Single Image Super-Resolution for Text-based Image : Benchmark and GUI

Jongyoon Kim

April 2021

Final year project thesis submitted
in support of the degree of Bachelor of Engineering
in BEng Electrical and Electronic Engineering

Department of Electrical & Electronic Engineering
University of Bristol

DECLARATION AND DISCLAIMER

Unless otherwise acknowledged, the content of this thesis is the original work of the author. None of the work in this thesis has been submitted by the author in support of an application for another degree or qualification at this or any other university or institute of learning.

The views in this document are those of the author and do not in any way represent those of the University.

The author confirms that the printed copy and electronic version of this thesis are identical.

Signed: Jongyoon Kim

Dated: 25/04/2021

Table of Contents

Single Image Super-Resolution for Text-based Images : Benchmark and GUI

Jongyoon Kim, *University of Bristol*

Abstract

Reading texts on low-resolution images is difficult. The low-resolution text-based images are usually produced due to the bad performance of the old camera or the target object captured on the image is far from the camera. A various number of methods to improve low-resolution image to high-resolution have been proposed. This problem is also known as Single-Image Super-Resolution (SISR). The SISR methods have been developed from mathematical and statistical methods to machine learning and deep-learning-based methods. The tool to general users trying to use SISR methods, especially targets to up-sample text-based images, is the project's main objective. Therefore, 12 SISR methods are assessed, and the Graphical User Interface is built in this project. The SISR methods up-sample four image sets with scaling factor of 2, 3, 4 and 8. The up-sampled images are compared with ground-truth images by using four image quality assessment methods. The benchmark emphasis that 'ICBI' is the dominating method to up-sample the low-resolution image in the experimenting condition with scaling factor of 2, 4 and 8. For the experiment setting with a scaling factor of 3, as 'ICBI' was not available to process, 'EDSR' and 'ESPCN' showed the best performing. The GUI is then developed by taking the code used for benchmark as backend, and the codes are wired into front-end code and user interface. Although the benchmark result showed that 'ICBI' is the best performing SISR method, the up-sampled image had some artefacts and wiggly lines compare to other methods. The reason for this issue and the limitation of image quality assessment methods are dealt on the discussion. Finally, this project is concluded by addressing future research related to the limitation and problems.

***Index Terms*—Image Processing, Image analysis, Image quality, High-resolution imaging, benchmark**

I. Introduction

Image magnification is typical action in various devices, mostly done on small portable devices such as smartphones and tablet PCs. The sampling resolution of the electronic devices is not high enough for users to enlarge the captured image, especially when the device is the old model. Furthermore, the image's content which is captured from the past with a low-resolution camera is difficult to understand. Therefore, the technique to improve the image's resolution is researched by estimating the unknown pixels between known pixels. This technique becomes a significant computer vision problem in various application fields [1], [2] such as medical image capturing [3]–[6], optical devices for biology [7]–[9], object recognition/detection for general purpose and surveillance/security purpose [10]–[14], amongst others.

The technique is called super-resolution (SR), which predicts the high-resolution (HR) image from a combination of the low-resolution (LR) images. The SR method can be visualized as Figure I-1, which is an example that used a scaling factor of 2. Each box on Figure I-1 represents a pixel from image, and the value in the box illustrates the colour value using RGB scale (R: Red, G: Green, B: Blue). The RGB values are limited in the range of 0~255. ‘0’ means none of the colour is included to the pixel (black) while

The code implementation of this project is available at: https://github.com/ArtemisDicoTiar/sisr_project. All images resulted from this project is on the separate Github repository: https://github.com/ArtemisDicoTiar/SISRProjectImage.

255 means the colour is fully applied to the pixel. The SR method takes the pixel values as a matrix: Equation (I-1). However, some SR methods trying to normalise the pixel's RGB scale into 0~1 to make the calculation during SR process simpler as multiplication of values below 1 will make the value smaller while 0~255 scale does not. The pixel value using 0~1 scale can be illustrated as Figure I-2, and the SR method will use the pixel value as matrix written in Equation (I-2.

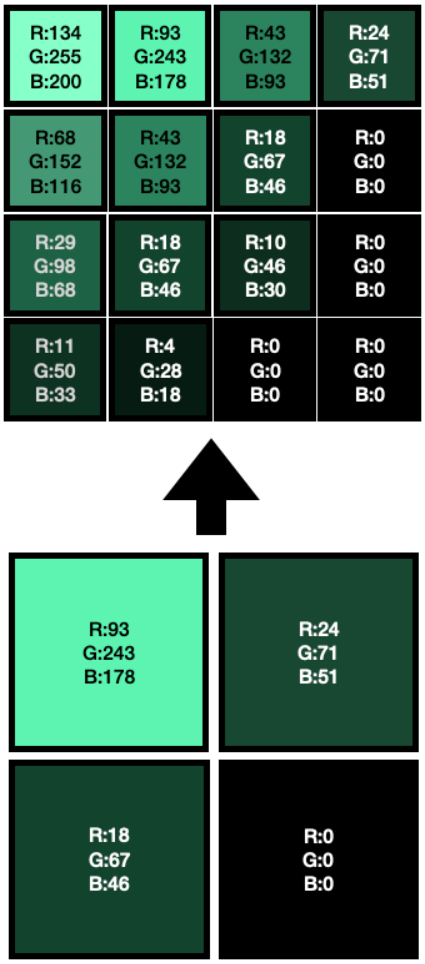

Figure I-1: Diagram for super-Resolution method basic understanding

$$\begin{bmatrix} [93 & 243 & 178] & [24 & 71 & 51] \\ [18 & 67 & 46] & [0 & 0 & 0] \end{bmatrix} \quad (I\text{-}1)$$

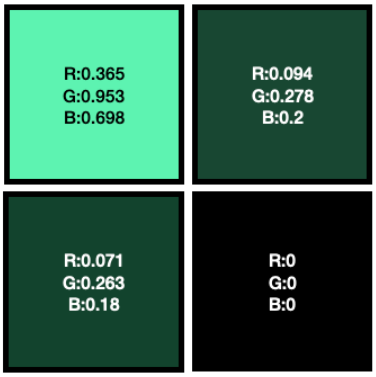

Figure I-2: Diagram of the pixel from an image using 0~1 scale of RGB

$$\begin{bmatrix} [0.365 & 0.953 & 0.698] & [0.094 & 0.278 & 0.2] \\ [0.071 & 0.263 & 0.18] & [0 & 0 & 0] \end{bmatrix} \quad (I\text{-}2)$$

There are various classical SR methods proposed in the computer vision community. Interpolation methods are proposed on early stage, includes bicubic interpolation [15], [16], Lanczos [17]. Other classical strategies are sparse representation methods [18], [19], example-based methods [20], [21], patch-based methods [21]–[23], edge-based methods [24], [25] and combination of other methods [26].

In addition to classical methods, the application of machine learning and deep learning to the field of computer vision leads to significant performance improvement where the classical methods come to a dead end as the classical methods use more complex mathematical equation which increased processing time and memory usage. For better efficiency of the SR method, direct end-to-end mapping with interpolated LR and HR images with Convolutional Neural Network (CNN) was proposed, called SRCNN [27]. Starting from SRCNN, other CNN methods such as Fast SRCNN (FSRCNN) [28], Deeply Recursive CNN (DRCNN) [29], Very Deep Convolutional Networks (VDSR) [30] and Efficient Sub-Pixel Convolutional Networks (ESPCN) [31] are proposed for shorter processing time and better up-sampled images. The deep learning model design can be differed by model frameworks, up-sampling methods, network

design, and learning strategies [32]. Most recent models use enhanced model designs and result in better performances, such as Super-Resolution Generative Adversarial Network (SRGAN) [33] and Content Adaptive Re-sampler (CAR) [34].

In this project, a comprehensive overview of various Single-Image Super-Resolution (SISR) methods will be given. There are SR surveys already done by others [32], [35]–[39], but this project is going to focus on the performance of the SR method on text-based images such as scanned images, images taken by mobile devices. The importance of up-sampling text-based images is due to the recognition either by human or machine. The text recognition and detection technologies challenge detecting characters from LR images that may have issues of blurry, degraded, and distorted pixels [40], [41]. Therefore, text-based images will be used for the dataset of the benchmark. Furthermore, the methods will also be assessed with general condition images and images with patterns to compare the performance with different input images.

The detailed contents of this project are outlined in the following. The SISR methods used for this project is illustrated in Section II, including a general explanation about each category of SISR methods. The way of how up-sampled image assessed is illustrated in Section III. The benchmark-setting and project implementation is then described in Section IV. This section includes benchmark pseudo-code and GUI pseudo code. Lastly, illustrating the benchmark result and limitation of the project concludes the paper (Section V - VII).

II. Single-Image Super-Resolution Methods

A. Interpolation methods

Interpolation is a function that satisfying following equation (II-1) for given image v where $v_{m,n}$ represents uniformly sampled pixels.

$$v_{m,n} = u(m,n) \text{ for all } m,n \in \mathbb{Z}(Integer) \quad (II\text{-}1)$$

There are three commonly used interpolation methods, Nearest Neighbour, Bilinear and Bicubic interpolation. The graphical view of the three interpolation methods is described in Figure II-1 [16].

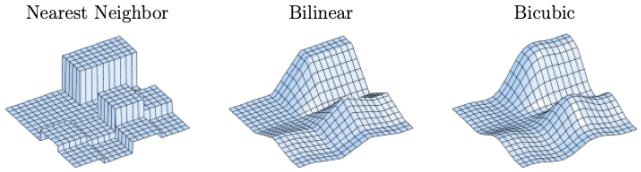

Figure II-1: Linear Image Interpolation methods applied to uniformly spaced input data [16]

Because basic interpolation algorithms are commonly used as a baseline to compare another method, this project selected four interpolation algorithms which are going to be described below.

Nearest Neighbor Interpolation

The Nearest Neighbour (NN) interpolation is the simplest interpolation method. The interpolated pixel is a duplication of the nearest pixel. From the Figure II-2, unknown pixel (u, v) can be estimated by comparing the distance between nearby known pixels, (i, j), (i+1, j), (i, j+1) and (i+1, j+1). The nearest known pixel is (i, j), so the unknown target pixel (u,v) will take the value same as (i, j) [42]. The explained method can be written as Equation (II-2), where [k] results round to nearest integer from k.

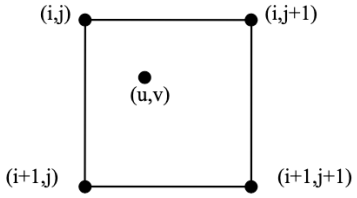

Figure II-2: Diagram of nearest neighbor interpolation algorithm [16]

$$p_{unknown}(u,v) = p_{known}([u],[v]) \quad (II\text{-}2)$$

Bilinear Interpolation

The bilinear interpolation processes like nearest neighbour interpolation, but it uses a linear function to estimate the unknown pixel value. From Figure II-3, A, B, C and D pixels are set a known pixel and the values are (i, j), (i+1, j), (i, j+1) and (i+1, j+1). To estimate the value of unknown pixel P, pixel E and F should be estimated by applying linear function with known

pixel values on the same rows as a primary stage. Then, pixel P can be calculated by applying a linear function with estimated pixel E and F. The explained processes can be illustrated using Equation (II-3) - (II-5) [42].

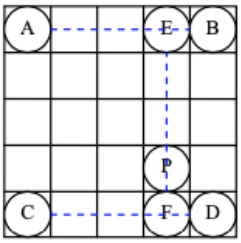

Figure II-3: Diagram of bilinear interpolation algorithm [42]

$$\begin{aligned} f(E) &= f(i+v, j) \\ &= \{f(B) - f(A)\} * v + f(A) \\ &= \{f(i+1, j) - f(i,j)\}v + f(i,j) \end{aligned} \quad \textit{(II-3)}$$

$$\begin{aligned} f(F) &= f(i+v, j+1) \\ &= \{f(D) - f(C)\} * v + f(C) \\ &= \{f(i+1, j+1) - f(i,j+1)\}v + f(i,j+1) \end{aligned} \quad \textit{(II-4)}$$

$$\begin{aligned} f(P) &= f(i+v, j+u) \\ &= \{f(F) - f(E)\} * u + f(E) \\ &= (1-u)f(E) + uf(F) \\ &= (1-u)(1-v)f(i,j) - (1-u)vf(i+1,j) \\ &\quad + u(1-v)f(i,j+1) \\ &\quad + uvf(i+1,j+1) \end{aligned} \quad \textit{(II-5)}$$

Because the calculation assumes that the target pixel lies on the linear relationship between nearby two pixels, the method is called bilinear interpolation.

Bicubic Interpolation

Bicubic interpolation is similar to bilinear interpolation, but it refers 16 nearby known pixels, unlike bilinear, which uses four nearby known pixels. The unknown pixel (u,v) on Figure II-4 can be estimated by a similar process done on bilinear interpolation but use bicubic estimation to four pixels. The estimation of a pixel on (u,v) involves nearby pixels to be also estimated. For this case, it should estimate pixel having x-axis value of 'u' from all rows. With the two pixel values, the target pixel (u,v) can be estimated using the estimated pixels with the x-axis value, 'u'. The way of estimating the pixel value follows Figure II-5, but it uses cubic estimation [42].

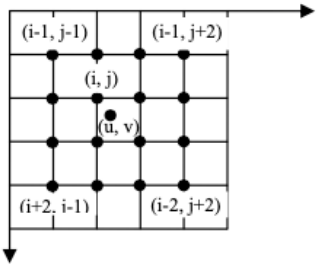

Figure II-4: Bicubic interpolation algorithm diagram [42]

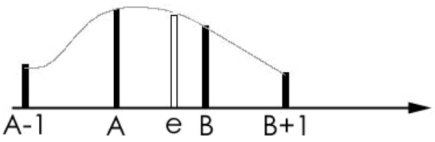

Figure II-5: nonlinear interpolation algorithm diagram [42]

Because the influence of each known pixel is stronger than bilinear interpolation, the estimated unknown pixel gets value more natural to nearby pixel values (Figure II-1). Furthermore, the bicubic interpolation kernel is smoother than bilinear interpolation, so the up-sampled image shows a smoother edge [16].

Lanczos Interpolation

Lanczos interpolation uses kernel written on Equation (II-6) and (II-7) [43]. The Lanczos kernel, which is windowed sinc function, is applied to estimate the unknown pixel value with few pixels. When 2nd order is chosen then, 16 nearby pixels are used to estimate unknown pixel and 3rd order selection make 36 neighbour pixels are used to calculate unknown pixel value. The high order leads to refer more pixels, the computation time increases as more addition and multiplication with sin function is applied then other interpolation methods[44], [45].

$$Lanczos_2(x) = \begin{cases} \dfrac{\sin(\pi x)}{\pi x} \dfrac{\sin\left(\frac{\pi x}{2}\right)}{\frac{\pi x}{2}}, |x| < 2 \\ 0, |x| \geq 2 \end{cases} \quad \textit{(II-6)}$$

$$Lanczos_3(x) = \begin{cases} \dfrac{\sin(\pi x)}{\pi x} \dfrac{\sin\left(\frac{\pi x}{3}\right)}{\frac{\pi x}{3}}, |x| < 3 \\ 0, |x| \geq 3 \end{cases} \quad \textit{(II-7)}$$

Interpolation performance overall expectation

According to [42], the subjective evaluation of the nearest neighbour comments that the up-sampled pixel looks like a mosaic, bilinear is blurry and not sharp and bicubic looks fuzzy but sharper. Besides, the contour of the image is unclear for both the Nearest neighbour and bilinear interpolation. The bilinear interpolation methods are assessed as having a serrate edge. The bicubic is the most improved interpolation compare to the other two methods. The mean of SNRs from 10 result image from three interpolation methods shows that Bicubic shows the highest value. In contrast, the Nearest neighbour shows the lowest value, which means that bicubic shows the best performance.

B. Deep-learning methods

Deep learning methods are selected for the project as the methods are the state-of-the-art technique in SISR research fields. The reason for each method selection is described at the end of each method description.

Fast Super-Resolution Convolutional Neural Network (FSRCNN)

The FSRCNN is an enhanced model of SRCNN, which has faster runtime and better PSNR (PSNR is described in Section III.B). Therefore, to understand FSRCNN, reviewing SRCNN is necessary.

The SRCNN is done by four steps illustrated in Figure II-6 [46]. Primarily, bicubic interpolation is applied to LR image to achieve desired resolution as pre-processing, which is not described in Figure II-6. With the interpolated image, 9*9 convolution is applied to the image for patch extraction from LR image and each patch is represented as a high-dimensional vector. The vector is comprehended as feature maps. The feature maps are now mapped to other feature maps non-linearly using 1*1 convolution. The non-linearly mapped feature maps are then connected to the HR image by reconstruction process patch-wise with 5*5 convolution layer.

The SRCNN showed better performance than the A+ algorithm by about 0.2dB, KK algorithm by 0.3dB and NE+LLE algorithm by 0.7dB [46]. The running time is faster than the A+ algorithm when 9*9, 1*1 and 5*5 convolution is used for each step, respectively. However, when the convolution filter size increased from 1*1 to 3*3 or 5*5, the running time gets slower than the A+ algorithm as feature maps mapping requires more computation processes.

Even though SRCNN showed better performance than traditional hand-crafted statistical methods, the time complexity of the SRCNN becomes high when the size of the target HR image increases, as the time complexity is illustrated in Equation (II-8). According to Equation (II-8), 'f' means the size of the filter, 'n' means the number of filters, and S_HR refers to the size of the HR image. Therefore, FSRCNN is proposed to achieve a faster processing speed [28].

$$O\{(f_1^2 n_1 + n_1 f_2^2 n_2 + n_2 f_3^2) S_{HR}\} \quad \textit{(II-8)}$$

The FSRCNN is modified from SRCNN with three characteristics. A deconvolution layer is implemented at the last stage of the network. Furthermore, the image does not have to be interpolated at the first stage so that the HR image can be learned from the LR image. Besides, the structure of the mapping layer has been changed as reducing input feature dimension before mapping and increase the feature dimension as before reducing. This process sets to be repeated 'm' times to control the accuracy and complexity of the mapping stage. Lastly, filter size has been scaled down, whereas the number of mapping layers has been increased. The overall process is illustrated in Figure II-7, which also shows the difference between SRCNN and FSRCNN.

The SRCNN was targeted for this project, but the processing time was longer than expected, so FSRCNN is chosen to secure more computing time for other methods and image quality analysis.

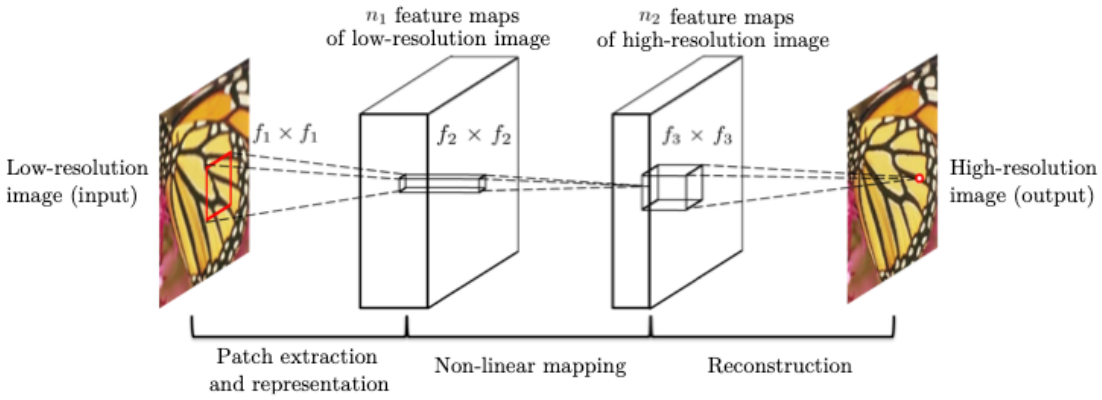

Figure II-6:Diagram of SRCNN process [46]

Very Deep convolutional network (VDSR)

Increasing the depth of the convolutional network improves the accuracy of image recognition on a large scale [47]. Therefore, Kim et al. [30] tried to implement a very deep convolutional network for the SR problem, so their final model uses 20 weight layers. The VDSR model architecture is illustrated in Figure II-8. The LR image is interpolated, and the image is used as input of the network. While proceeding the interpolation step, any interpolation can be used. Therefore, four interpolation methods illustrated in section II.A will be applied before VDSR in this project. The interpolated image is then processed on cascaded convolution layers, and ReLu layers 'D-1' times and 'D'th convolution layer result will be added to the interpolated image. The convolution and ReLu layers use residual information of the image and it is added to interpolated image on the last stage. They used residual information for model learning because it converges faster than the standard CNN structure. Also, CNN uses residual information to show better performance. By comparing residual learning and standard learning with the same epochs, the PSNR difference is about 9.5 dB after ten epochs with various initial learning rate, 0.1, 0.01 and 0.001.

Furthermore, the convergence of residual learning is smoother and more stable than standard learning[30]. Lastly, the residual learning also led to faster runtime even though the convolution layer increased compared to SRCNN. The PSNR of VDSR is higher than SRCNN by about 0.9dB.

