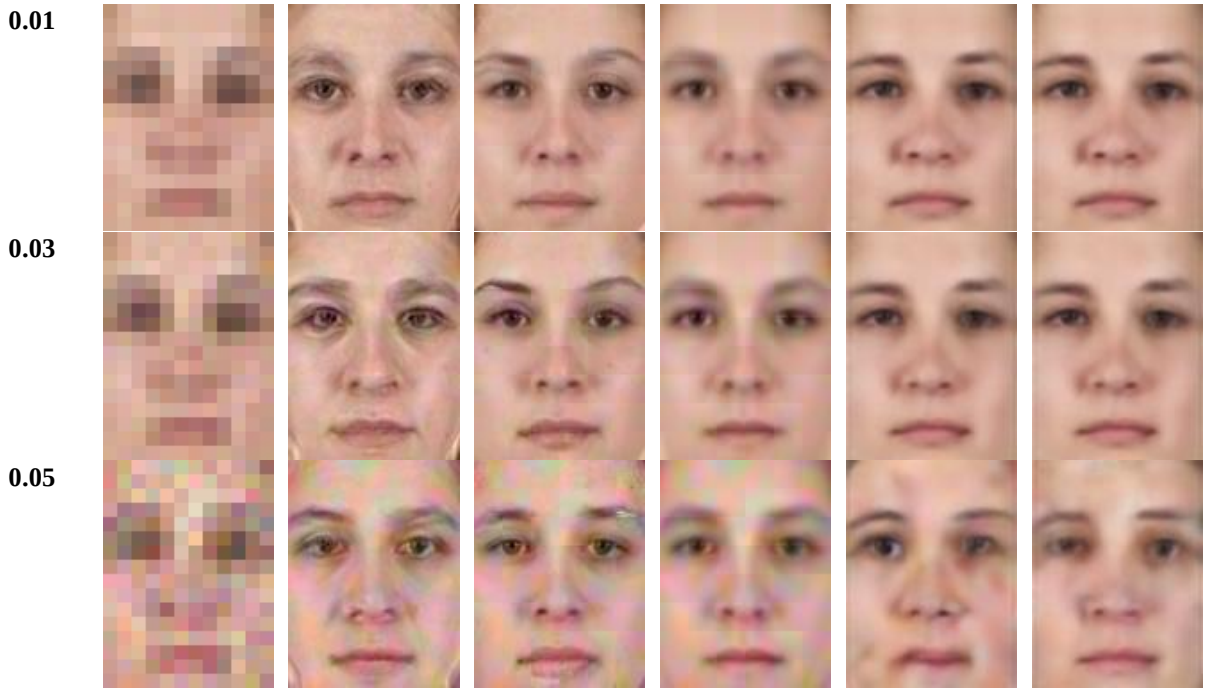

Figure 24 results of different methods with different noise levels (different values of sigma)

5.7 Constrained and unconstrained datasets

The constrained and unconstrained datasets also alter the performance of super-resolution. Bicubic interpolation has similar results for both constrained and unconstrained datasets. Eigentransformation does not work on unconstrained datasets because the image must be captured on a similar environment or settings, otherwise, the results will be poor. Both neighbours embedding and sparse representation methods perform better on a constrained dataset. These two algorithms have 1 – 2dB decrease if unconstrained datasets are used.