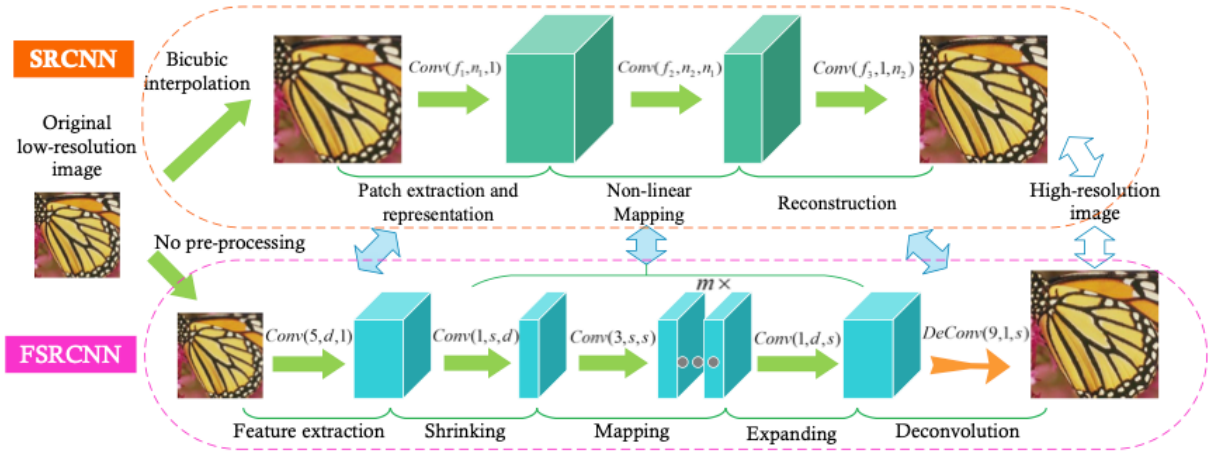

Figure II-7: Diagram of comparison between SRCNN and FSRCNN (Conv: convolution layer) [28]

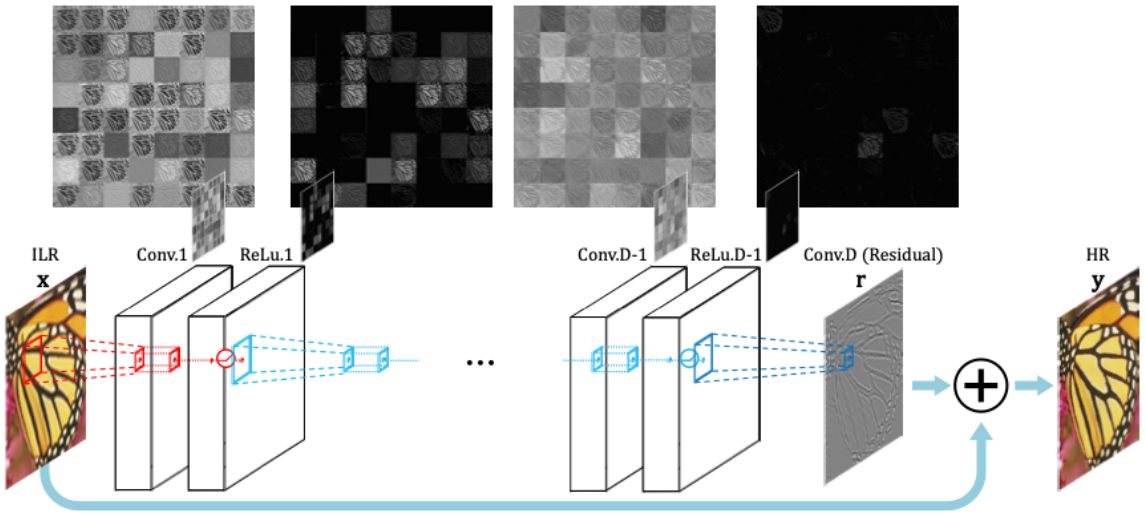

Figure II-8: Diagram of VDSR network structure [30]

The VDSR is the next state-of-the-art model for SISR after SRCNN, so the VDSR is selected for this project. Moreover, it is meaningful to see this deeper CNN structure that uses residual information also shows good performance for text-based images.

Deeply Recursive Convolutional (Neural) Network (DRC(N)N)

The same author of VDSR has proposed the DRCN, but the method they used for SR was different. The network uses a recursive layer, which repeats the same filter and feature map, for simplicity of the model parameter even the depth is increased due to additional convolution layers. The DRCN's network is consists of three sub-networks: embedding, inference and reconstruction networks (illustrated in Figure II-9). The embedding network takes interpolated LR image and converts it into a set of feature maps. Then this set of feature maps are passed to the inference network. The inference network is constructed with recursion of a convolution layer, ReLU layer and mapping layer. After the inference network, the output feature maps are transformed into the original image space, which results in an HR image. The recursive model is simple but powerful to implement. However, training a deep recursive network is difficult because of the vanishing gradient problem. Therefore, Kim et al. [29] applied two technique to solve this issue: recursive-supervision and skip-connection.

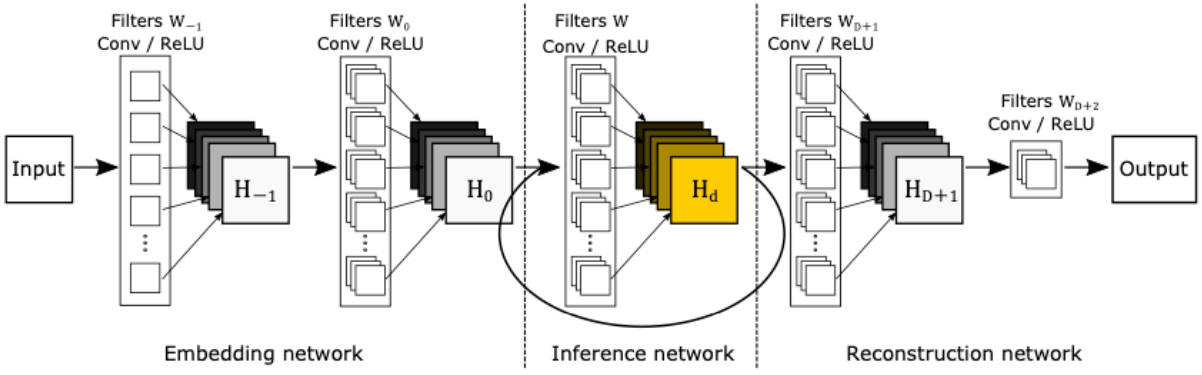

Figure II-9: Diagram of DRCN model structure [29]

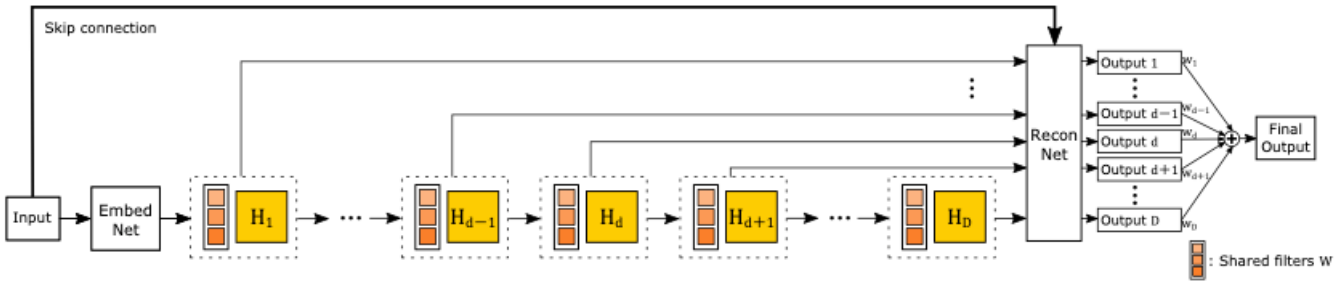

Figure II-10: Diagram of DRCN with recursive-supervision and skip-connection [29]

The reconstruction network takes feature maps from the inference network, which means feature maps from any recursion step can be passed. Therefore, interim feature maps for each recursion are also passed to the reconstruction network for the reconstruction process. This results in the final output to be a summation of each feature maps multiplied by weight for each map written as Equation (II-9). This ensemble improves performance by opposing the effect of vanishing/exploding gradients as one backpropagation path is reduced. Furthermore, the significance of choosing the optimal number for recursion is reduced because when the recursions are too deep for the image, the weight for deeper recursion will become smaller.

$$y = \sum_{d=1}^{D} w_d * \hat{y}_d \qquad \textit{(II-9)}$$

Also, skip connection is added to the reconstruction network, as illustrated in Figure II-10. The network's capacity can be saved for storing result feature maps of every recursion from the inference network. Also, the exact copy of the feature map can be used to predict the reconstruction network. This can be done as LR image and HR image does not have a considerable difference, which means most pixels will have similar value. This understanding will improve the performance of the learning process. Therefore, the vanishing gradients problem will be solved.

The DRCN model is selected for this project to evaluate how a recursive network works for text-based images, even with deep layers like VDSR.

Efficient Sub-Pixel Convolutional Neural network (ESPCN)

The ESPCN model is the first model that uses LR image space as input of the model instead of using an interpolated image which is HR image space. The model also uses CNN structure to process SR. The efficient sub-pixel convolution layer is introduced to the network structure. As interpolating the LR image to HR image space can be replaced into the complex filter that can be trained for each feature maps, the computation complexity is reduced, and the performance of the SR is improved. According to Shi et al. [31], the ESPCN showed 0.15dB PSNR improvement for images and 0.39dB PSNR improvement for videos.

The efficient sub-pixel convolution neural network is constructed as illustrated in Figure II-11. The sub-pixel convolution process is done on the hidden layer for 'L' times with fl*fl convolution filters on the 'L-1'th layer. On the last layer, the feature map is used to reconstruct the HR image output.

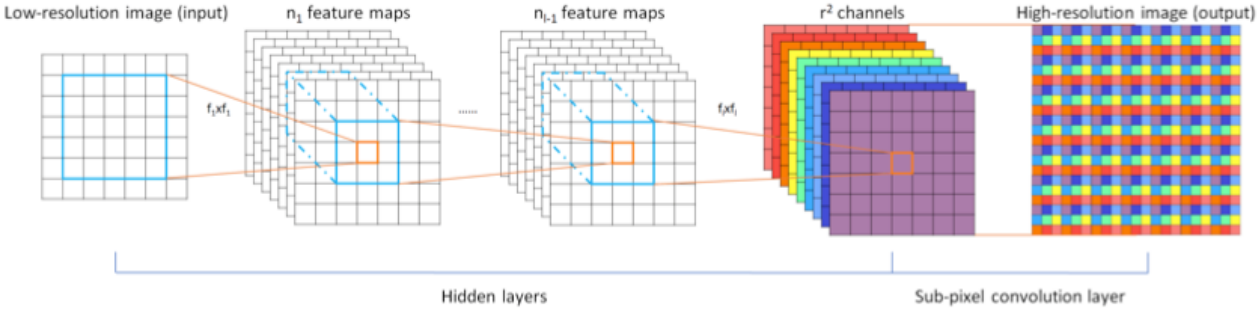

Figure II-11: Diagram of ESPCN network [31]

The ESPCN model is selected to compare the performance difference between the model that uses interpolated LR image for model input and ESPCN, which directly takes LR image as input. The interpolation process before applying to the model can lose the edge information during interpolation. Therefore, the ESPCN model is listed on the benchmark target models.

Laplacian Pyramid Super-Resolution Network (LapSRN)

There are few problems with formerly designed SR models. The ESPCN applies bicubic interpolation before main process of the model, which causes an increase in computational cost. In addition, it does not provide high-frequency information to the reconstruction stage. Besides, the previous models use L2-loss for model optimisation. The L2-loss (also known as Mean-Square Error) is challenging to pass multi-model distribution of HR patches, and an identical LR patch may result in diver HR patches. Therefore, L2 loss causes over-smoothed and inconsistent human visual perception on reconstructed HR image. Lastly, previous models had one stage for up-sampling to HR image. As only one step is applied for up-sampling, up-sampling the image with a significant scale factor makes the learning process for the mapping function difficult. To solve these issues, LapSRN is proposed [48].

The LapSRN primarily applied progressive up-sampling with residual information from each process stage for reconstructing the image. As the up-sampling process is done by the power of two, the target up-sampling scale factor is available for 2, 4 and 8. The model directly extracts features from the LR image instead of applying interpolation as pre-process. This reduces computation power similar to ESPCN.

The name of LapSRN originated from the use of the Laplacian Pyramid for the feature extraction branch, as described in Figure II-12. The Laplacian Pyramid is an encoding process that samples the image with Laplacian operators. Therefore, it can encode outstanding image features, and so it is applicable to image analysis processes [49].

The feature extraction branch is structured with multiple levels of construction layers and one deconvolution layer. The deconvolution layer on each level outputs to the next level's convolution layer and reconstruction branch for summing the residual information with it. The feature maps from lower levels are shared with higher levels, so the network's non-linearity can be increased, and the network can learn more complex mapping.

The other branch, the image reconstruction branch, up-samples the image by the scale of 2 with a deconvolution layer. Then the resulting image is combined with the residual image from the feature extraction branch.

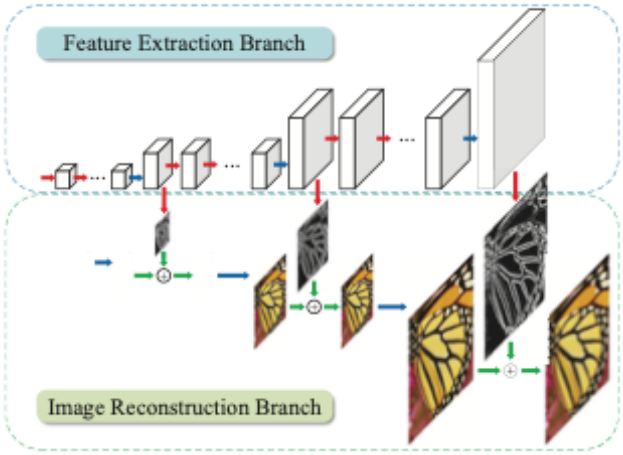

Figure II-12: Diagram of LapSRN structure that consists of feature extraction branch and image reconstruction branch [48]

The text-based images demand to have more precise edge information when the image is up-sampled. However, the previously explained models have only one stage for up-sampling contrast to feature extraction branch of LapSRN, which has multiple stages. This can cause loss of edge information, so this model is selected for the project to be benchmarked.

Enhanced Deep Residual Network (EDSR)

The EDSR model is the most recent model used for this project. This model based on SRResNet to solve the SR problem, but optimised network architecture is used for EDSR by analysing the ResNet and removing unnecessary parts from ResNet[50]. Primarily, the architecture of EDSR is illustrated in Figure II-13. The LR image is entered into the convolution layer and then a few numbers of residual blocks. The convolution layer is applied after the last residual block. The image after the last convolution layer is summed with the initial image that passed the first

convolution layer as residual information is handled on residual blocks. In the last stage, up-sampling is processed by deconvolution layers.

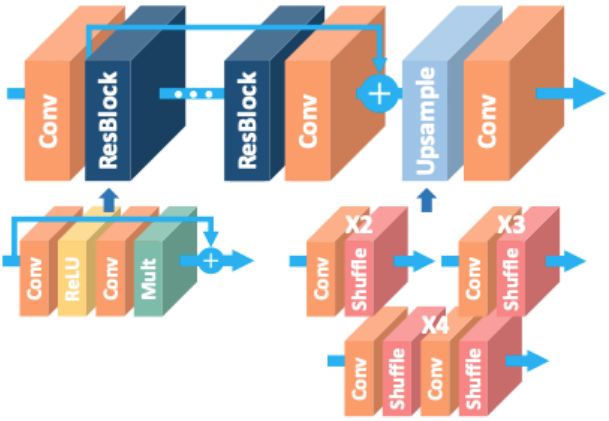

Figure II-13: Diagram of EDSR architecture [50]

The critical point of this model is that although the EDSR method based on SRRestnet, the residual block has been optimised by removing the convolution layer and batch normalisation (BN) layer. As the batch normalisation process normalises the feature, removing it can give more flexibility to the network. Also, this optimisation led to performance increase and memory usage reduction.

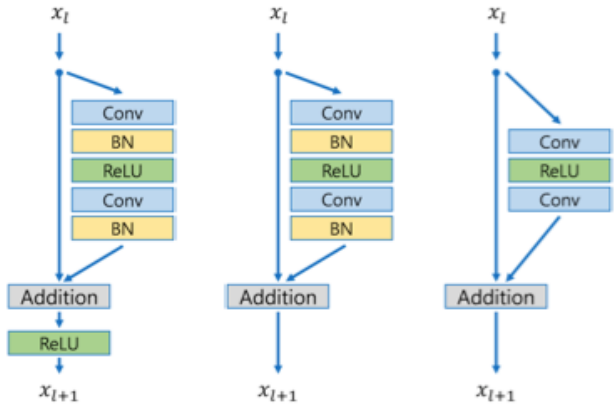

Figure II-14: Comparison of residual blocks between original, SRResNet and EDSR (from left to right) [50]

This model is selected for the project to see the residual network's performance when applied for text-based images. Because the residual network is effectively used to avoid vanishing or exploding gradient problem, the model can show better performance than others. Besides, a deep-learning SR survey paper by Yang et al [35], reported that EDSR showed the best performance as 32.62 dB (PSNR). Therefore, the EDSR is selected to benchmark for text-based images.

C. Edge-preserving methods

This section describes the methods that use hand-crafted features, unlike deep-learning models. The hand-crafted up-sampling methods targeted to find a better interpolation method and correct the interpolated pixel to better value with the objective function. Because the hand-crafted features/methods must be built by human intuition, it can miss some significant features of the image or trifle feature that results in a significant difference in the performance. However, the hand-crafted features are usually built with mathematical equations for a specific purpose. This means the method can show good result for specific condition. For this project, the methods that understand the edge of the character is needed. Therefore, edge-preserving hand-crafted methods are included for this project.

Improved New Edge-Directed Interpolation (iNEDI)

The up-sampling methods that preserve edge features have been proposed, but they had instability and computational difficulty. The iNEDI method improved from NEDI, which used bilinear interpolation and adaptive interpolation to perform better than linear interpolation. Simultaneously, it reduces the computational complexity of the covariance-based adaptive interpolation method [51]. The iNEDI was proposed to solve the following limitation from NEDI [52].

The use of squared window produced directional artefacts and made the algorithm non-isotropic, so iNEDI replaced with an approximately circular window to reduce the side effects.

The NEDI method uses coefficients that need to be calculated even the covariance is unchanging, which will lead to a bad solution from small error. Although this issue has been solved from the NEDI using bilinear interpolation for the similar local grey level variation above the threshold value, iNEDI proposed to use bicubic interpolation. This cannot improve the performance when grey-scale variation below threshold value. However, it can have better accuracy and edge direction preservation.

The significant problem of NEDI is how to make the window used for pixel value estimation

to be on the almost identical edge. This is the most important problem because when the window does not follow the edge, it can cause the predicted pixel to have a strange value compared to nearby pixels, as illustrated in Figure II-15. This is the case when having a more vital constant covariance constraint. This issue can be solved by growing region that observes neighbour pixels. This process is called edge "segmentation".

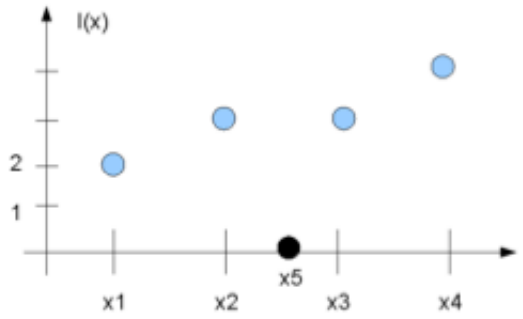

Figure II-15: Example of 1D interpolation showing wrong covariance interpolation result (Black pixel is predicted pixel and blues are obtained from a weighted sum of two neighbour pixels) [52]

While removing deficient regions, the over-constrained system can cause a significant error of interpolated element vector from a small error in input. This issue is solved in iNEDI, by adding a constant value to the grayscale to make the value to be away from zero.

The final problem from NEDI is that the interpolated pixel gets a changed value for global brightness, which means the interpolated value also considers the difference between global absolute values. This problem has been solved by applying subtraction of the average of the four neighbour intensities instead of NEDI constraint.

The iNEDI does not show dramatic improvement assessed by PSNR where it has a similar value compare to bicubic interpolation. However, the image showed a more precise contour, and this leads to the best qualitative assessment having an average value of 3.82/5, where bicubic achieve 2.64/5. Therefore, this method is selected for this project which is expected to have good performance to preserve the edge of text objects.

Interactive Curvature Based Interpolation (ICBI)

The ICBI method is proposed by Reddy et al. [53] to improve the performance of the up-sampling method by using two-step grid filling and iterative correction of interpolated pixels. The pixel correction has been done by obtaining minimum value from an objective function that refers a second-order directional derivatives of image intensity.

The two steps grid filling is processed with the FCBI algorithm with initialised new values. Primarily, two diagonal directions are calculated using eight neighbour pixel values. The second-order derivatives $I_{11}(2i+1, 2j+1)$ and $I_{22}(2i+1, 2j+1)$ can be calculated by Equation (II-10) and (II-11) it is illustrated as a graphical explanation in Figure II-16. Then, the average of the two neighbours in the direction is assigned to the point described in Equation (II-12).

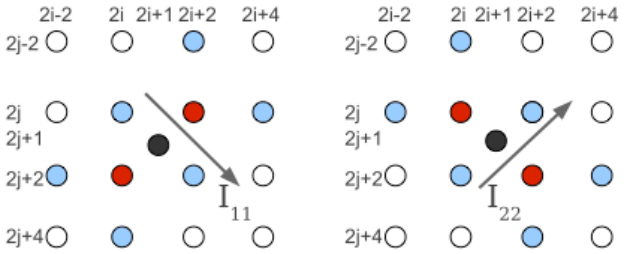

Figure II-16: Diagram of FCBI algorithm, estimating unknown black pixel. [53]

$$\tilde{I}_{11}(2i+1,2j+1) = I(2i-2,2j+2) + I(2i,2j) + I(2i+2,2j-2) + 2(2i,2j+4) + I(2i+2,2j+2) + I(2i+4,2j) - 3I(2i,2j+2) - 3I(2i+2,2j) \quad (II\text{-}10)$$

$$\tilde{I}_{22}(2i+1,2j+1) = I(2i,2j-2) + I(2i+2,2j) + I(2i+4,2j+2) + I(2i-2,2j) + I(2i,2j+2) + I(2i+2,2j+4) - 3I(2i,2j) - 3I(2i+2,2j+2) \quad (II\text{-}11)$$

$$\begin{cases} \dfrac{I(2i,2j) + I(2i+2,2j+2)}{2}, if\ \tilde{I}_{11} < \tilde{I}_{22} \\ \dfrac{I(2i+2,2j) + I(2i,2j+2)}{2}, otherwise \end{cases} \quad (II\text{-}12)$$

The interpolated values are retouched by the iterative procedure, which targets making the "energy" function obtain minimum. The function is given by combining each interpolated pixel with a different weight which depends on the

local continuity of the second-order derivatives. The function components are desired to have minima when the desired image properties are achieved. However, computing this procedure to every pixel requires a higher computational cost. Therefore, ICBI proposes to apply a greedy strategy just for calculating the local minimum of every pixel, which will be used as a reasonable initial value to iterate the procedure with a shorter processing time.

The objective function, which is the "energy" function for this method, should be minimised. The first interpolation step can result in the energy function as Equation (II-13). The weights ('omega' on Equation (II-13)) are set to 1 if the first-order derivative in the same direction is smaller than the threshold value and 0 otherwise. This method will lead to an image not having smoothness when there is a massive discontinuity between pixels. Therefore, this term is called "curvature smoothing", and the name of this algorithm to be ICBI. The optimisation process of minimising the sum of the "curvature smoothing" term removes artefacts effectively, but it can lead to an over-smoothed image. By converting the second-order derivative estimation as actual directional curvature, the curvature smoothing effect can be reduced.

$$
\begin{aligned}
&U_C(2i+1,2j+1)\\
&=\omega_1(|I_{11}(2i,2j)-I_{11}(2i+1,2j+1)|\\
&\qquad +|I_{22}(2i,2j)-I_{22}(2i+1,2j+1)|)\\
&+\omega_2(|I_{11}(2i,2j)-I_{11}(2i+1,2j+1)|\\
&\qquad +|I_{22}(2i,2j)-I_{22}(2i+1,2j+1)|)\\
&+\omega_3(|I_{11}(2i,2j)-I_{11}(2i+1,2j+1)|+\\
&|I_{22}(2i,2j)-I_{22}(2i+1,2j+1)|)\\
&+\omega_4(|I_{11}(2i,2j)-I_{11}(2i+1,2j+1)|+\\
&|I_{22}(2i,2j)-I_{22}(2i+1,2j+1)|)
\end{aligned}
\qquad \textit{(II-13)}
$$

According to Giachetti and Asuni [54], the ICBI showed +0.71 dB and +0.42 dB of PSNR for a scale factor of 2 and 4, respectively. Besides, as this method effectively makes the edge clearer, which is the vital functionality to up-sample the text-based images, the ICBI is chosen for this project to be assessed.

III. Image Quality Assessment

While the image up-sampling and predicting methods have been developed and researched, evaluating the resulting image from the methods has been designed in various strategies, usually with statistical analysis. The image quality analysis methods are categorised into two, subjective image quality assessments and objective image quality assessments [55].

A. Subjective Image Quality Assessments

The subjective assessment focuses more on human perception than numerical analysis. This assessment is done by taking a survey about the images, including up-sampled images with various methods. Because this SR technology is developed to have better visual understanding from low-resolution images for human, it is crucial to have the perception of human for assessing the up-sampling methods. However, this assessment requires several people to assess, the cost and time requirement is high. Therefore, this subjective assessment has been dropped for this project. Furthermore, some IQA methods are found that the methods tried to have a high correlation between subjective assessment and numerical analysis during the literature review. By using the method, the project can have perspective of how human evaluate the up-sampled images, indirectly.

B. Objective Image Quality Assessments

Peak Signal-to-Noise Ratio (PSNR)[55]

The PSNR is the most used objective metric for image quality analysis. This metric is straightforward and convenient to use. However, when this metric is developed, human visual system was not considered. Therefore, PSNR cannot correlate highly with human visual perception. The more excellent PSNR value represents a higher similarity between the up-sampled image and the original image based on the equation derived. The value of PSNR is calculated by comparing pixel by pixel and not trying to understand the structure of the image. This is the reason why it cannot highly correlate with the human visual perception. Even though the image with high PSNR can look weird, this can be a good numerical value to assess the up-

sampling method, which evaluates the similarity between the original and up-sampled image. The equation describing the PSNR is written on Equation (III-1).

$$PSNR = 20\,log_{10}\,MAX_{Original} - 10\,log_{10}\,MSE \quad \textit{(III-1)}$$

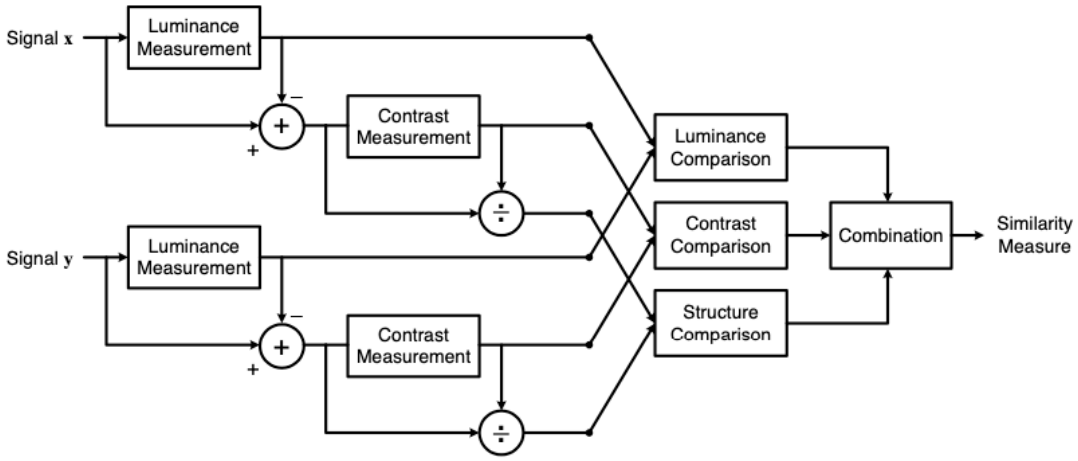

Figure III-1: SSIM measurement flow system diagram [56]

Root-Mean-Square Error (RMSE)

Like PSNR, this also compares the original ground-truth image and up-sampled image pixel by pixel. Based on the equation written on equation (III-2), the smaller value of RMSE means the similarity between the two images is high.

$$RMSE = \sqrt{MSE}$$

$$MSE = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}[Original\ -Upsampled]^2 \quad \textit{(III-2)}$$

Structural Similarity Index Measure (SSIM)

SSIM method is developed based on that human visual recognition uses structural information of an image as a feature to understand the quality of the image [56]. This method is different from PSNR and RMSE as SSIM tried to handle classical image quality assessments' problem by comparing local patterns, including luminance and contrast of the image. In conclusion, this method assesses reduction in structural information following the process described in Figure III-1. The equation to obtain SSIM is written on Equation (III-3), Greek letter mu means mean value, sigma means standard deviation of each image (x and y). The sigma with the subscript of x and y means cross-covariance of x and y images (two image x and y is used to compare the similarity). The c with different subscripts is a small constant for preventing instability of the equation as $\mu_x^2 + \mu_y^2$ reaches zero and a similar issue for deviation bracket.

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \quad \textit{(III-3)}$$

Feature Similarity Index Metric (FSIM) [57]

FSIM method uses low-level features to imitate human visual perceptions. The primary feature used for FSIM is the phase congruency (PC, PC in equation) which measures the significance of local structure. However, the phase congruency does not include contrast information, whereas the human visual system gets information from the contrast. Therefore, Gradient Magnitude (GM, G in equation) is applied as a secondary feature that can be calculated easily by applying convolution of gradient operators such as Sobel, Prewitt and Scharr operators [58], [59]. The gradient operator applied for partially differentiated f(x) is described in

Table III-1. With these two features extracted from the image, the local similarity map is primarily calculated, and then the similarity map is pooled up into a single index value, FSIM. The index can be calculated with Equation (III-4). The equation uses matrix x, which is the matrix of pixel values abstracted by the example flow illustrated in Figure IV-1. The Ω on summation (upper case Sigma) is used to inform the whole image is used for calculation. All 'T' on equations mean constant to add the stability of each function. 'I' and 'Q' represents the chrominance information as FSIM uses the 'YIQ' colour scale to assess the image.

$$FSIM_c = \frac{\Sigma_\Omega S_{PC} \cdot S_G(\mathrm{x}) \cdot \left[S_I(\mathrm{x}) \cdot S_Q(\mathrm{x})\right]^\lambda \cdot PC_m(\mathrm{x})}{\Sigma_\Omega PC_m(\mathrm{x})}$$

Where

$$\begin{aligned} S_{PC}(\mathrm{x}) &= \frac{2PC_1(\mathrm{x}) \cdot PC_2(\mathrm{x}) + T_1}{PC_1^2(\mathrm{x}) + PC_2^2(\mathrm{x}) + T_1} \\ S_G(\mathrm{x}) &= \frac{2G_1(\mathrm{x}) \cdot G_2(\mathrm{x}) + \mathrm{T}_2}{G_1^2(\mathrm{x}) + G_2^2(\mathrm{x}) + T_2} \\ S_I(\mathrm{x}) &= \frac{2I_1(\mathrm{x})\mathrm{I}_2(\mathrm{x}) + T_3}{I_1^2(\mathrm{x}) + \mathrm{I}_2^2(\mathrm{x}) + \mathrm{T}_3} \\ S_Q(\mathrm{x}) &= \frac{2Q_1(\mathrm{x})\mathrm{Q}_2(\mathrm{x}) + T_4}{Q_1^2(\mathrm{x}) + \mathrm{Q}_2^2(\mathrm{x}) + \mathrm{T}_4} \\ PC_{m(\mathrm{x})} &= \max(PC_1(\mathrm{x}), \mathrm{PC}_2(\mathrm{x})) \end{aligned} \qquad \textit{(III-4)}$$

Table III-1: Partial derivatives of f(x) with different gradient operators

	$G_x(\mathrm{X})$	$G_y(\mathrm{X})$
Sobel	$\frac{1}{4}\begin{bmatrix}1 & 0 & -1\\ 2 & 0 & -2\\ 1 & 0 & -1\end{bmatrix} * f(\mathrm{X})$	$\frac{1}{4}\begin{bmatrix}1 & 2 & 1\\ 0 & 0 & 0\\ -1 & -2 & -1\end{bmatrix} * f(\mathrm{X})$
Prewitt	$\frac{1}{3}\begin{bmatrix}1 & 0 & -1\\ 1 & 0 & -1\\ 1 & 0 & -1\end{bmatrix} * f(\mathrm{X})$	$\frac{1}{3}\begin{bmatrix}1 & 1 & 1\\ 0 & 0 & 0\\ -1 & -1 & -1\end{bmatrix} * f(\mathrm{X})$
Scharr	$\frac{1}{16}\begin{bmatrix}3 & 0 & -3\\ 10 & 0 & -10\\ 3 & 0 & -3\end{bmatrix} * f(\mathrm{X})$	$\frac{1}{16}\begin{bmatrix}1 & 0 & -1\\ 2 & 0 & -2\\ 1 & 0 & -1\end{bmatrix} * f(\mathrm{X})$

IV. Benchmark Setting and Implementation

A. Image Datasets

The image sets for this benchmark are categorized into three, and four image sets are used. 'Set5' and 'Set14' in 'General scene images', 'Urban100' in 'Pattern images' and 'KAIST text scene database tc-100' for 'text-based images'. However, the computing resource for this project is only a laptop with a CPU (i7-8750H). The time for running models/methods requires more than 10 minutes for edge-preserving methods. Therefore, some image sets are reduced to 10 images instead of using whole images. Therefore, a total of 39 images used for this project.

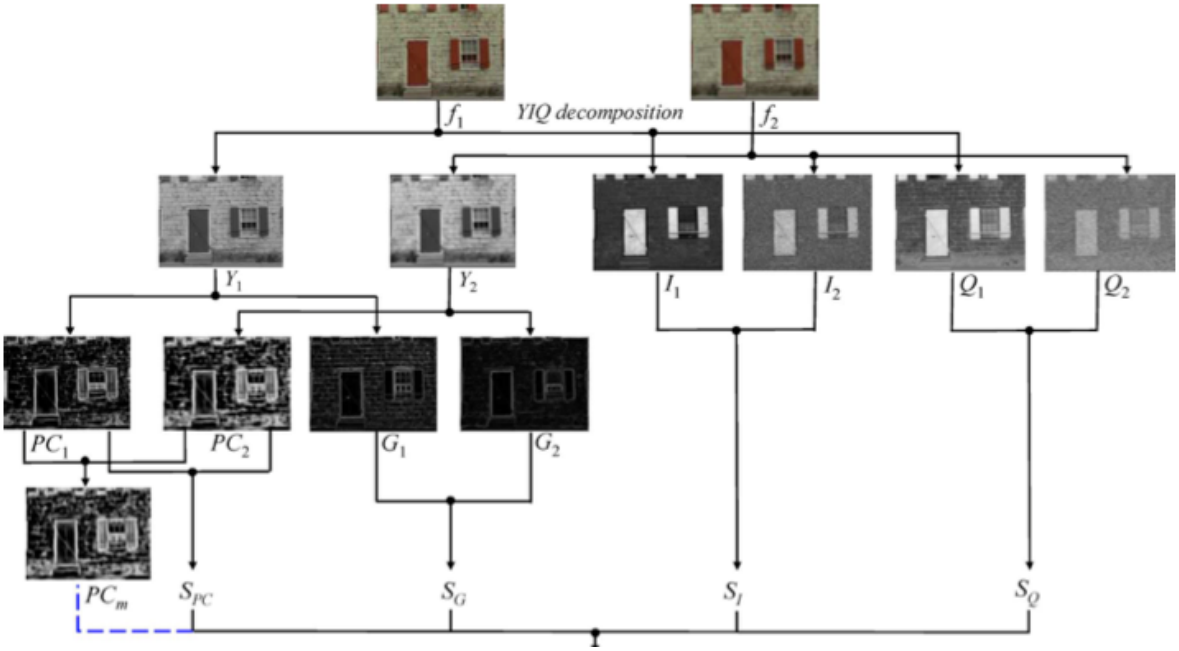

Figure IV-1: Diagram of FSIM computation flow (f1: original image, f2: processed image)) [57]

General Scene Images: Set5, Set14

The general scene image sets are used for the control group to compare the performance from text images. The most commonly used general image sets, Set5 [60] and Set14 [61], are used for this project. The set5 includes five images, and set14 includes 14 images. Examples of images are “bird”, “butterfly wing”, “Lena” (woman with dress worn) and “barbara”.

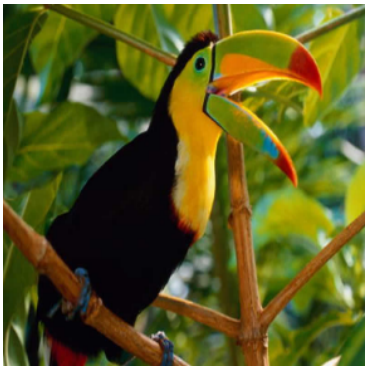

Figure IV-2: Set5 example image (bird) [60]

Figure IV-3: Set14 example image (barbara) [61]

Pattern Images: Urban100

The pattern image set from Urban100 [62] includes images from urban, including patterns such as grid shape because of windows on a skyscraper. The pattern image set is included in this project to assess the distortion from the up-sampling methods. However, this image set includes 100 images, the runtime for processing benchmark increases dramatically. Therefore, only the first ten images are selected.

Figure IV-4: Urban100 example image (img_005) [32]

Text Images: KAIST Scene Text Database-tc100

The text-based images are the essential dataset for this project. To consider the text-based image taken by the mobile camera, scanned documentation and images at a standstill are discarded. Therefore, image is taken from the various environment, including indoor and outdoor, and different light conditions are used for this project [63], [64]. Similar to 'Urban100', only the first ten images are used to reduce the benchmark runtime.

Figure IV-5: KAIST Scene text database-tc100 example image [32]

B. Color channel

The most commonly used image colour channel is RGB, but the YCbCr colour channel is also used for Super-Resolution [46]. RGB channel expresses a pixel with three colours, Red, Green and Blue, whereas YCbCr uses Y, Cb and Cr, which are luma component, Chrominance of blue and Chrominance of red, respectively.

Using different colour channels has not been provided in performing the SR methods and evaluation result in the early deep-learning research. Therefore, most earlier deep learning models used the YCbCr channel, which had convenience on Y's luminance component. However, recent models and more traditional methods use the RGB scale, which is the absolute value for each pixel as the RGB scale showed better performance than YCbCr. The image with RGB channel trained with YCbCr and with RGB shows about 4dB [65].

Therefore, this project will use RGB space for all deep-learning pre-trained models and mathematical interpolation methods.

C. Benchmark flow design

The flow of the benchmark is designed as illustrated in Figure IV-6. The ground-truth (original) image is down-sampled with a down-sampler and saved on separate storage. The down-sampling process is done for four scaling factors: 2, 3, 4 and 8. The down-sampled images are then up-sampled with up-sampling methods explained in Section II. The up-sampled images are also stored on another storage. In the final stage, the up-sampled images are compared with the ground-truth original image using IQA metrics. The IQA metrics are also stored in CSV format for post-processing, such as graph plotting, statistical analysis.

D. Benchmark availability by scaling factor

As illustrated on section II, some methods and models are built for up-sampling to a specific scale only. Some of the models/methods are not available for some scale factor on the benchmark. The list of availability is written in Table IV-1. Therefore, a scaling factor of 3 and 8 can be unfair compared to other methods, but it is essential to find the best alternative method. Other available methods are benchmarked.

Table IV-1: Up-sampling methods/models availability by scaling factor

Method\Scale	x2	x3	x4	x8
Nearest	O	O	O	O
Bilinear	O	O	O	O
Bicubic	O	O	O	O
Lanczos	O	O	O	O
iNEDI	O		O	O
ICBI	O		O	O
DRCN	O	O	O	
EDSR	O	O	O	
ESPCN	O	O	O	
FSRCNN	O	O	O	
VDSR_NN	O	O	O	O
VDSR_Bilinear	O	O	O	O
VDSR_Bicubic	O	O	O	O
VDSR_Lanczos	O	O	O	O
LapSRN	O		O	

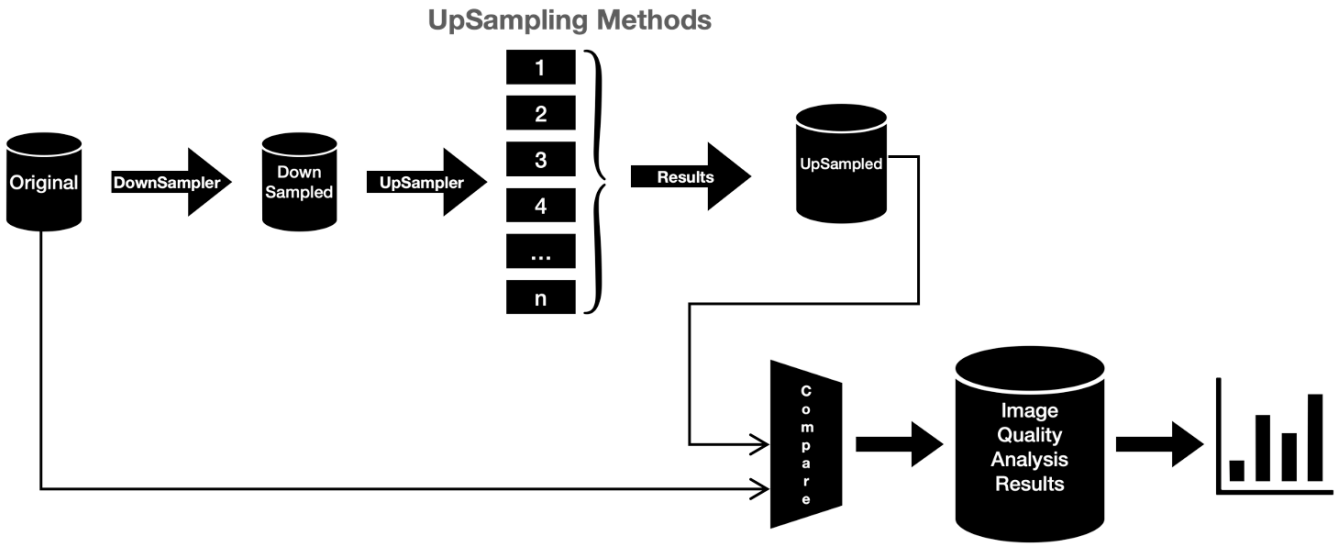

Figure IV-6: Benchmark process diagram

E. Benchmark implementation (codes)

The benchmark implementation with Python is uploaded on https://github.com/ArtemisDicoTiar/sisr_project. The general understanding of the implemented codes is provided by a short pseudo-code written in the following sections. The pseudo-code format follows Python syntax in general cases.

Down-sampling process

The down-sampler is built by taking some pixels from the original image sparsely. The pseudo-code of the down-sampler is implemented on Pseudo Code 1. The resulting matrix is then converted to an image using PNG for file extension and saved on separate storage. When loading and saving the image 'OpenCV2' library is used to process. (% operator results remainder of dividing two numbers before and after.)

Pseudo Code 1: Pseudo code of Down-sampler implementation

```
START
FOR y_index IN image's rows:
  FOR x_index IN image's columns:
    IF x_index % scaling_factor == 0 AND y_index %
scaling_factor == 0:
      APPEND the pixel value to target down-
sampled matrix

RETURN down-sampled matrix
END
```

Up-sampling process

The up-sampling process is mainly done with pre-built code and libraries that support graphical processing. Therefore, the process itself does not include complex processes. However, the pre-built codes distributed by each model proposers are developed on their computing environments. The resulting image exporting and target image importing processes of the pre-built codes are slightly modified to fit general computing environments. The pre-built codes written in MATLAB is run on 'MATLAB engine' in Python code, and the deep-learning codes are run on Tensorflow or PyTorch session. Finally, the comprehensive pseudo-code for the up-sampling process is written below (Pseudo Code 2).