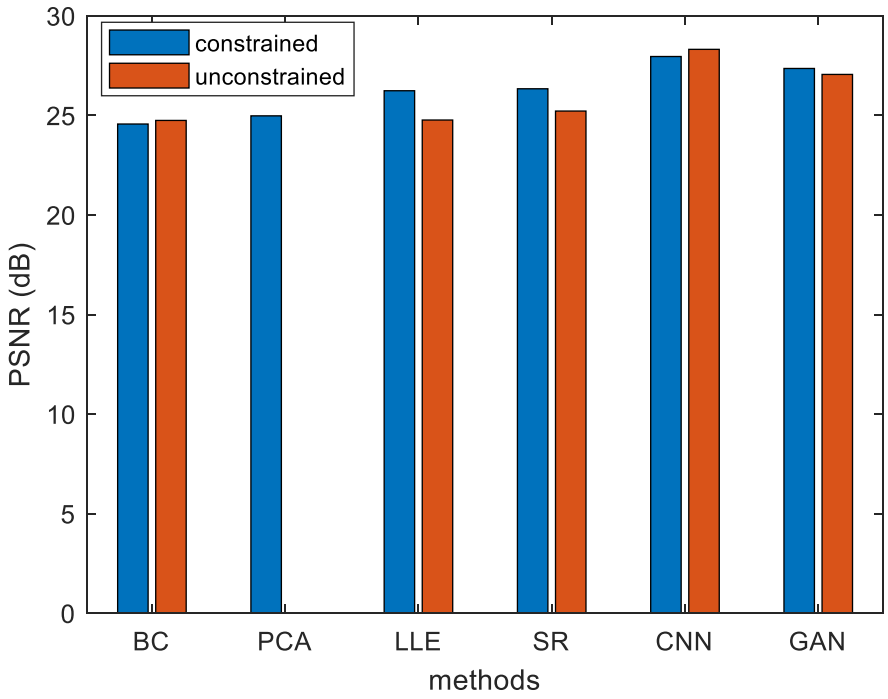

Figure 25 Results using constrained and unconstrained datasets

CNN method performs better on an unconstrained dataset. It performs 0.4 dB better in an unconstrained dataset. This is because there are only 200 training samples for the constrained dataset. The deep neural network could not learn many features in the constrained dataset. Since there are 10, 000 training samples in the unconstrained dataset, more features could be learnt. Thus, the performance is better for the unconstrained dataset using deep neural networks.

5.8 Upscaling factor

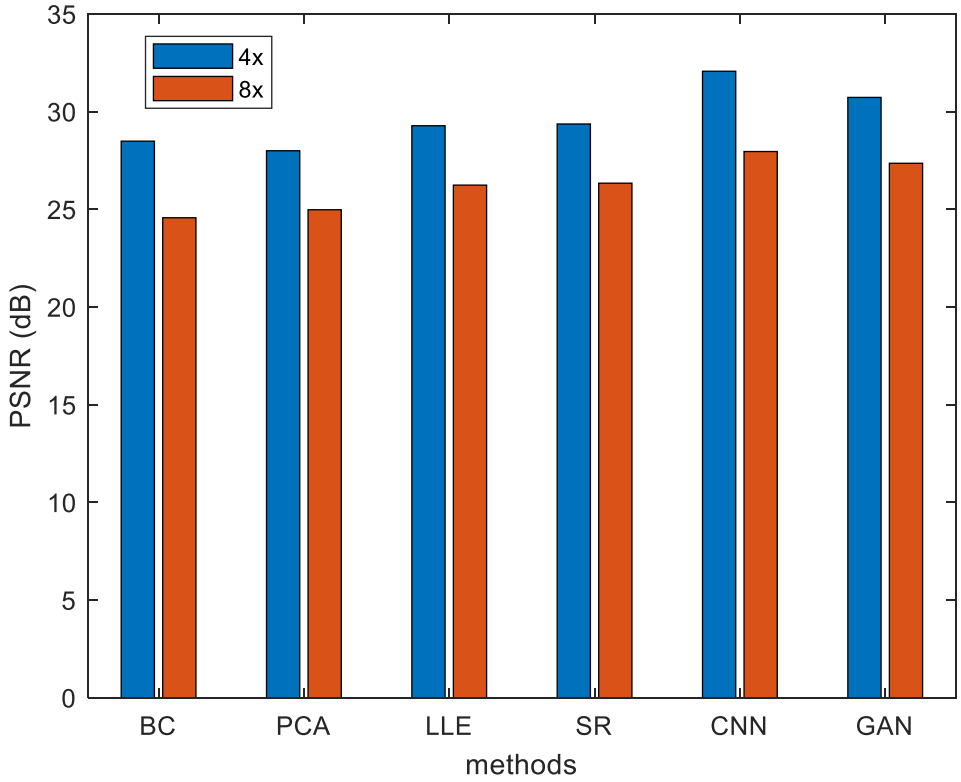

Figure 26 Results using different upscaling factors

The performance of the upscaling factor of 4 is better than that of 8. This is obvious because there are more points needed estimation using larger upscaling factor. Considering the same output size of 128x128. If the upscaling factor is 4, the input size is 32x32. If the upscaling factor is 8, the input size is 16x16. The information contained in a 32x32 image is 4 times more than that of a 16x16 image. Therefore, the larger upscaling factor, the worse performance will have. It is shown that the results will normally have 3 dB drops from upscaling factor of 4 to upscaling factor 8.