Pseudo Code 2: Pseudo code for up-sampling process

```
START
FOR scaling_factor IN [2, 3, 4, 8]:
  FOR image_set IN [Set5, Set14, Urban100, Text]:
    FOR target_image IN image_set:
      FOR upSamplingMethod IN upSamplingMethods:
        img = upSamplingMethod(target_image)
        SAVE img ON storage name: image_set
END
```

IQA metrics outputting / result graph plotting

The up-sampled images are then applied to IQA metrics. The process is used multiple times for every image, so the code itself is implemented in object-oriented programming using class. The IQA metrics can be implemented by directly using matrix calculation. However, unlike

'RMSE' and 'PSNR', which can be easily implemented with simple matrix calculation that directly compares pixel value on the same coordinate, 'FSIM' and 'SSIM' use more complex matrix calculation more nearby pixels. Therefore, the IQA metrics are also implemented with the libraries that efficiently and effectively use the computing resources. The pseudo-code of extracting IQA metric results is written on

Pseudo Code 3. The code for the graph plot can be differed by changing x-axis, y-axis settings. For this reason, the pseudo-code for the graph plot is not provided in this paper.

Pseudo Code 3: Pseudo code of IQA metrics process

```
START
FOR scaling_factor IN [2, 3, 4, 8]:
  FOR image_set IN [Set5, Set14, Urban100, Text]:
    FOR target_image IN image_set:
      original_img=load(from=Grountruth image
storage)
      upSampled_img=load(from=upSampled image
storage)

      FOR IQA_metric IN [RMSE, PSNR, FSIM, SSIM]:
        ON result_table
        APPEND{
          IQA_metric(original_img, upSampled_img)}
END
```

F. Graphical User Interface (GUI) Implementation

The graphical user interface is also developed besides the benchmark code implementation. The GUI is usually built with three codes and files for specific purposes: Front-end design, Front-end code, and Backend Code. The front-end design is the graphical layout that users see and interact with it. The front-end code is the functionality of the viewable window. The code includes the react of user's action such as button click and dropdown selection. Lastly, the backend code is the key function of the program. The backend code's action is invisible to the user directly as they run behind the viewing window (this is why the code is called as 'backend). For this project, the program's primary function is to up-sample the target image or experiment with the up-sampling methods with an image and outputs both IQA metrics and up-sampled images. Therefore, the program's main functions, which already implemented benchmark code implementation, are built on the backend.

Front-end User Interface (UI) design

The front-end UI is developed with 'QtDesigner', an official design tool for 'PyQt5'. The front-end viewport is set as three windows. The windows are the SISR program setting window, processing bar window and result window.

The primary window illustrated in Figure IV-7 shows a selection button for which up-sampling methods to be used for processing, a dropdown box for scaling factor selection and another dropdown box that selects program mode, either experimental or production. The image can be selected by pressing the '…' button in the middle. The button will lead to another window that browses the user's local directory. By pressing 'OK' on the bottom closes the primary window and opens the progress bar window.

The following window shows the current processing model/method and how much the program has processed as textbox and progress bar on Figure IV-8. This window closes after all methods selected on the primary setting window. When all methods are processed, the final window is then visible.

In Figure IV-9, the final window shows an up-sampled image with the target input image. The up-sampled images can be saved by pressing the 'Save As' button. Also, the IQA metrics are visible as a bar graph on the right side of the window. The graph can be saved and can save the data as CSV. The order of up-sampled image and IQA metric graphs can be sort by selecting desired IQA metric on the dropdown box next 'Sort by:'. All temporary data and result images are deleted; therefore, the warning window appears.

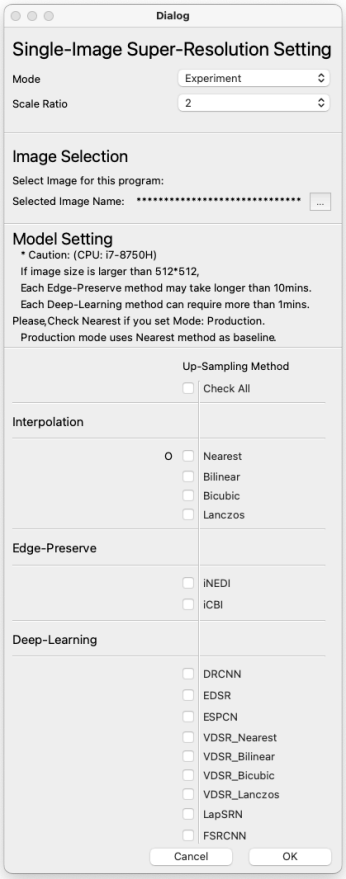

Figure IV-7: Front-end first view for setting SR experiment and production

Front-end code implementation

The parts that require actions on the front-end are buttons, dropdown box, checkbox, and window transition.

The checkbox and dropdown box actions are already provided by the 'PyQt5' application programming interface (API). Therefore, the only appropriate functions should be applied to the 'stateChanged' API function. The applied function changes the variable value and stores values in a separate class. The checkboxes and dropdown boxes on the first window use API's state class to update the variable value.

The button click is like the checkbox and dropdown box, so applying an appropriate function to the 'OK', 'Cancel' and 'clicked' API function will make the buttons work. The cancel button is connected to the window closes, and the program ends from the system. The ok button will also close to progress to the next stage's window or finish the program if the current window is the last result window. Lastly, 'Save As', 'Export Graph' and 'Export Data' buttons open the system browser window to obtain the saving directory. When the saving directory is selected, the target data is saved on the directory.

Lastly, the window transition is done by checking the either 'OK' button is pressed or the all selected up-sampling process is done. The window transition is processed by closing the current window and opening the next window in the same computing process. Values from each window are stored on the class variable. The class is initialised at the beginning of the front-end program. By saving the value on the common class, the value can be used by other stage windows. This value storing process is done on the first window: program setting.

The code can be found on the Github repository, and the pseudo-code is not provided as its functionality does not require a complicated process, and the length of code is long to bind UI design and backend.

Backend code implementation

The backend code is the main functionality of this program. All up-sampling methods and models are run by backend code. Based on the program setting value obtained from the first window, the up-sampling methods are run while the second window is opened.

The backend codes are mainly built following the codes built for the benchmark. Therefore, the pseudo-code on Pseudo Code 4 shows a similar structure compare to the benchmark's pseudo-codes. The up-sampled images and IQA metrics are saved on a temporary directory as front-end loads from the file to make the data visible.

Pseudo Code 4: Pseudo code for backend of GUI

```
START

FOR selectedMethod IN selectedMethods:
    SAVE selectedMethod(dataClass.targetImage)
    ON temporary directory
    AS str(selectedMethod + original image
name).png

FOR iqaMetric IN iqaMetrics:
    APPEND iqaMetric(upSampledImage)
    ON iqaMetricsTable

SAVE iqaMetricsTable
ON temporary directory
AS iqaMetricTable.csv

END
```

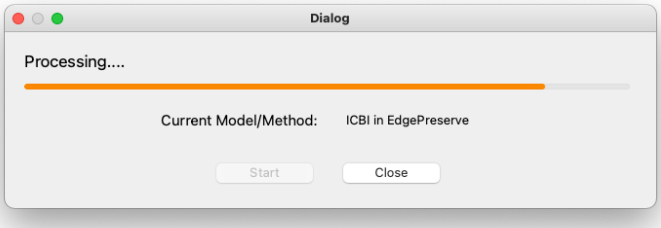

Figure IV-8: Front-end second window showing processing status

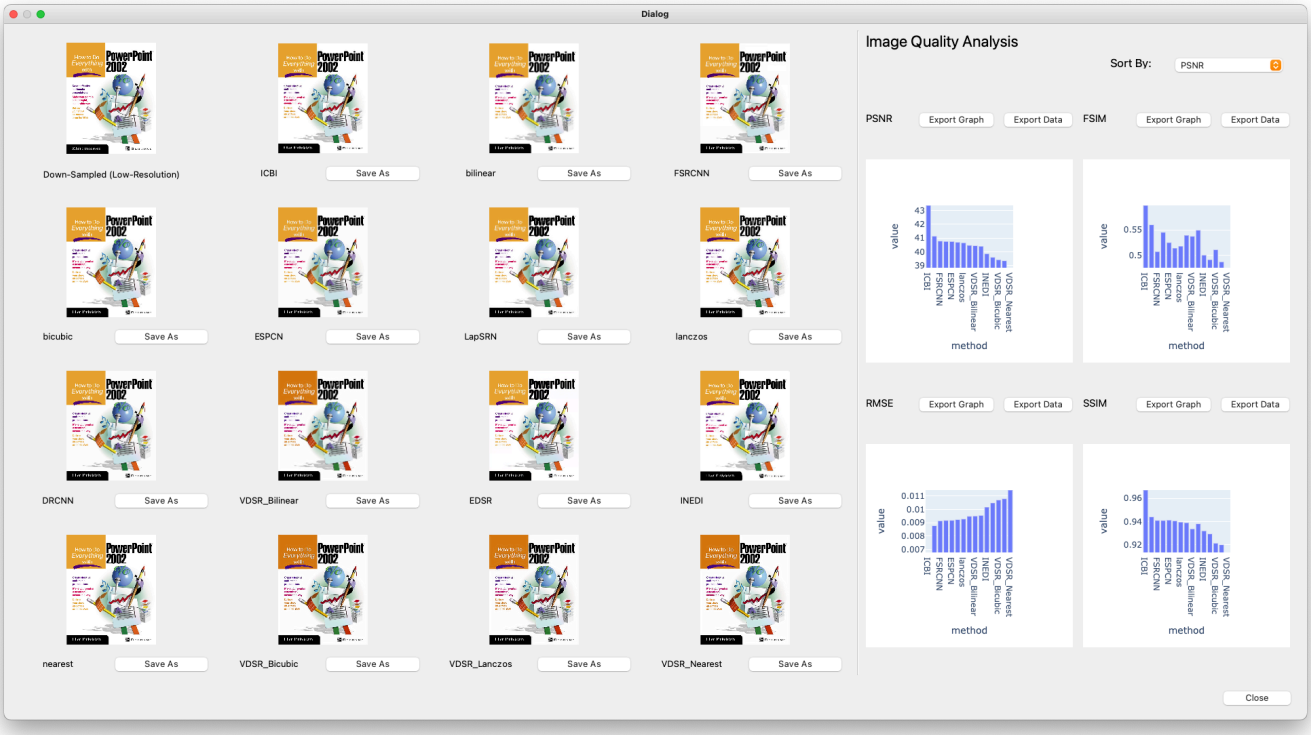

Figure IV-9: Front-end final window with up-sampled images and IQA metrics

V. Experiment Results

The benchmark results are massive to illustrate on one table. Therefore, the sections are divided by image sets and the tables are separated by the scale factor. For the laconism of the content, all tables with the details are described in Appendix. The up-sampled, down-sampled and ground truth image can be found on the 'Github' repository written on the footnote of the first page.

Furthermore, the IQA result with whole decimals are challenging to fit into this paper, so the paper uses IQA metrics rounded at the 4th decimal point. The full decimal point results are also available on the 'Github' repository and the implemented codes. The IQA result not available due to methods'/models' setting remains as blank on all tables. To provide better understanding about the result, example result images are plotted for every section.

General Scene Images: Set5

The 'Set5' image set contains general scene images. The IQA metrics of Set 5 image set show 'ICBI' is the dominating method. The IQA metrics that do not show 'ICBI' as the best method are the case where 'ICBI' is not available for the condition such as the benchmark setting with scaling factor of 3.

While analysing the result for scaling factor of 3, 'EDSR' shows the best performance for 'FSIM' while 'bilinear interpolation' shows the

best performance for ‘PSNR’ and ‘RMSE’, ‘ESPCN’ shows the best for ‘SSIM’.

The cropped result images for ‘butterfly’ with a scaling factor of 2 on Figure V-1 shows that most up-sampling methods output smooth and easily recognisable border lines. The nearest neighbour only showed stair-like grid shape image pixels. This can also be found on IQA metrics that results for scaling factor of 2 have similar values.

The example result image for ‘butterfly’ with a scaling factor of 4 on Figure V-2 shows that ‘EDSR’ and ‘LapSRN’ have clear and sharp butterfly’s wing pattern. This is smoother with the zoomed image in Figure V-3. However, the IQA metrics result that ‘ICBI’ is showing the best performance.

Furthermore, the cropped image for ‘butterfly’ with a scaling factor of 8 is illustrated in Figure V-4. As the experiment condition requires to up-sample from extremely low-resolution, most of the images show barely recognisable butterfly’s wing pattern. The ‘ICBI’ even showed artefacts of red dots on the image.

Overall, objective IQA metrics result that the ICBI is the best performing method for the Set 5 dataset. However, the ICBI method’s result images show some artefacts and the hardly identifiable edge of the object.

Table V-1: IQA metrics with the average for Set 5 image dataset. (The best performance method for each scaling factor shows bold and underlined IQA metric value.)

iqa	factor	DRCNN	EDSR	ESPCN	FSRCNN	ICBI	INEDI	LapSRN	VDSR_ bicubic	VDSR_ bilinear	VDSR_ lanczos	VDSR_ nearest	bicubic	bilinear	lanczos	nearest
FSIM	2	0.701	0.747	0.743	0.738	**0.753**	0.694	0.736	0.689	0.716	0.664	0.651	0.736	0.743	0.726	0.736
	3	0.631	**0.645**	0.64	0.637				0.596	0.624	0.565	0.532	0.637	0.643	0.629	0.54
	4	0.57	0.574	0.572	0.566	**0.641**	0.534	0.571	0.544	0.565	0.535	0.455	0.571	0.576	0.567	0.441
	8					**0.51**	0.424		0.442	0.441	0.437	0.359	0.447	0.448	0.442	0.309
PSNR	2	50.084	52.366	52.415	52.173	**54.638**	50.501	52.248	47.184	48.143	46.681	46.068	52.174	52.325	52.054	50.379
	3	47.856	48.342	48.466	48.385				44.877	46.11	44.18	44.3	48.262	**48.551**	48.109	46.824
	4	46.174	46.223	46.486	46.351	**49.623**	44.886	46.38	43.98	45.068	43.836	42.883	46.197	46.675	46.043	44.917
	8					**45.183**	41.375		41.87	42.294	41.737	40.721	42.424	42.964	42.263	41.373
RMSE	2	0.003	0.003	0.003	0.003	**0.002**	0.003	0.003	0.005	0.004	0.005	0.005	0.003	0.003	0.003	0.003
	3	0.004	0.004	0.004	0.004				0.006	0.005	0.007	0.006	0.004	**0.004**	0.004	0.005
	4	0.005	0.005	0.005	0.005	**0.004**	0.006	0.005	0.007	0.006	0.007	0.008	0.005	0.005	0.005	0.006
	8					**0.006**	0.009		0.009	0.008	0.009	0.01	0.008	0.008	0.008	0.009
SSIM	2	0.988	0.993	0.993	0.993	**0.996**	0.988	0.993	0.979	0.983	0.977	0.974	0.993	0.993	0.993	0.989
	3	0.98	0.983	**0.983**	0.982				0.963	0.972	0.957	0.959	0.982	0.983	0.981	0.975
	4	0.971	0.971	0.973	0.972	**0.986**	0.957	0.972	0.953	0.963	0.951	0.943	0.971	0.973	0.97	0.962
	8					**0.962**	0.917		0.925	0.933	0.923	0.909	0.935	0.941	0.932	0.921

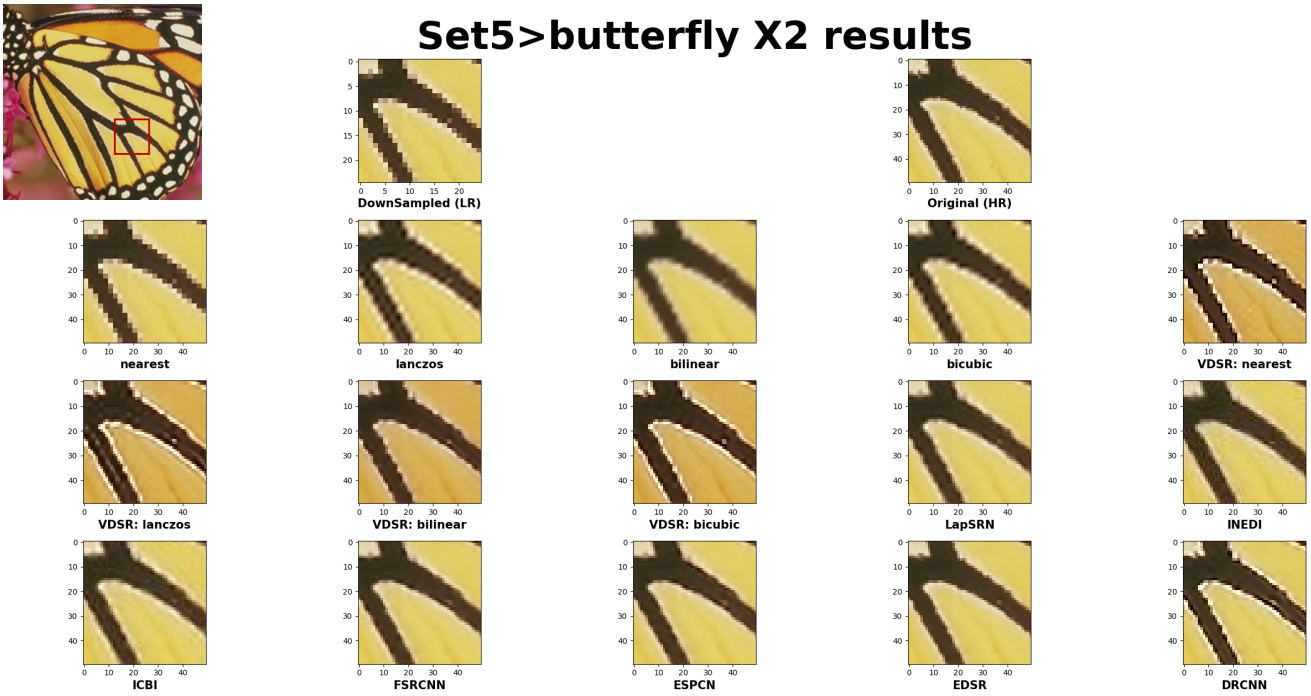

Figure V-1: Set5, butterfly image with a scaling factor of 2, zoomed image.

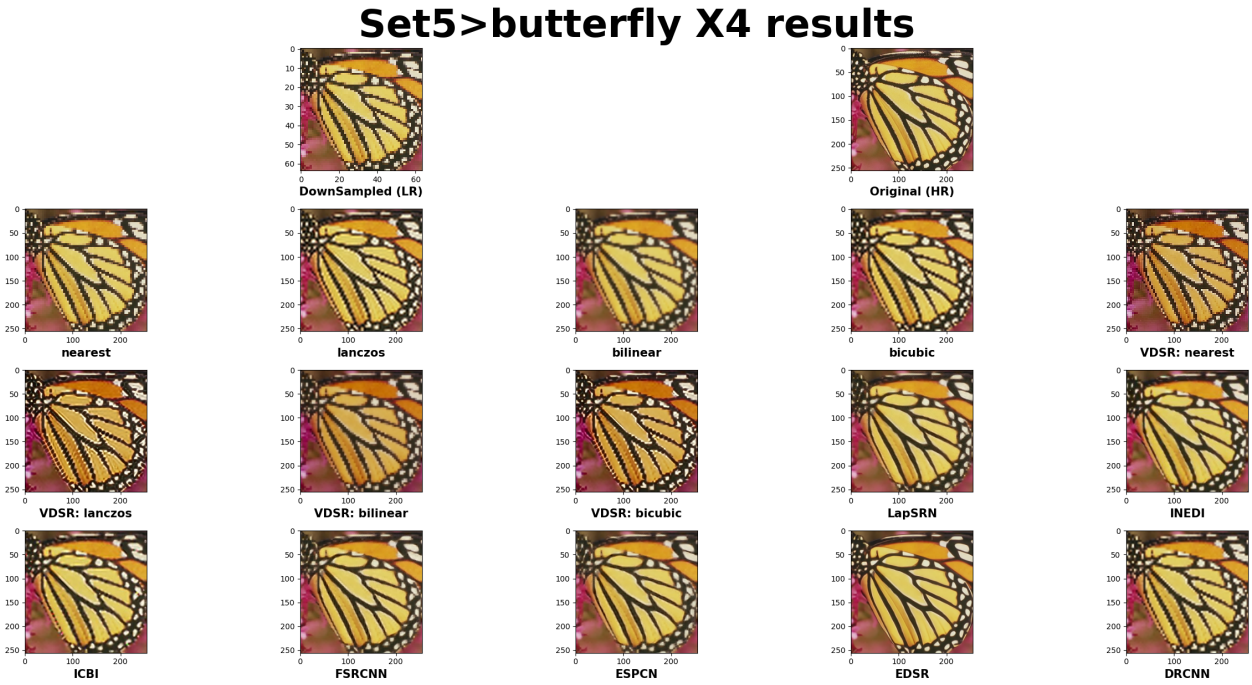

Figure V-2: Set5, butterfly image with a scaling factor of 4, overall result

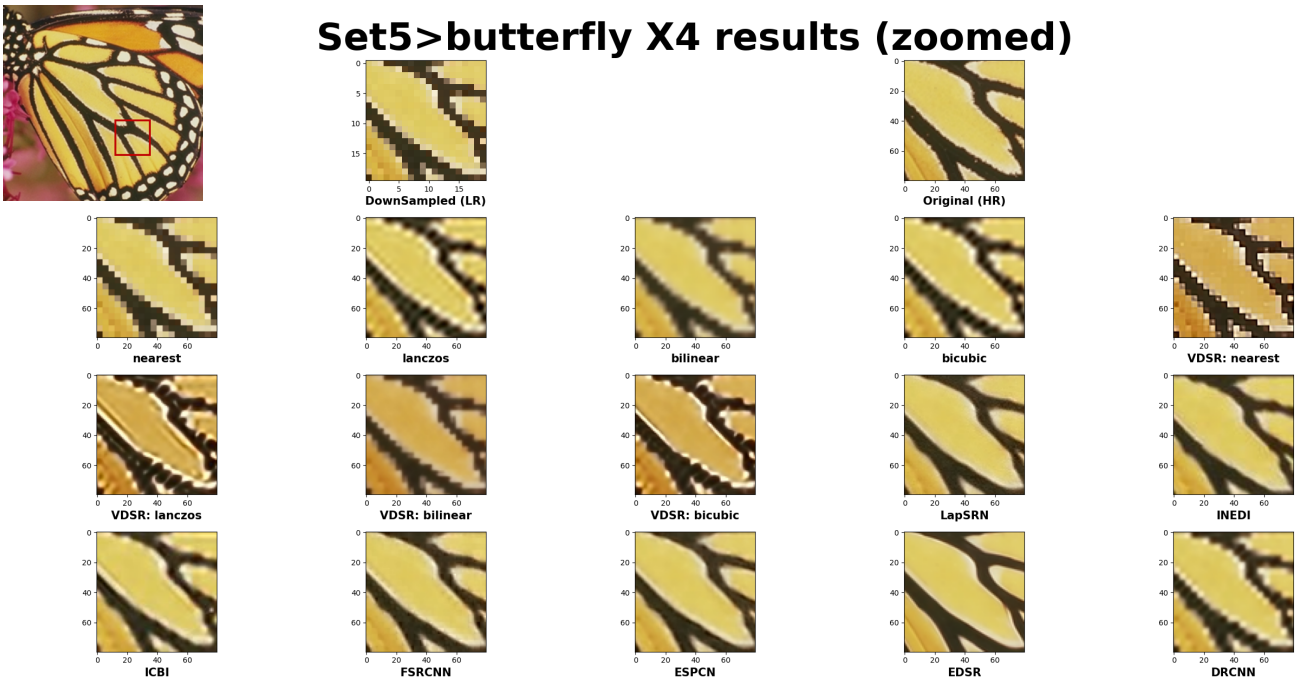

Figure V-3: Set5, butterfly image with a scaling factor of 4, zoomed image.

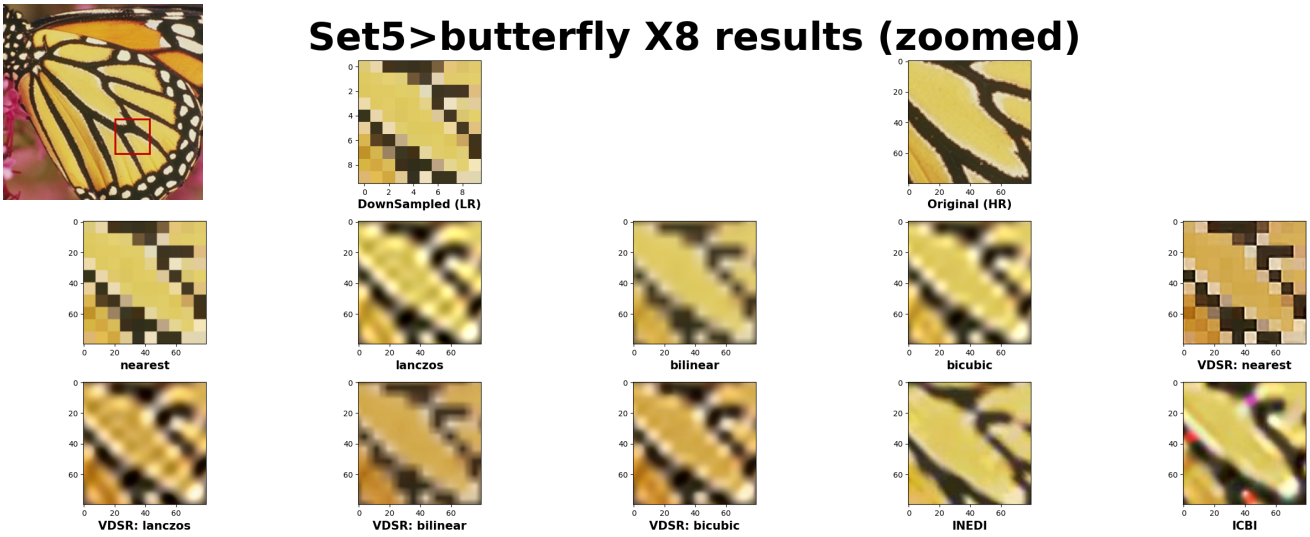

Figure V-4: Set5, butterfly image with a scaling factor of 8, zoomed image.

General Scene Images: Set14

The IQA metrics of Set14 image set also show that 'ICBI' is the dominating method. The experiment condition with scaling factors of 3 and 8 are not available to some method described in Table IV-1. For a scaling factor of 3, 'bicubic interpolation' showed the best performance for 'FSIM' and 'bilinear interpolation' for other IQA metrics.

The cropped up-sampled images of 'zebra' for scaling factor of 2 on Figure V-5 show a transparent black and white pattern. Some methods, including 'bilinear interpolation' and 'bicubic interpolation', show smoothness between white and black, making it challenging to recognise the border. Other methods show clear boundary but 'VDSR with Lanczos pre-interpolation' shows some artefact nearby the boundary.

The same image with a scaling factor of 4 is illustrated in Figure V-6 and cropped image to

analyse boundary lines is illustrated in Figure V-7. From Figure V-6, there are some stair-like pixels on 'VDSR' up-sampled images. Other methods show a clear zebra's pattern, smooth and sharp. However, due to distorted information stored on the down-sampled image, the pattern on the zebra's ankle is flatten as grey (combination of black and white). It is having a pattern but distorted, so its pattern does not match with the original image. The zoomed image on Figure V-7 with the same region as Figure V-5 shows that basic interpolation methods, 'VDSR' and 'DRCN', shows bad boundary line either stair-like pixels or artefacts near the boundary line. The sharpest and clear boundary line can be seen on 'EDSR', but it is not the best performing method on IQA metrics. The 'ICBI' is the dominant method for scaling by 4, but it shows the black artificial line on the white region.

Lastly, the image process with a down-sampled by scaling factor of 8 is illustrated in Figure V-8. Basic interpolation methods and 'VDSR' shows stair-like block pixels and the artificial line next to the boundary. The 'iNEDI' and 'ICBI' show clear boundary, but the methods also show red and yellow dots on the white region.

Table V-2: IQA metrics with the average for Set 14 image dataset. (The best performance method for each scaling factor shows bold and underlined IQA metric value.)

iqa	factor	DRCNN	EDSR	ESPCN	FSRCNN	ICBI	INEDI	LapSRN	VDSR_ bicubic	VDSR_ bilinear	VDSR_ lanczos	VDSR_ nearest	bicubic	bilinear	lanczos	nearest
FSIM	2	0.66	0.714	0.71	0.704	**0.726**	0.659	0.704	0.658	0.688	0.627	0.613	0.705	0.708	0.69	0.707
	3	0.59	0.595	0.589	0.588				0.563	0.585	0.541	0.509	**0.597**	0.597	0.591	0.514
	4	0.527	0.522	0.517	0.513	**0.582**	0.486	0.516	0.514	0.522	0.509	0.447	0.529	0.528	0.528	0.428
	8					**0.458**	0.392		0.404	0.403	0.399	0.349	0.406	0.406	0.401	0.299
PSNR	2	47.07	49.445	49.565	49.329	**51.117**	47.983	49.529	45.645	47.029	45.134	44.229	49.413	49.693	49.231	48.019
	3	45.775	46.219	46.443	46.382				43.287	44.916	42.731	43.023	46.208	**46.536**	46.049	45.047
	4	44.54	44.563	44.891	44.822	**46.84**	43.525	44.84	42.486	44.068	42.417	41.898	44.601	45.126	44.432	43.524
	8					**43.634**	40.685		41.107	41.602	40.975	40.079	41.568	42.115	41.418	40.654
RMSE	2	0.005	0.004	0.003	0.004	**0.003**	0.004	0.004	0.005	0.005	0.006	0.006	0.004	0.003	0.004	0.004
	3	0.005	0.005	0.005	0.005				0.007	0.006	0.008	0.007	0.005	**0.005**	0.005	0.006
	4	0.006	0.006	0.006	0.006	**0.005**	0.007	0.006	0.008	0.006	0.008	0.008	0.006	0.006	0.006	0.007
	8					**0.007**	0.01		0.009	0.009	0.009	0.01	0.009	0.008	0.009	0.01
SSIM	2	0.978	0.987	0.988	0.987	**0.99**	0.982	0.987	0.97	0.977	0.967	0.96	0.987	0.988	0.986	0.983
	3	0.972	0.974	0.975	0.975				0.951	0.964	0.945	0.949	0.974	**0.975**	0.973	0.967
	4	0.963	0.963	0.966	0.965	**0.976**	0.954	0.965	0.941	0.958	0.94	0.935	0.963	0.967	0.962	0.955
	8					**0.955**	0.923		0.926	0.934	0.924	0.91	0.936	0.942	0.933	0.923

Figure V-5: Set14, zebra image with a scaling factor of 2, zoomed image.

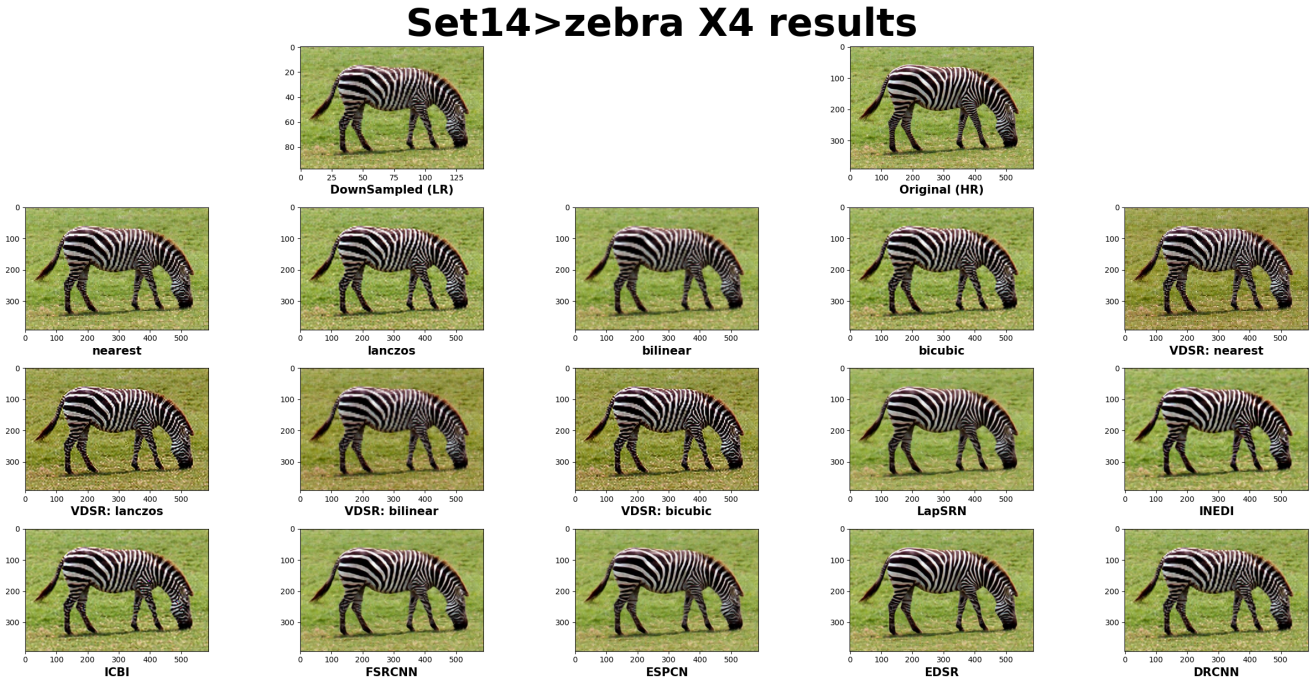

Figure V-6: Set14, zebra image with a scaling factor of 4, overall result

Figure V-7: Set14, zebra image with a scaling factor of 4, zoomed image.

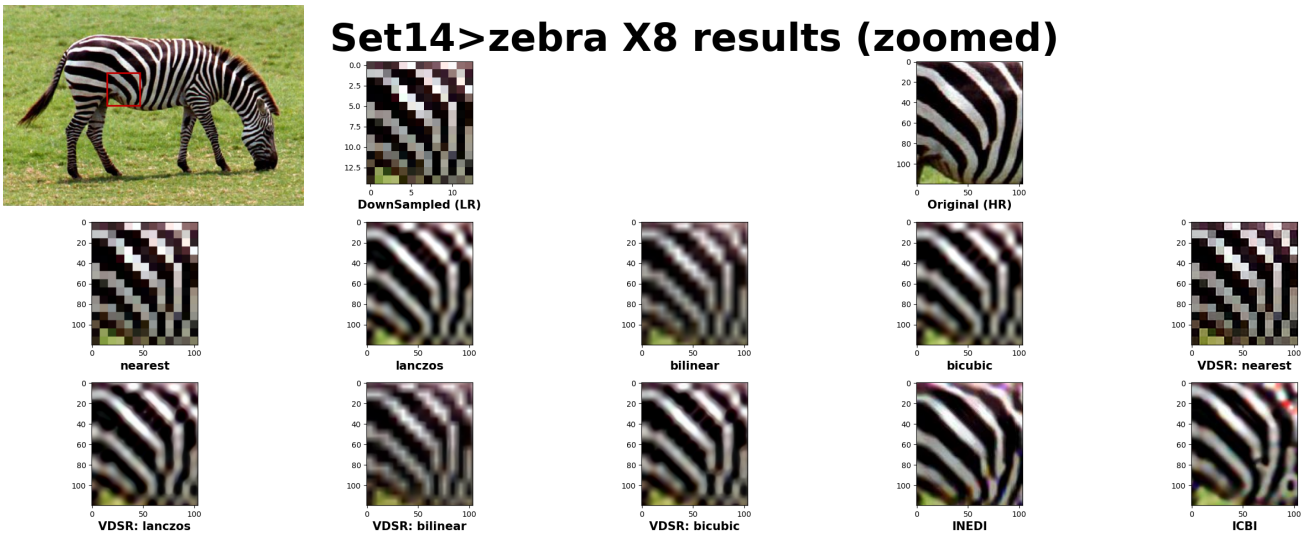

Figure V-8: Set14, zebra image with a scaling factor of 8, zoomed image.

Pattern Images: Urban100

The IQA metrics for partial Urban100 image set (first ten images) on Table V-3 shows that 'ICBI' is the dominating method. For the scaling factor of 3, where 'ICBI' is not available for the condition, 'EDSR' is the best method with 'FSIM' and 'SSIM', whereas 'bilinear interpolation' shows the best performance for 'PSNR' and 'RMSE'.

The up-sampled results with a scaling factor of 2 on Figure V-9 show mostly clear columns while thin wires are up-sampled with stair-like pixels. The 'EDSR' showed two recognisable wires, whereas other methods result in either almost one wire or two wires with aliasing.

For the result from the scaling factor of 4, all the methods show region divided by thick columns with clear borderline. However, the window region shows some artefacts and aliasing due to two thin wires. The window region on 'EDSR' showed sharp and smooth column and wires except for few window regions. The down-sampled image may cause the window region

with aliasing lost information of the wire. The basic interpolation methods and 'VDSR' shows wiggly wires in Figure V-11. Even more, the basic interpolation methods, 'VDSR', 'ICBI' and 'DRCN(N)' failed to up-sampled the thick columns.

Lastly, the up-sampled results with a scaling factor of 8 are illustrated in Figure V-12. All the methods failed to up-sample the pixels for not only thin wires but also the thick columns. The up-sampled images all show wiggly and stair-like pixels with aliasing. Also, the 'ICBI' shows blue and red dots artefacts.

Table V-3: IQA metrics with average for part of Urban100 image dataset. (The best performance method for each scaling factor shows bold and underlined IQA metric value.)

iqa	factor	DRCNN	EDSR	ESPCN	FSRCNN	ICBI	INEDI	LapSRN	VDSR_ bicubic	VDSR_ bilinear	VDSR_ lanczos	VDSR_ nearest	bicubic	bilinear	lanczos	nearest
FSIM	2	0.634	0.704	0.696	0.688	**0.707**	0.65	0.69	0.639	0.671	0.614	0.599	0.685	0.693	0.671	0.686
	3	0.566	**0.584**	0.575	0.571				0.538	0.567	0.513	0.494	0.571	0.574	0.563	0.489
	4	0.505	0.509	0.505	0.501	**0.566**	0.475	0.507	0.489	0.506	0.482	0.433	0.506	0.507	0.504	0.399
	8					**0.449**	0.385		0.397	0.398	0.394	0.348	0.399	0.4	0.396	0.286
PSNR	2	44.483	47.304	47.559	47.323	**49.012**	46.047	47.493	44.48	46.112	44.067	43.05	47.399	47.728	47.176	46.03
	3	43.811	44.0	44.572	44.45				41.958	43.855	41.367	41.824	44.274	**44.634**	44.103	43.169
	4	42.81	42.548	43.205	43.063	**44.683**	41.823	43.137	41.13	43.023	41.06	40.735	42.855	43.425	42.679	41.846
	8					**41.945**	39.572		40.218	40.86	40.049	39.139	40.381	41.009	40.205	39.488
RMSE	2	0.006	0.005	0.004	0.005	**0.004**	0.005	0.004	0.006	0.005	0.007	0.007	0.005	0.004	0.005	0.005
	3	0.007	0.007	0.006	0.006				0.008	0.007	0.009	0.008	0.006	**0.006**	0.007	0.007
	4	0.008	0.008	0.007	0.007	**0.006**	0.009	0.007	0.009	0.007	0.009	0.01	0.008	0.007	0.008	0.008
	8					**0.008**	0.011		0.01	0.009	0.01	0.012	0.01	0.009	0.01	0.011
SSIM	2	0.964	0.98	0.981	0.98	**0.985**	0.972	0.981	0.965	0.975	0.962	0.953	0.98	0.981	0.979	0.974
	3	0.957	0.959	**0.963**	0.962				0.938	0.957	0.93	0.937	0.961	0.963	0.959	0.951
	4	0.946	0.944	0.95	0.949	**0.963**	0.928	0.949	0.925	0.948	0.923	0.919	0.946	0.952	0.945	0.935
	8					**0.933**	0.89		0.908	0.918	0.905	0.888	0.911	0.921	0.908	0.895

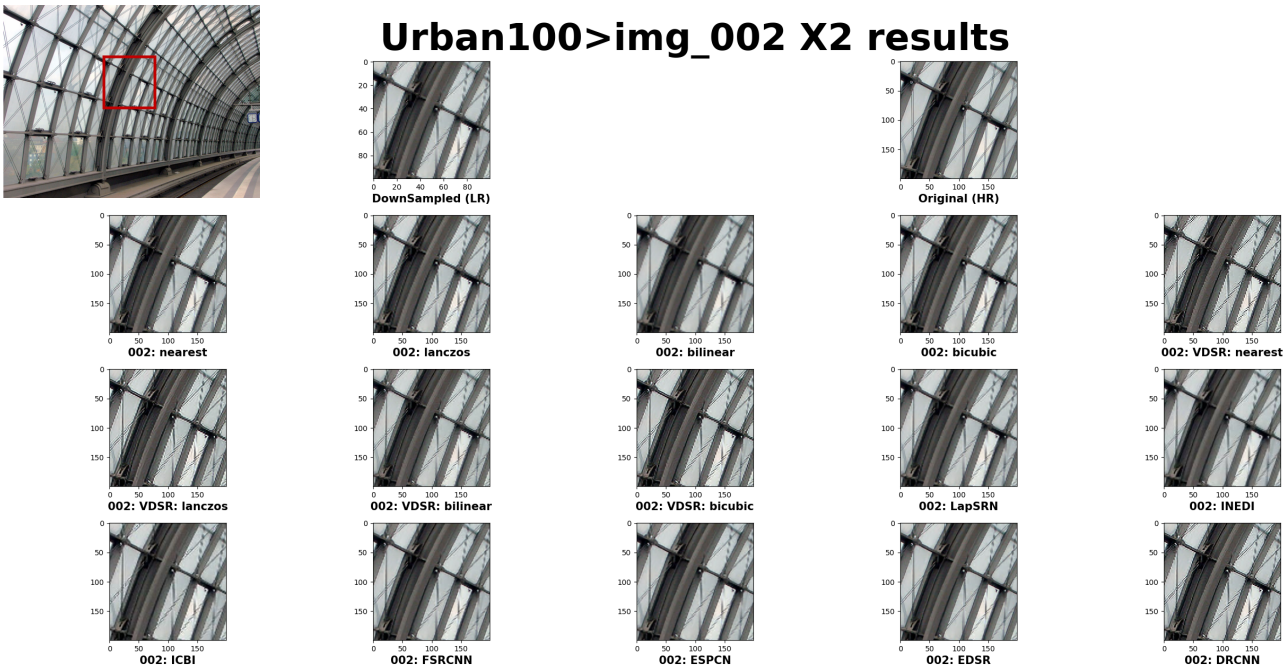

Figure V-9: Urban100, img_002 image with a scaling factor of 2, zoomed image.