6. Future Work

Noise robust deep learning for facial image super-resolution

From the experiments, it is shown that the CNN method drops significantly when there is input is very noisy for a constrained dataset. Facial image super-resolution is different from the general image because face image shares a similar structure such as eyes, mouth, etc. Different neural networks architecture and more training samples may help in improving the noise performance of the deep neural network for facial image-super-resolution.

Moreover, the current approach is to learn an end-to-end mapping function from a noisy LR image to a noiseless HR image. It means the image denoising and super-resolution is done simultaneously. These two operations could be separated into two operations. The results might be different and need more investigation.

Measurement on the perceptual quality or face recognition rate for facial image super-resolution

From the results, it is shown that GAN may produce a more photo-realistic facial image than CNN does, but GAN has a poor PSNR than CNN. That means PSNR and SSIM are not enough for measuring the visual quality of the image generated by GAN.

Moreover, although the image is super-resolved, the face identity may not be preserved, and the facial image generated may become another person. A more objective way such as face recognition rate measurements could be done on the results super-resolved image.

Investigation on degradation process of down-sampling kernels

In practical, the LR image may not be generated by blurring and down-sampling operations. The result becomes poor if the training samples and testing samples are not under the same degradation process. A real LR and HR images training pairs may help to improve the super-resolution in a real situation, but it may require a physical capturing process to generate a more realistic training sample.

7. Conclusion

In this project, the majority of the machine learning algorithms for facial image super-resolution are implemented and their performances are measured. The deep learning methods outperform all conventional machine learning methods in terms of PSNR and SSIM. The deep learning is also fast in a real case because no optimization problem is solved when there is a new input. The conventional machine learning such as sparse representation enjoys the noise-robust characteristics for face hallucination.

8. References

[1] S. C. Park, M. K. Park, and M. G. Kang, "Super-resolution image reconstruction: a technical overview," *IEEE signal processing magazine*, 2003.

[2] 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.

[3] 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.

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

[5] M. Bishop, "Pattern Recognition and Machine Learning," *Springer*, 2006.

[6] I. Goodfellow, Y. Bengio, A. Courville. "Deep Learning," *MIT Press*, 2016.

[7] R. Keys "Cubic convolution interpolation for digital image processing," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 1981.

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

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

[10] 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.

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

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

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

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

[15] 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.

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

[17] W. Shi, J. Caballero, F. Huszar, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, & Z. Wang, "Real-Time Single Image and Video Super-Resolution Using an Efficient Sub-Pixel Convolutional Neural Network, " *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[18] J. Kim, J. K. Lee, K. M. Lee, "Deeply-Recursive Convolutional Network for Image Super-Resolution," *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[19] J. Kim, J. K. Lee, K. M. Lee, "Accurate Image Super-Resolution Using Very Deep Convolutional Networks, " *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[20] T. Tong, G. Li, X. Liu, Q. Gao, " Image Super-Resolution Using Dense Skip Connections," *International Conference on Computer Vision*, 2017.

[21] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, Y. Fu, " Residual Dense Network for Image Super-Resolution, " *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[22] 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, " " *IEEE Conference on Computer Vision and Pattern Recognition*, Aug 2017.

[23] 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.

[24] J. Johnson, A. Alahi, L. Fei-Fei, "Perceptual Losses for Real-Time Style Transfer and Super-Resolution, " *European Conference on Computer Vision*, 2016.

[25] Y. Blau, T. Michaeli, "The Perception-Distortion Tradeoff, " *IEEE Conference on Computer Vision and Pattern Recognition Workshop*, 2018.

[26] P. Getreuer, " Linear Methods for Image Interpolation, " *Image Processing On Line*, 2011

[27] J. Mairal, F. Bach, J. Ponce and G. Sapiro, "Online Learning for Matrix Factorization and Sparse Coding, " *Journal of Machine Learning Research*, volume 11, pages 19-60. 2010.

[28] S. Ma, J. Correll, B. Wittenbrink, " The Chicago face database: A free stimulus set of faces and norming data, " *Behavior Research Methods*, Jan. 2015.

[29] C. Sagonas, E. Antonakos, G, Tzimiropoulos, S. Zafeiriou, M. Pantic. "300 faces In-the-wild challenge: Database and results, " *Image and Vision Computing (IMAVIS), Special Issue on Facial Landmark Localisation "In-The-Wild"*. 2016.

[30] B. Huang, M. Ramesh, T. Berg, E. Learned-Miller, " Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments," *University of Massachusetts, Amherst, Technical Report*, Oct. 2007