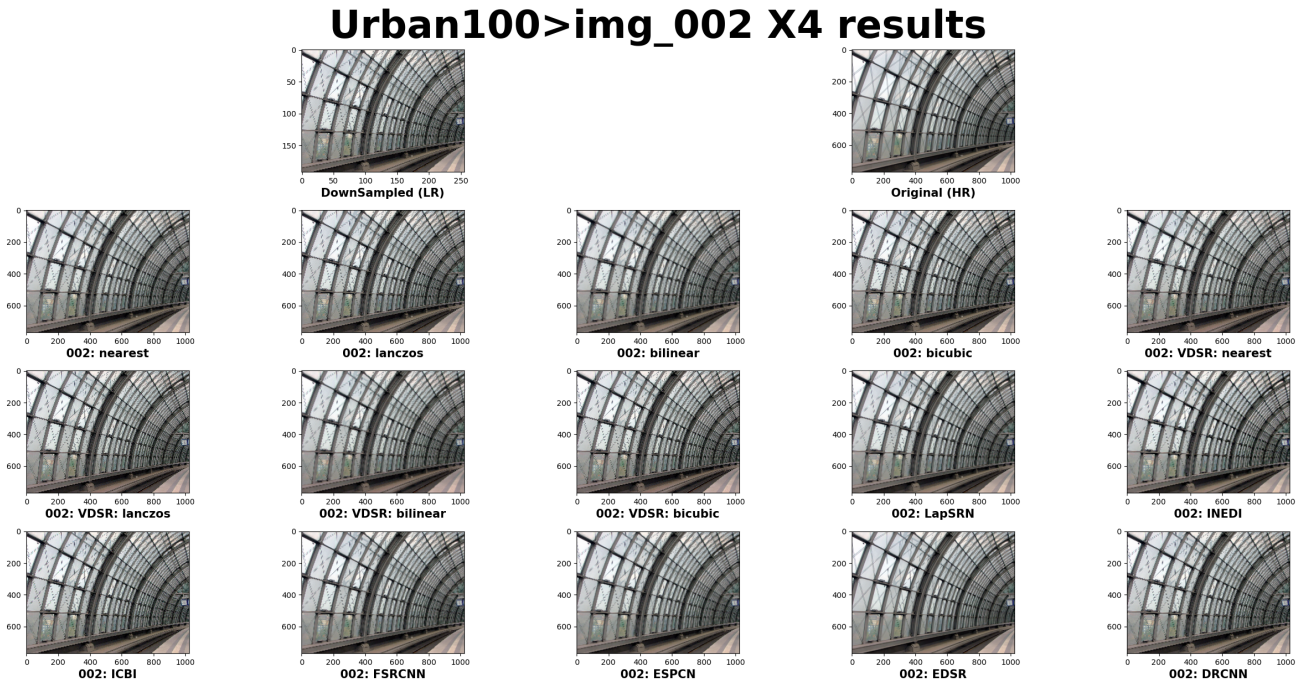

Figure V-10: Urban100, img_002 image with a scaling factor of 4, overall result

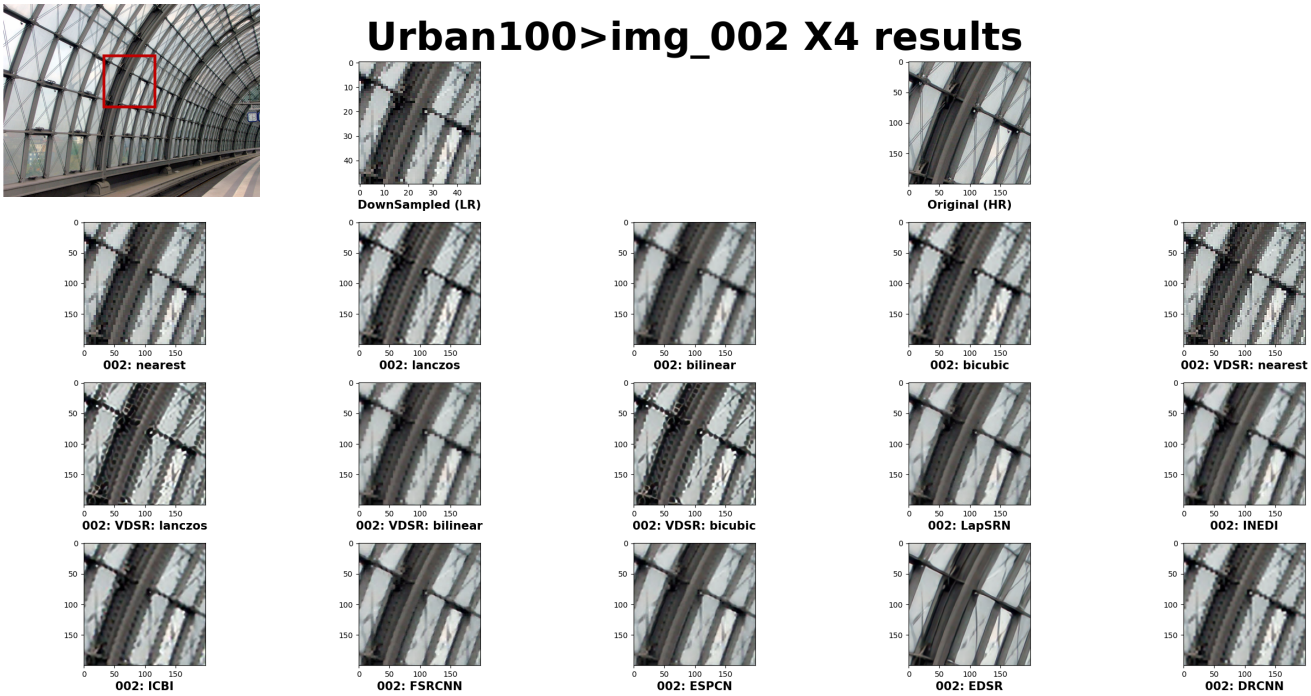

Figure V-11: Urban100, img_002 image with a scaling factor of 4, zoomed image.

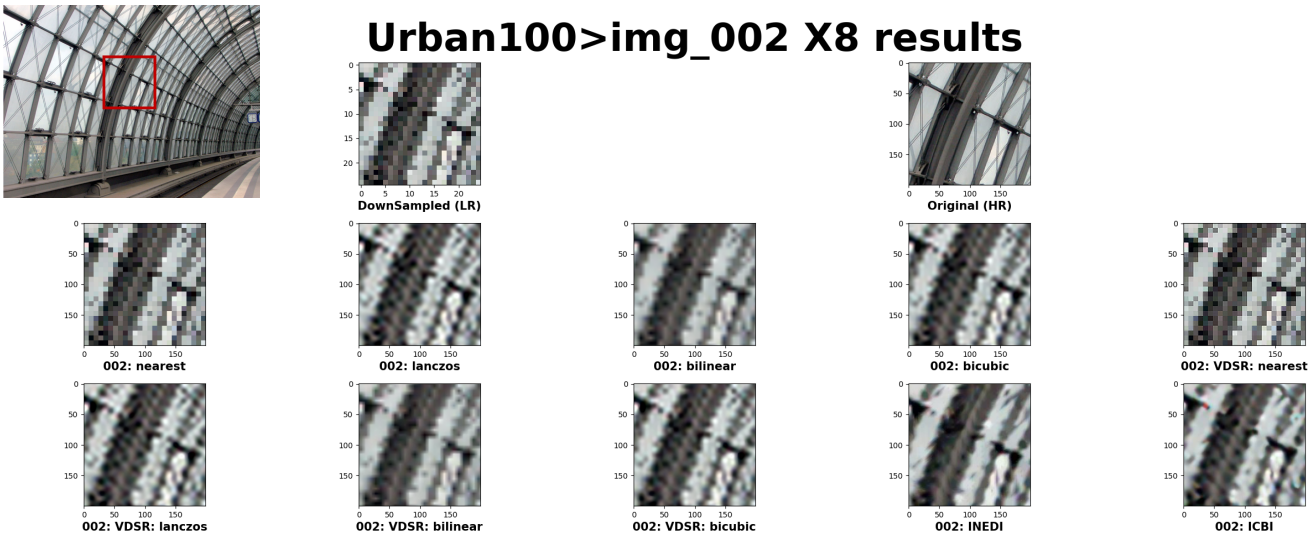

Figure V-12: Urban100, img_002 image with a scaling factor of 8, zoomed image.

Text Images: KAIST Scene Text Database tc-100

The IQA metrics of 'KAIST Scene Text Database tc-100 (following, tc-100)' also result that 'ICBI' is the best-performing methods. For the scaling factor of 3, where the condition that 'ICBI' cannot be processed, 'FSIM' shows that 'VDSR with bilinear pre-interpolation' is the best performing method. At the same time, other IQA metrics results that 'bilinear interpolation' having the best performance.

The IQA metrics of Set14 image set also show that 'ICBI' is the dominating method. The experiment condition with scaling factors of 3 and 8 are not available to some method described in Table IV-1. For a scaling factor of 3, 'bicubic interpolation' showed the best performance for 'FSIM' and 'bilinear interpolation' for other IQA metrics.

The result images with a scaling factor of 4 are illustrated in Figure V-13. All the methods show a clear boundary line of the text. There are some wiggling pixel values on the blue background between texts from 'VDSR' and 'DRCN(N)'.

The result images with a scaling factor of 4 are illustrated in Figure V-14. The texts are easily recognisable for all methods' result. However, the zoomed result on Figure V-15 shows that 'basic interpolation methods', 'VDSR' and 'DRCN(N)' results stair-like block pixels on the curve of the character. The other methods show the smooth curve of the character. The 'EDSR', 'LapSRN', 'iNEDI', 'FSRCNN' and 'ESPCN' show clear and smooth boundary line compare to other methods.

The zoomed images resulted with a scaling factor of 8 is plotted in Figure V-16. Because down-sampled image lost important information from the ground-truth image, most of the results show characters challenging to read. Furthermore, 'iNEDI' and 'ICBI' show yellow artificial pixels. The full-size image is significant, and human visual perception is processed with structural information. All result images with a scaling factor of 8 on Figure V-17 are readable. However, the characters are wiggly and squashed with nearby characters, so it is barely recognised.

Table V-4: IQA metrics with average for part of KAIST Scene Text Database tc-100 image dataset. (The best performance method for each scaling factor shows bold and underlined IQA metric value.)

iqa	factor	DRCNN	EDSR	ESPCN	FSRCNN	ICBI	INEDI	LapSRN	VDSR_ bicubic	VDSR_ bilinear	VDSR_ lanczos	VDSR_ nearest	bicubic	bilinear	lanczos	nearest
FSIM	2	0.594	0.617	0.62	0.617	**0.627**	0.596	0.616	0.602	0.622	0.585	0.584	0.608	0.611	0.604	0.6
	3	0.531	0.542	0.544	0.546				0.535	**0.549**	0.52	0.496	0.533	0.537	0.53	0.502
	4	0.483	0.5	0.495	0.497	**0.524**	0.462	0.499	0.487	0.502	0.483	0.431	0.485	0.491	0.484	0.443
	8					**0.443**	0.384		0.406	0.405	0.404	0.349	0.4	0.4	0.398	0.325
PSNR	2	54.978	56.456	56.449	56.294	**58.23**	54.838	56.477	49.311	49.977	48.654	48.721	56.277	56.465	56.204	54.997
	3	52.312	52.705	52.871	52.884				47.601	48.37	47.084	47.104	52.752	**52.955**	52.655	51.569
	4	50.629	50.645	50.867	50.794	**53.254**	49.023	50.784	46.774	47.589	46.744	46.081	50.673	51.051	50.548	49.653
	8					**48.901**	45.243		44.94	45.265	44.849	44.21	46.644	47.073	46.523	45.851
RMSE	2	0.002	0.002	0.002	0.002	**0.001**	0.002	0.002	0.004	0.003	0.004	0.004	0.002	0.002	0.002	0.002
	3	0.003	0.002	0.002	0.002				0.004	0.004	0.005	0.005	0.002	**0.002**	0.002	0.003
	4	0.003	0.003	0.003	0.003	**0.002**	0.004	0.003	0.005	0.004	0.005	0.005	0.003	0.003	0.003	0.003
	8					**0.004**	0.006		0.006	0.006	0.006	0.006	0.005	0.005	0.005	0.005
SSIM	2	0.996	0.997	0.997	0.997	**0.998**	0.996	0.997	0.978	0.979	0.977	0.977	0.997	0.997	0.997	0.996
	3	0.993	0.993	0.994	0.994				0.973	0.976	0.972	0.972	0.993	**0.994**	0.993	0.991
	4	0.989	0.99	0.99	0.99	**0.994**	0.986	0.99	0.97	0.973	0.97	0.967	0.99	0.99	0.989	0.987
	8					**0.984**	0.971		0.962	0.964	0.961	0.956	0.977	0.979	0.976	0.972

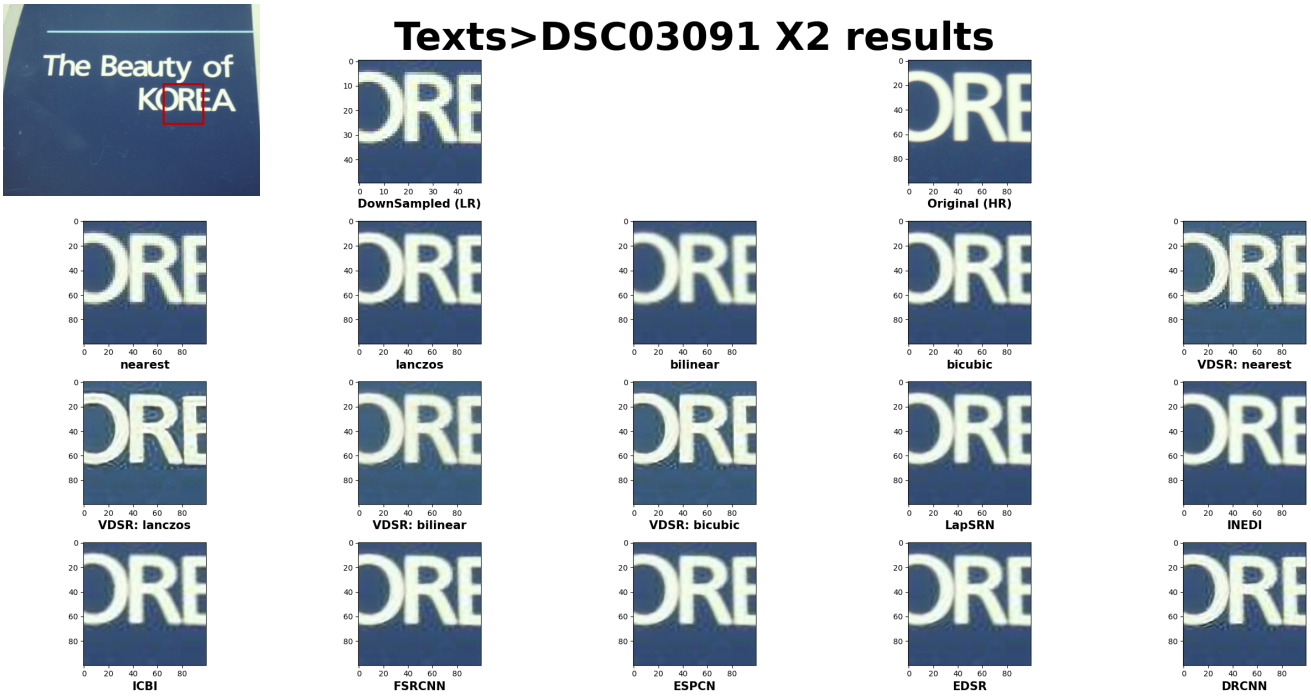

Figure V-13: tc-100, DSC3091 image with a scaling factor of 2, zoomed image.

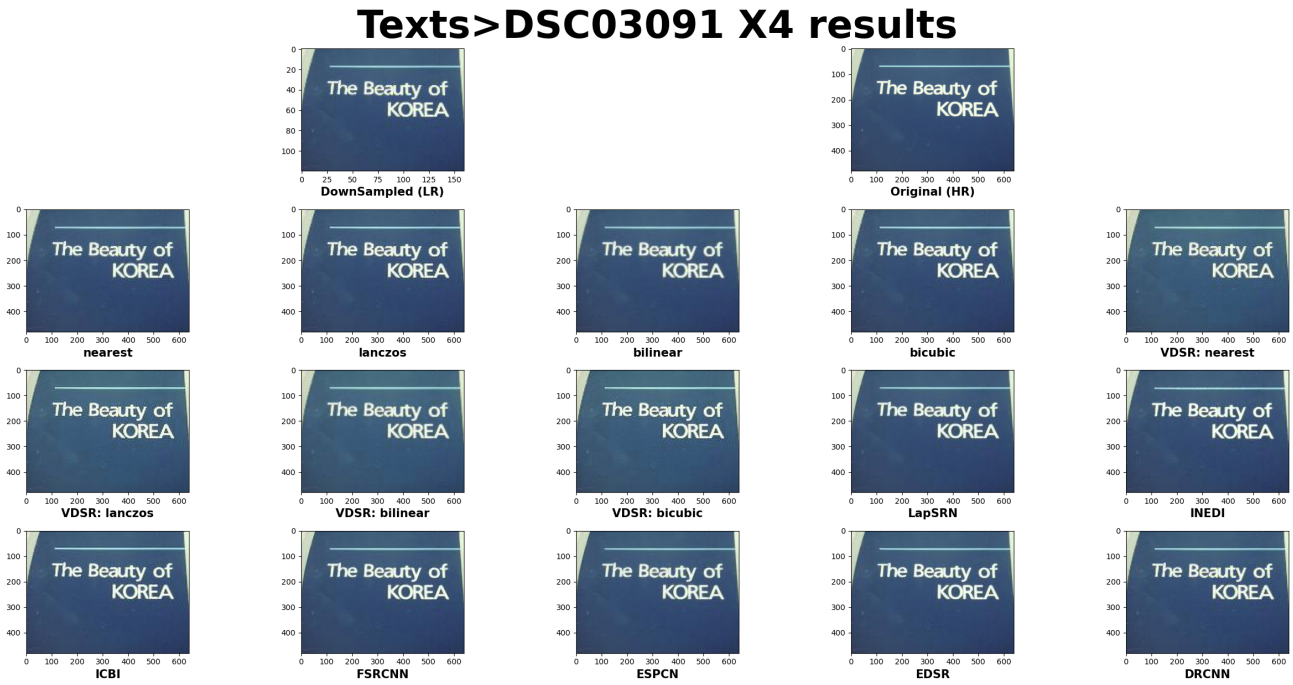

Figure V-14: tc-100, DSC3091 image with a scaling factor of 4, overall result

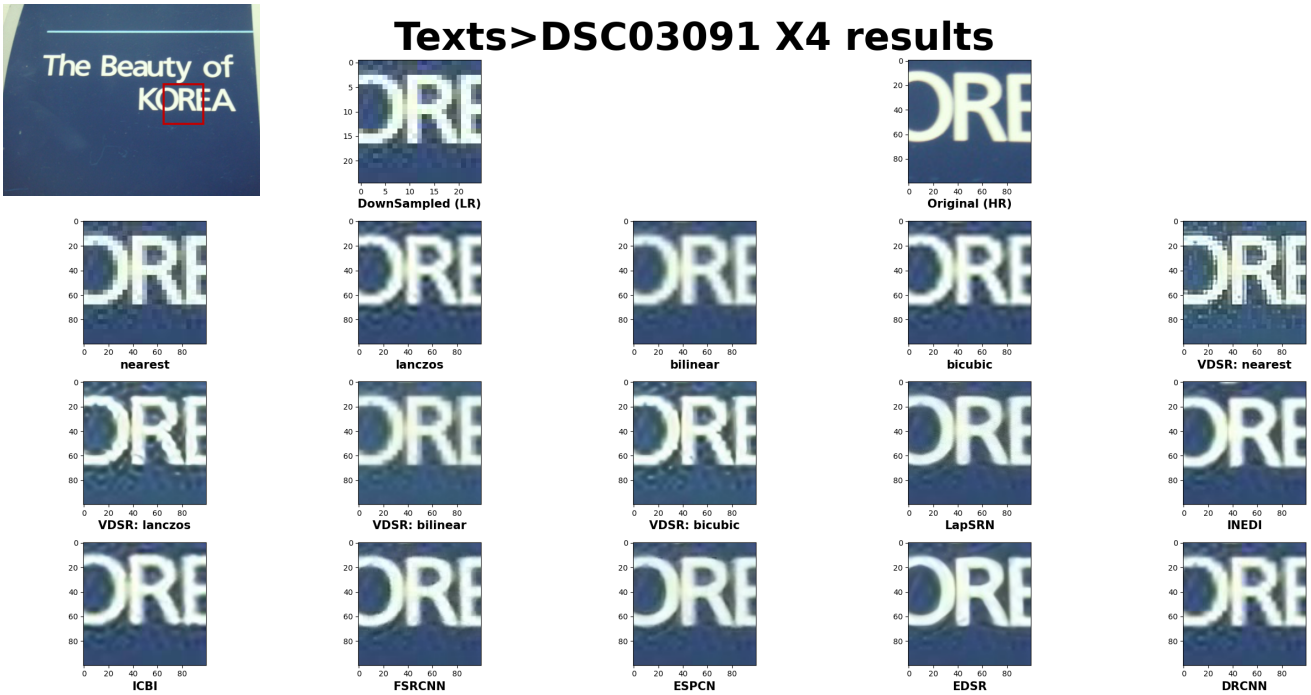

Figure V-15: tc-100, DSC3091 image with a scaling factor of 4, zoomed image.

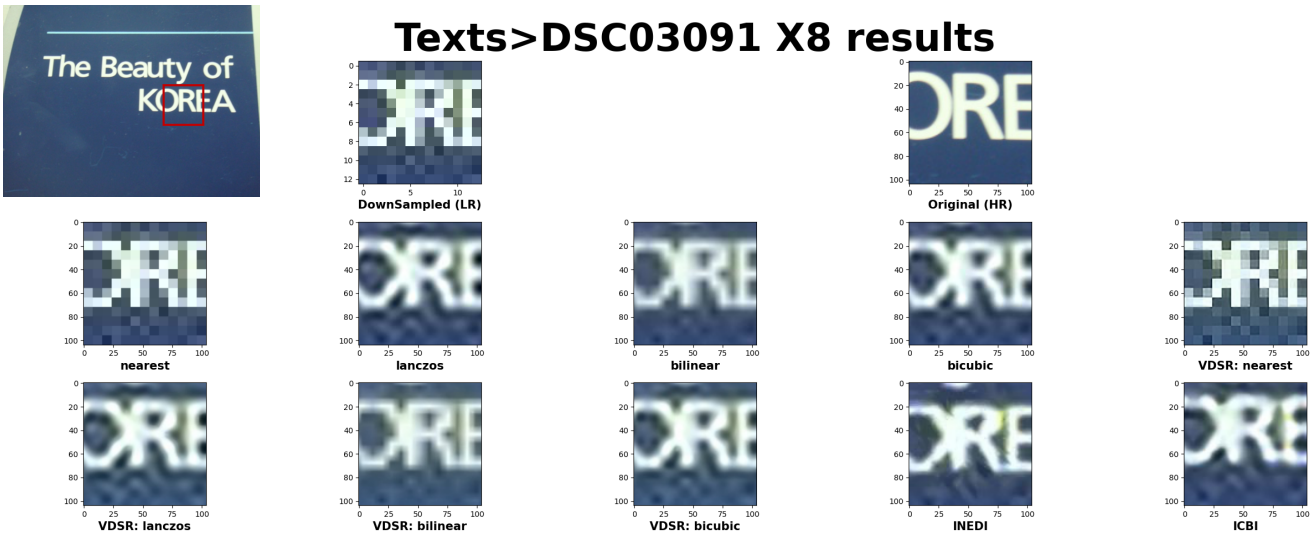

Figure V-16: tc-100, DSC3091 image with a scaling factor of 8, zoomed image.

Figure V-17: tc-100, DSC3091 image with a scaling factor of 8, overall result.

Overall conclusion

Overall results based on objective IQA metrics can be interpreted as 'ICBI' is the dominating method for up-sampling most categories and scaling factors (except scaling factor of 3, which is unavailable to perform with 'ICBI'). The PSNR difference between 'ICBI' and the worst-performing method is about 5dB, while the difference between bilinear interpolation is about 1.5dB. The RMSE shows about 0.003 difference both for the worst-performing method and bilinear interpolation. This is because of the calculation of RMSE, which targeted to have 0 when the similarity between up-sampled image and ground-truth image increases. Therefore, the number of RMSE is extremely small. The FSIM difference between 'ICBI' and the worst method is about 0.1 for all scaling factor, while the difference between bilinear and 'ICBI' is around 0.01~0.05. The SSIM difference between 'ICBI' and the worst-performing method is about 0.03, and the difference between bilinear interpolation and 'ICBI' is about 0.01.

The 'RMSE' and 'PSNR' are used to compare the similarity between the up-sampled image and the ground-truth image. The +1.5dB of PSNR between 'bilinear interpolation' and 'ICBI' means 'ICBI' is a well-performing method. (Other proposal papers usually mentions that their method is a better way to up-sample by referring to +0.3~3.0dB of PSNR). The 'SSIM' and 'FSIM' results that ICBI has better IQA metrics than bilinear interpolation by about +0.03.

According to Figure V-18, 19, 20 and 21, the IQA metric shows that all methods perform best on a scaling factor of 2 and worsen as the scaling factor increases. In addition, 'SSIM', 'PSNR' and 'RMSE' shows that all the methods perform best on 'tc-100 (text dataset)' and worst on 'Urban100'. This means all methods struggle to up-sample image with the pattern. For 'FSIM', 'tc-100 (text dataset)' shows the worst performance most methods. This can be interpreted as the understanding of the up-sampled image based on feature is difficult for text-based images compare to other image sets.

The mean IQA metrics without considering the dataset are plotted on Figure V-22, 23, 24 and 25. The figures show that 'ICBI' shows the best performance for all IQA metrics, and most other deep-learning-based methods show generally outstanding performance.

Table V-5: The average IQA metrics of all categories by each IQA metric. (blank means the method/model is unavailable for the benchmark condition.)

iqa	factor	DRCNN	EDSR	ESPCN	FSRCNN	ICBI	INEDI	LapSRN	VDSR_ bicubic	VDSR_ bilinear	VDSR_ lanczos	VDSR_ nearest	bicubic	bilinear	lanczos	nearest
FSIM	2	0.642	0.691	0.688	0.682	**0.699**	0.645	0.682	0.643	0.67	0.618	0.607	0.679	0.684	0.668	0.678
PSNR	2	48.827	51.078	51.187	50.972	**52.847**	49.556	51.143	46.477	47.679	45.956	45.308	51.016	51.265	50.86	49.599
RMSE	2	0.004	**0.003**	0.003	0.003	0.003	0.004	0.003	0.005	0.004	0.005	0.006	0.003	0.003	0.003	0.004
SSIM	2	0.981	0.989	0.989	0.989	**0.992**	0.984	0.989	0.972	0.978	0.969	0.964	0.989	0.989	0.988	0.985
FSIM	3	0.574	**0.585**	0.581	0.58				0.553	0.576	0.531	0.505	0.579	0.582	0.573	0.508
PSNR	3	47.206	47.576	47.863	47.803				44.245	45.667	43.67	43.91	47.645	**47.943**	47.499	46.453
RMSE	3	0.005	0.005	**0.004**	0.005				0.007	0.006	0.007	0.007	0.005	0.004	0.005	0.005
SSIM	3	0.974	0.976	**0.978**	0.977				0.955	0.966	0.95	0.953	0.977	0.978	0.976	0.97
FSIM	4	0.516	0.52	0.516	0.513	**0.571**	0.483	0.516	0.504	0.518	0.499	0.44	0.517	0.52	0.516	0.426
PSNR	4	45.855	45.804	46.183	46.085	**48.281**	44.652	46.112	43.416	44.814	43.346	42.782	45.901	46.394	45.744	44.829
RMSE	4	0.006	0.006	0.005	0.005	**0.004**	0.006	0.005	0.007	0.006	0.007	0.008	0.006	0.005	0.006	0.006
SSIM	4	0.966	0.966	0.969	0.968	**0.979**	0.956	0.968	0.946	0.96	0.945	0.94	0.967	0.97	0.966	0.959
FSIM	8					**0.459**	0.392		0.408	0.407	0.404	0.35	0.408	0.409	0.404	0.303
PSNR	8					**44.735**	41.63		41.945	42.426	41.812	40.961	42.66	43.199	42.508	41.761
RMSE	8					**0.006**	0.009		0.008	0.008	0.009	0.01	0.008	0.007	0.008	0.009
SSIM	8					**0.958**	0.926		0.93	0.937	0.928	0.916	0.94	0.946	0.938	0.928

Table V-6: The up-sampling method's name by minimum and maximum value from each IQA metric. The difference between the minimum and maximum value and the difference between bilinear interpolation and maximum value.

iqa	factor	Minimum value method	Maximum value method	IQA Difference min - max	IQA Difference Bilinear - max
FSIM	2	VDSR_nearest	ICBI	0.092	0.015
PSNR	2	VDSR_nearest	ICBI	7.539	1.582
RMSE	2	EDSR	VDSR_nearest	0.003	0.003
SSIM	2	VDSR_nearest	ICBI	0.028	0.003
FSIM	3	VDSR_nearest	EDSR	0.08	0.003
PSNR	3	VDSR_lanczos	bilinear	4.273	0.0
RMSE	3	ESPCN	VDSR_bicubic	0.003	0.003
SSIM	3	VDSR_lanczos	ESPCN	0.028	0.0
FSIM	4	nearest	ICBI	0.145	0.051
PSNR	4	VDSR_nearest	ICBI	5.499	1.887
RMSE	4	ICBI	VDSR_nearest	0.004	0.003
SSIM	4	VDSR_nearest	ICBI	0.039	0.009
FSIM	8	nearest	ICBI	0.156	0.05
PSNR	8	VDSR_nearest	ICBI	3.774	1.536
RMSE	8	ICBI	VDSR_nearest	0.004	0.003
SSIM	8	VDSR_nearest	ICBI	0.042	0.012

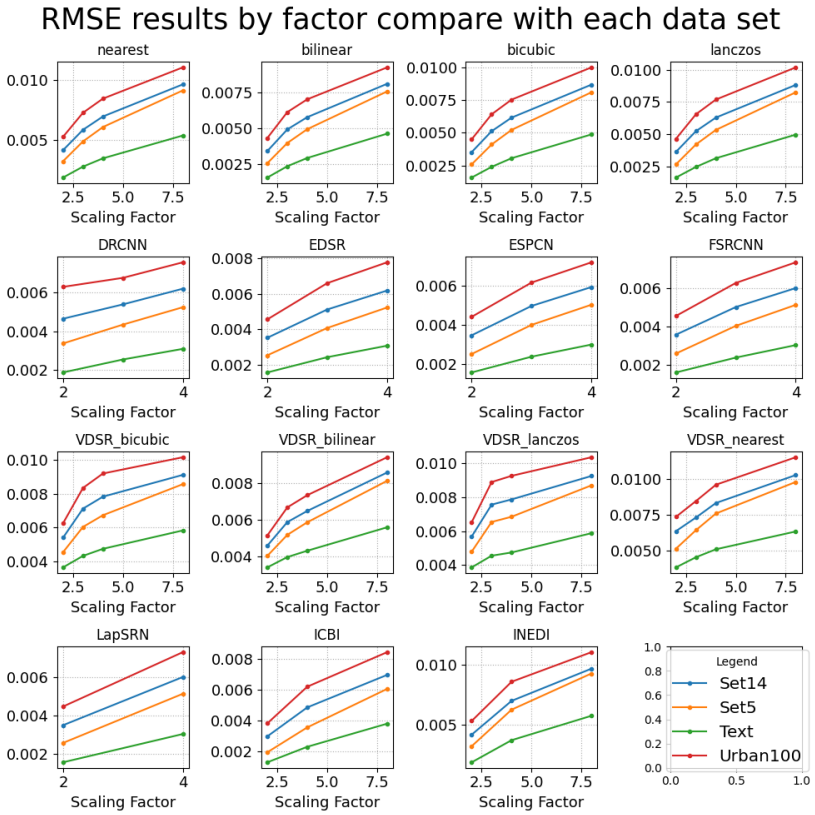

Figure V-18: RMSE results of each up-sampling methods (x-axis: scaling factor, y-axis: RMSE value)

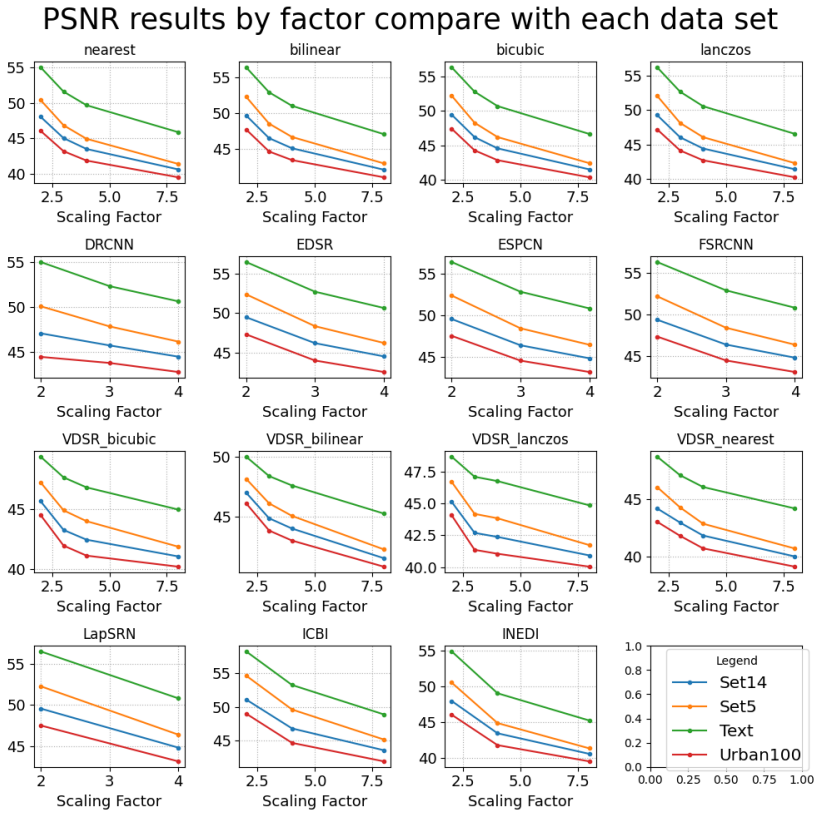

Figure V-19: PSNR results of each up-sampling methods (x-axis: scaling factor, y-axis: RMSE value)

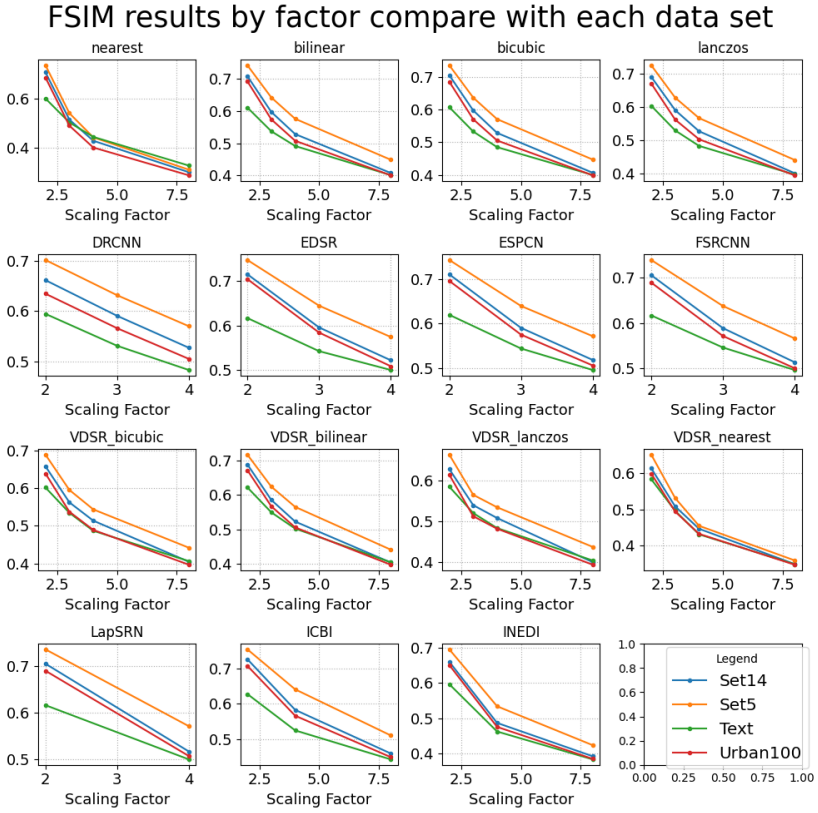

Figure V-20: FSIM results of each up-sampling methods (x-axis: scaling factor, y-axis: RMSE value)

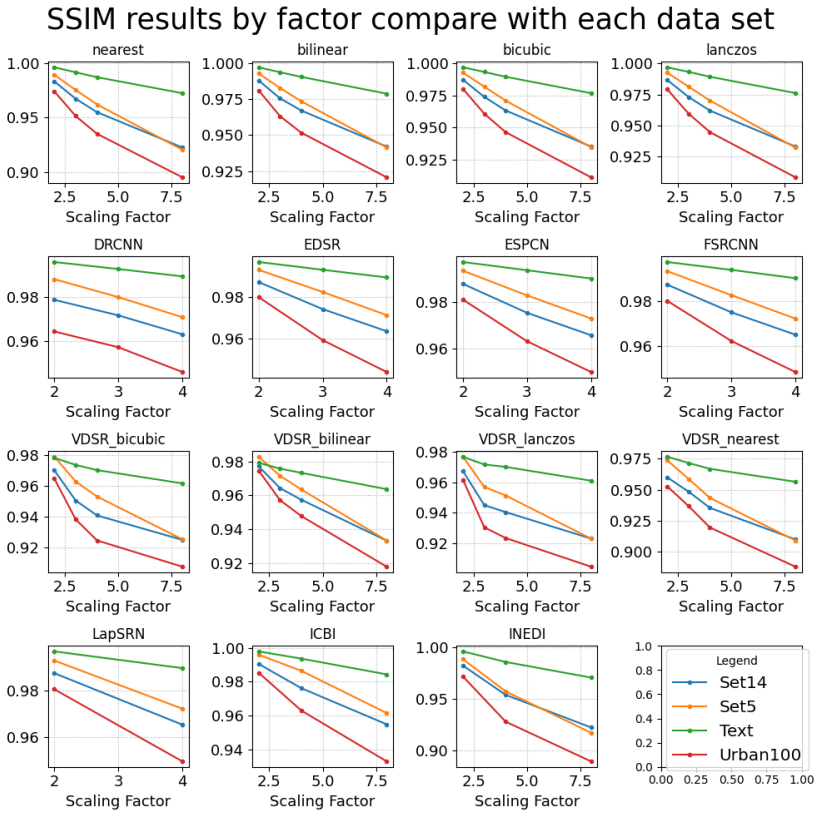

Figure V-21: SSIM results of each up-sampling methods (x-axis: scaling factor, y-axis: RMSE value)

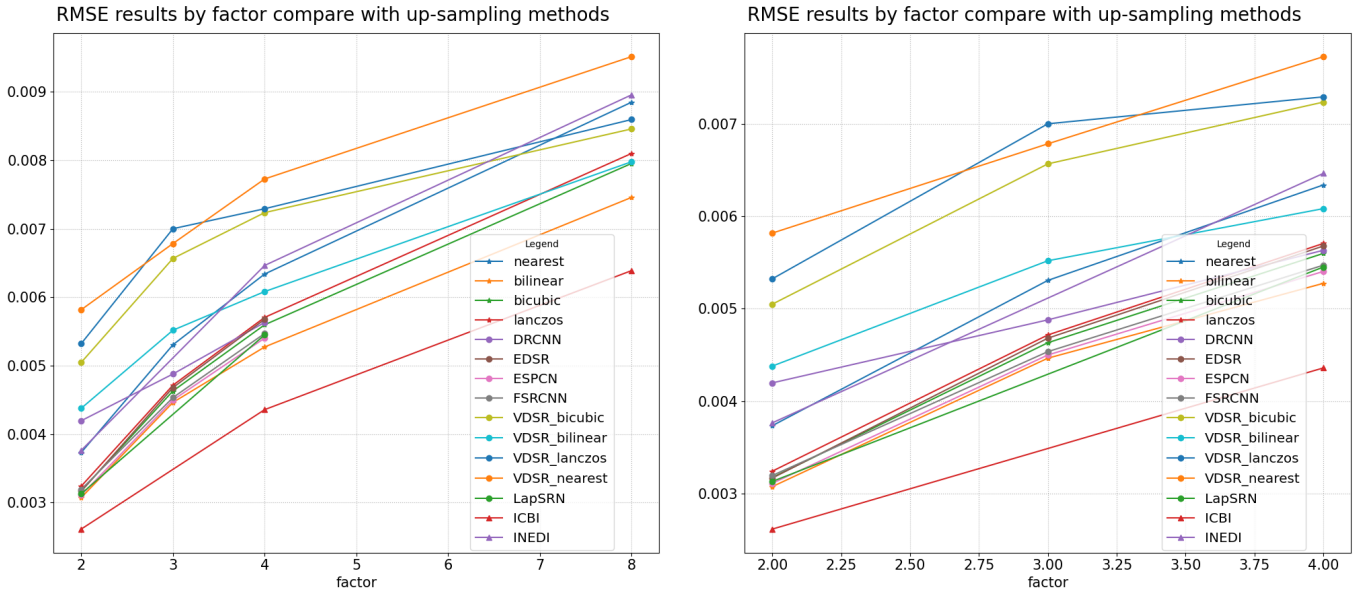

Figure V-22: The mean RMSE value of each up-sampling methods

(x-axis: scaling factor, y-axis: RMSE value, left graph: scaling factor 2-8, right graph: scaling factor 2-4)

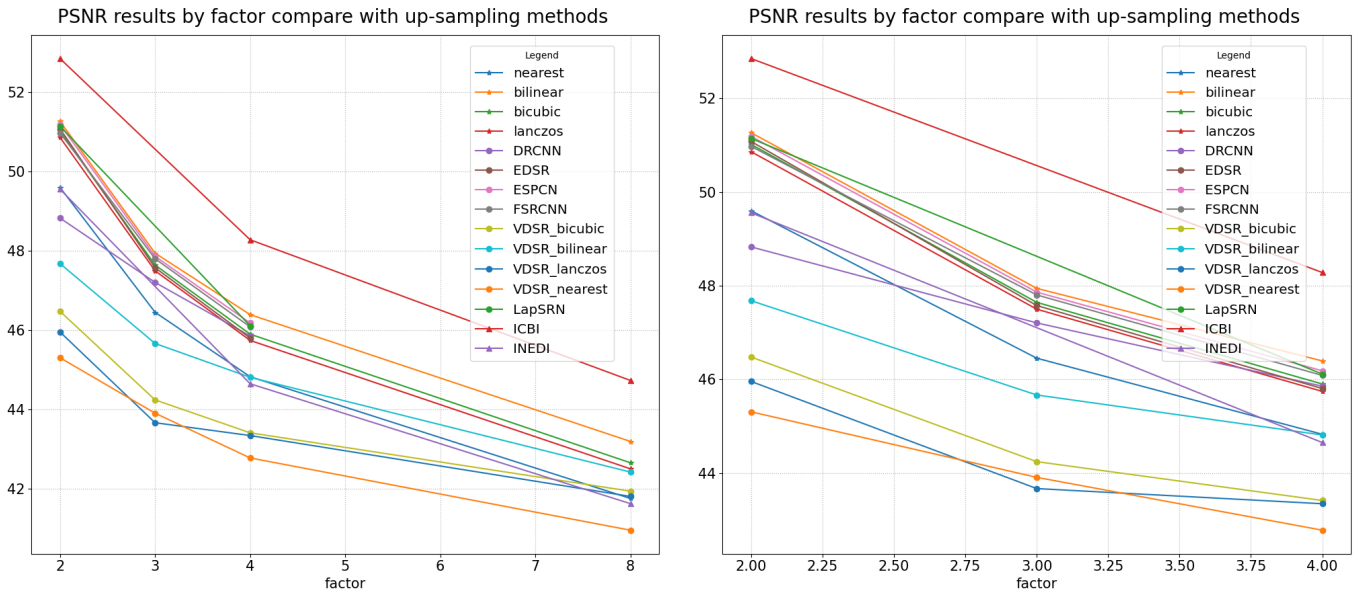

Figure V-23: The mean PSNR value of each up-sampling

(x-axis: scaling factor, y-axis: PSNR value, left graph: scaling factor 2-8, right graph: scaling factor 2-4)

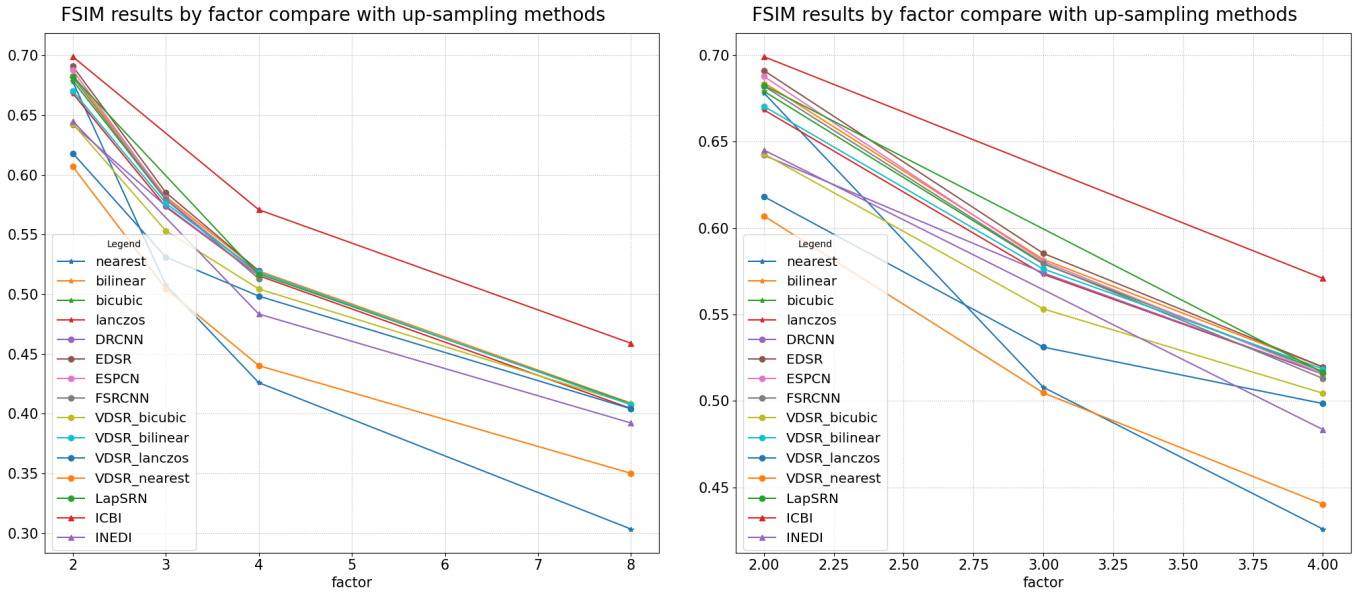

Figure V-24: The mean FSIM value of each up-sampling methods

(x-axis: scaling factor, y-axis: FSIM value, left graph: scaling factor 2-8, right graph: scaling factor 2-4)

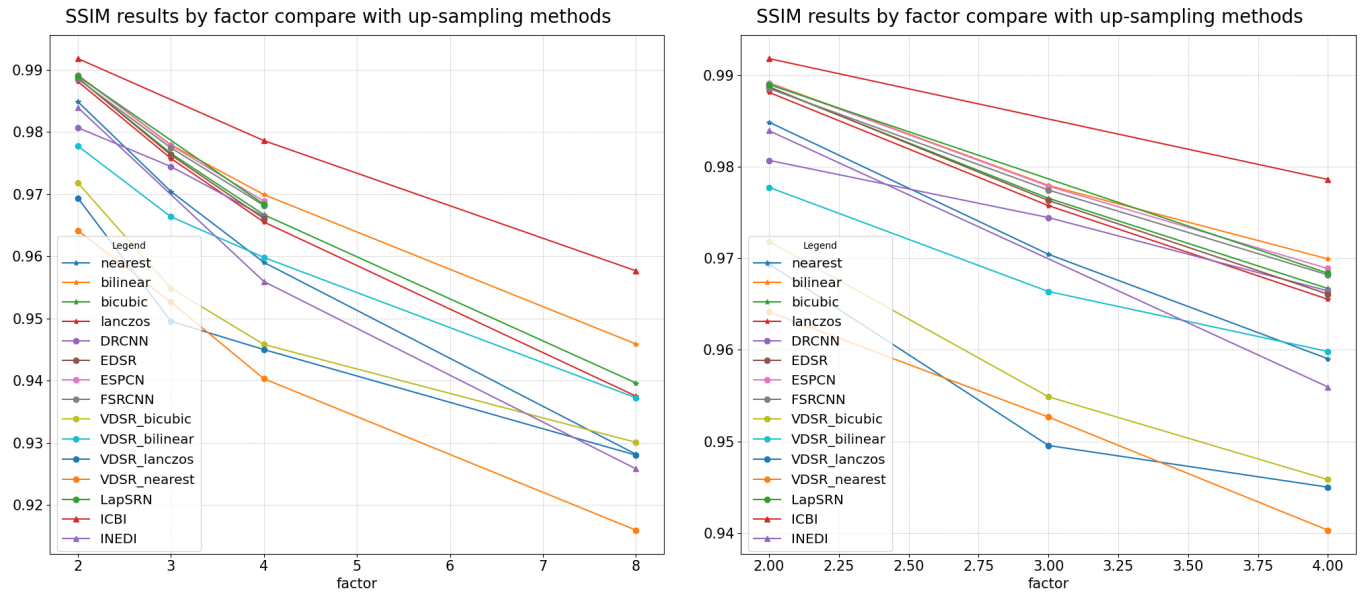

Figure V-25: SSIM results of each up-sampling methods

(x-axis: scaling factor, y-axis: SSIM value, left graph: scaling factor 2-8, right graph: scaling factor 2-4)

VI. Discussions

The objective Image Quality Assessment metrics result that 'ICBI' is the best performing method. However, by verifying every result image, the limitations of IQA metrics are recognised. In addition, limitations and issues that happened during the project will be described in this section.

Limitation of objective IQA metrics

The FSIM and SSIM are proposed to overcome the limitation of existing objective IQA methods. It is designed to have a high correlation between human visual assessment, which is subjective IQA. However, the IQA metrics still showed some difference between human visual recognition and the numerical result from the metrics. This means the IQA metrics is having challenges of having a high correlation with human visual perception.

The limited computing resource (Laptop with i7-8750H) caused the increase of computing time for both SR methods and IQA methods. Nevertheless, the other IQA metrics such as 'Universal Image Quality Index (UIQ)', 'Signal to Reconstruction Error ratio (SRE)' and 'Information theoretic-based Statistic Similarity Measure (ISSM)' were considered for this project. The limited time for the project and the computing power made it to cut off some IQA methods. If more computing power such as a Linux server with GPU was provided, more IQA metrics and more complex SR methods/models could be included in this project. Moreover, the following research topic will aim to make the IQA metric have a higher correlation between human visual perception using both statistical and deep-learning methods.

Pretrained deep-learning model issue

Due to limited computing resources, it was impossible to train each deep-learning model on the laptop. The laptop also does not have a compatible GPU that can do calculation acceleration for either 'TensorFlow' or 'PyTorch', which are deep-learning frameworks. Therefore, pre-trained models are used for the project to reduce the use of computing resource and time. However, the pre-trained models are trained by other images sets by model providers. The main goal of this project is to assess the performance of various SR methods for text-based images. Therefore, the models should be trained in not only general scene images but also the image that contains texts. However, most of pre-trained

models are using general scene images for training set. For this reason, deep-learning model may had performance drop for up-sampling text-based images.

Model/Method Running time

The runtime of models/methods is also an essential factor. The more time requires for the method the less image can proceed for an equal amount of time. The edge-preserving methods, 'ICBI' and 'iNEDI', showed about 100 times of runtime compared to other methods. Because this was not the case to be considered at the beginning of the project, each method's runtime was not measured and recorded. The reason for this issue may be caused by the way of processing the method. The statistical and mathematical methods process all the variables and pixel values during runtime. However, the deep-learning models achieves the filters and layers' variables and coefficient during training time and process the input image with matrix calculation. As deep-learning model uses less complex calculating terms compare to mathematical and statistical methods such as 'ICBI' and 'iNEDI', the runtime of deep-learning method is shorter than the others.

Furthermore, the preparation experiment of this project showed a small amount of similar runtime for all methods. Therefore, the runtime was decided not to be recorded as it increases the overall running time. If further research is available after this project, the runtime for each method and image will be recorded. Moreover, implementing the methods that uses mathematical and statistical strategies to deep-learning related framework can be done on future research. This may lead to new perspective of SR problem.

Artifacts produced on Edge-preserving Methods

The artefacts (red, yellow, and blue dots) have been produced on both 'iNEDI' and 'ICBI' for a scaling factor of 8. This may be caused by the up-sampling equation that both methods use refers to 4~16 nearby pixels. The down-sampled image (scaling factor of 8) shows hardly recognisable borderline between objects and background or objects and objects. This means that the two methods refer to the pixels from more than two different objects and backgrounds. If the colour of the pixels are complementary colours or have huge luminance difference, the two SR methods may result in some weird pixels.

This issue was not found on deep-learning methods, the state-of-the-art technique of SR problem. This is due to the difference in calculating target up-sample pixels. Therefore, it can be solved using deep-learning models that use various simple filters and layers instead of complex filters.

VII. Conclusions/Future Works

The text-based images captured still have low-resolution even the camera performance improves as people often try to capture the text from a far distance. In addition, the number plate on car is sometimes difficult to recognise when the CCTV on the highway is old model which has low capturing resolution. The captured image may be difficult to interpret easily as the character looks squashed. Several SISR methods have been developed and proposed for an extended period that advanced gradually to solve the low-resolution problem for text-based images and general scene images. Starting from 'basic interpolation methods', 'edge-preserving methods' and 'deep-learning methods' are dealt with for this project to assess the performance, especially for text-based images.

The benchmark has been proceeded by down-sampling the target image sets, including text-based images, besides general scene images and pattern images. Then the down-sampled images are up-sampled with the selected up-sampling methods. Basic interpolation methods are chosen to use as a baseline to compare other methods. Edge-preserving methods are selected to expect the methods have strength in up-sampling text-based images. Lastly, deep-learning methods are the state-of-the-art technology in the SISR field, but only the well-proved methods are selected. The up-sampled pictures are then compared with ground-truth images with IQA metrics.

Furthermore, to provide other regular users with ease of using up-sampling methods to obtain images with better recognition, the GUI is developed and supplied on the 'Github' repository. The GUI has been developed with 'PyQt5', which is a python framework.

The result of IQA metrics shows that 'ICBI', an edge-preserving method, is the best performing method. However, the verification of the up-sampled images results that IQA metrics cannot correlate highly with human visual reception. When the scaling factor increases, the general trend of all methods increases in IQA metrics. In addition, all methods show the best performance with a text-based image set. Taking mean value not considering the type of dataset results 'ICBI' is the best performing method. This result based on IQA metrics concludes that 'ICBI' is the best performing method for all types of images but shows best on text-based images. This can also inform people to use the deep-learning form and mathematically and statistically driven methods for up-sampling photos such as 'ICBI'.

This research topic, related to SISR, can be progressed in three directions, as discussed in Section VI. Primarily, more recent methods and more images can be analysed with better computation resources. Ultimately, the webpage that provides the benchmark of published SISR methods in real-time and the interface to test can be developed. Secondly, a more progressed model or method can be proposed for a specific condition. For instance, the SR methods with a scaling factor of 8 still need to be researched and developed. This may be more important to image with text as the public 3d map builders collect images while riding a car with a camera, far from the pedestrian road. Lastly, a better IQA metric can be developed to achieve a better correlation with human visual assessment. The metric may use not the only statistical model but also deep-learning-based feature selecting methods.

VIII. References

[1] W. T. Freeman, T. R. Jones, and E. C. Pasztor, "Example-based super-resolution," *IEEE Computer Graphics and Applications*, vol. 22, no. 2, pp. 56–65, Mar. 2002, doi: 10.1109/38.988747.

[2] D. Glasner, S. Bagon, and M. Irani, "Super-resolution from a single image," in *2009 IEEE 12th International Conference on Computer Vision*, Sep. 2009, pp. 349–356, doi: 10.1109/ICCV.2009.5459271.

[3] A. Rueda, N. Malpica, and E. Romero, "Single-image super-resolution of brain MR images using overcomplete dictionaries," *Medical Image Analysis*, vol. 17, no. 1, pp. 113–132, 2013, doi: https://doi.org/10.1016/j.media.2012.09.003.

[4] Y. Huang, L. Shao, and A. Frangi, "Simultaneous Super-Resolution and Cross-Modality Synthesis of 3D Medical Images Using Weakly-Supervised Joint Convolutional Sparse Coding," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5787–5796, 2017.

[5] H. Greenspan, "Super-Resolution in Medical Imaging," *The Computer Journal*, vol. 52, no. 1, pp. 43–63, Apr. 2008, doi: 10.1093/comjnl/bxm075.

[6] J. S. Isaac and R. Kulkarni, "Super resolution techniques for medical image processing," in *2015 International Conference on Technologies for Sustainable Development (ICTSD)*, Feb. 2015, pp. 1–6, doi: 10.1109/ICTSD.2015.7095900.

[7] L. Schermelleh *et al.*, "Super-resolution microscopy demystified," *Nature Cell Biology*, vol. 21, no. 1, pp. 72–84, 2019, doi: 10.1038/s41556-018-0251-8.

[8] S. J. Holden, S. Uphoff, and A. N. Kapanidis, "DAOSTORM: an algorithm for high- density super-resolution microscopy," *Nature Methods*, vol. 8, no. 4, pp. 279–280, 2011, doi: 10.1038/nmeth0411-279.

[9] G. Huszka and M. A. M. Gijs, "Super-resolution optical imaging: A comparison," *Micro and Nano Engineering*, vol. 2, pp. 7–28, 2019, doi: https://doi.org/10.1016/j.mne.2018.11.005.

[10] A. van Etten, "Satellite Imagery Multiscale Rapid Detection with Windowed Networks," in *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, Jan. 2019, pp. 735–743, doi: 10.1109/WACV.2019.00083.

[11] J. Shermeyer and A. van Etten, "The Effects of Super-Resolution on Object Detection Performance in Satellite Imagery," *CoRR*, vol. abs/1812.04098, 2018, [Online]. Available: http://arxiv.org/abs/1812.04098.

[12] B. K. Gunturk, A. U. Batur, Y. Altunbasak, M. H. Hayes, and R. M. Mersereau, "Eigenface-domain super-resolution for face recognition," *IEEE Transactions on Image Processing*, vol. 12, no. 5, pp. 597–606, May 2003, doi: 10.1109/TIP.2003.811513.

[13] F. Lin, C. Fookes, V. Chandran, and S. Sridharan, "Investigation into optical flow super-resolution for surveillance applications," in *WDIC 2005: APRS Workshop on Digital Image Computing: Workshop Proceedings:*, 2005, pp. 73–78.

[14] W. W. W. Zou and P. C. Yuen, "Very low resolution face recognition problem," in *Image Processing, IEEE Transactions on*, Apr. 2010, vol. 21, pp. 1–6, doi: 10.1109/BTAS.2010.5634490.

[15] J. A. Parker, R. v Kenyon, and D. E. Troxel, "Comparison of Interpolating Methods for Image Resampling," *IEEE Transactions on Medical Imaging*, vol. 2, no. 1, pp. 31–39, Mar. 1983, doi: 10.1109/TMI.1983.4307610.

[16] P. Getreuer, "Linear methods for image interpolation," *Image Processing On Line*, vol. 1, pp. 238–259, 2011.

[17] C. E. Duchon, “Lanczos filtering in one and two dimensions,” *Journal of Applied Meteorology and Climatology*, vol. 18, no. 8, pp. 1016–1022, 1979.

[18] K. I. Kim and Y. Kwon, “Single-Image Super-Resolution Using Sparse Regression and Natural Image Prior,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 6, pp. 1127–1133, Jun. 2010, doi: 10.1109/TPAMI.2010.25.

[19] J. Yang, J. Wright, T. S. Huang, and Y. Ma, “Image super-resolution via sparse representation,” *IEEE transactions on image processing*, vol. 19, no. 11, pp. 2861–2873, 2010.

[20] W. T. Freeman, T. R. Jones, and E. C. Pasztor, “Example-based super-resolution,” *IEEE Computer Graphics and Applications*, vol. 22, no. 2, pp. 56–65, Mar. 2002, doi: 10.1109/38.988747.

[21] G. Freedman and R. Fattal, “Image and video upscaling from local self-examples,” *ACM Transactions on Graphics (TOG)*, vol. 30, no. 2, pp. 1–11, 2011.

[22] O. mac Aodha, N. D. F. Campbell, A. Nair, and G. J. Brostow, “Patch based synthesis for single depth image super-resolution,” in *European conference on computer vision*, 2012, pp. 71–84.

[23] H. Chang, D.-Y. Yeung, and Y. Xiong, “Super-resolution through neighbor embedding,” in *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, 2004, vol. 1, pp. I–I.

[24] Y. Tai, S. Liu, M. S. Brown, and S. Lin, “Super resolution using edge prior and single image detail synthesis,” in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Jun. 2010, pp. 2400–2407, doi: 10.1109/CVPR.2010.5539933.

[25] L. Wang, S. Xiang, G. Meng, H. Wu, and C. Pan, “Edge-Directed Single-Image Super-Resolution Via Adaptive Gradient Magnitude Self-Interpolation,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 23, no. 8, pp. 1289–1299, Aug. 2013, doi: 10.1109/TCSVT.2013.2240915.

[26] D. Glasner, S. Bagon, and M. Irani, “Super-resolution from a single image,” in *2009 IEEE 12th International Conference on Computer Vision*, Sep. 2009, pp. 349–356, doi: 10.1109/ICCV.2009.5459271.

[27] C. Dong, C. C. Loy, K. He, and X. Tang, “Learning a deep convolutional network for image super-resolution,” in *European conference on computer vision*, 2014, pp. 184–199.

[28] C. Dong, C. C. Loy, and X. Tang, “Accelerating the Super-Resolution Convolutional Neural Network,” *CoRR*, vol. abs/1608.00367, 2016, [Online]. Available: http://arxiv.org/abs/1608.00367.

[29] J. Kim, J. K. Lee, and K. M. Lee, “Deeply-Recursive Convolutional Network for Image Super-Resolution,” Nov. 2015, [Online]. Available: http://arxiv.org/abs/1511.04491.

[30] J. Kim, J. K. Lee, and K. M. Lee, “Accurate Image Super-Resolution Using Very Deep Convolutional Networks,” Nov. 2015, [Online]. Available: http://arxiv.org/abs/1511.04587.

[31] W. Shi *et al.*, “Real-Time Single Image and Video Super-Resolution Using an Efficient Sub-Pixel Convolutional Neural Network,” *CoRR*, vol. abs/1609.05158, 2016, [Online]. Available: http://arxiv.org/abs/1609.05158.

[32] Z. Wang, J. Chen, and S. C. H. Hoi, “Deep Learning for Image Super-resolution: A Survey,” Feb. 2019, [Online]. Available: http://arxiv.org/abs/1902.06068.

[33] C. Ledig *et al.*, “Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network,” 2017.

[34] W. Sun and Z. Chen, “Learned Image Downscaling for Upscaling Using Content Adaptive Resampler,” *IEEE Transactions on Image Processing*, vol. 29, pp. 4027–4040, Apr. 2020, doi: 10.1109/TIP.2020.2970248.

[35] W. Yang, X. Zhang, Y. Tian, W. Wang, and J.-H. Xue, “Deep Learning for Single Image Super-Resolution: A Brief Review,” Aug. 2018, doi: 10.1109/TMM.2019.2919431.

[36] J. Tian and K.-K. Ma, “A survey on super-resolution imaging,” *Signal, Image and Video Processing*, vol. 5, no. 3, pp. 329–342, 2011.

[37] J. D. van Ouwerkerk, “Image super-resolution survey,” *Image and Vision Computing*, vol. 24, no. 10, pp. 1039–1052, 2006, doi: https://doi.org/10.1016/j.imavis.2006.02.026.

[38] K. Nasrollahi and T. B. Moeslund, “Super-resolution: a comprehensive survey,” *Machine Vision and Applications*, vol. 25, no. 6, pp. 1423–1468, 2014, doi: 10.1007/s00138-014-0623-4.

[39] Sung Cheol Park, Min Kyu Park, and Moon Gi Kang, “Super-resolution image reconstruction: a technical overview,” *IEEE Signal Processing Magazine*, vol. 20, no. 3, pp. 21–36, May 2003, doi: 10.1109/MSP.2003.1203207.

[40] Q. Ye and D. Doermann, “Text Detection and Recognition in Imagery: A Survey,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 7, pp. 1480–1500, Jul. 2015, doi: 10.1109/TPAMI.2014.2366765.

[41] A. Risnumawan, I. A. Sulistijono, and J. Abawajy, “Text detection in low resolution scene images using convolutional neural network,” in *International Conference on Soft Computing and Data Mining*, 2016, pp. 366–375.

[42] D. Han, “Comparison of Commonly Used Image Interpolation Methods,” 2013.

[43] K. Turkowski, “Filters for common resampling tasks,” *Graphics Gems I*, pp. 147–165, 1990.

[44] P. Parsania and D. Virparia, “A Comparative Analysis of Image Interpolation Algorithms,”

IJARCCE, vol. 5, pp. 29–34, Apr. 2016, doi: 10.17148/IJARCCE.2016.5107.

[45] C. Suresh, S. Singh, R. Saini, and A. K. Saini, "A Comparative Analysis of Image Scaling Algorithms," *International Journal of Image, Graphics and Signal Processing*, vol. 5, no. 5, p. 55, 2013.

[46] C. Dong, C. C. Loy, K. He, and X. Tang, "Image Super-Resolution Using Deep Convolutional Networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, pp. 295–307, 2016, Accessed: Apr. 02, 2021. [Online]. Available: https://arxiv.org/pdf/1501.00092.pdf.

[47] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *CoRR*, vol. abs/1409.1556, 2015.

[48] W.-S. Lai, J.-B. Huang, N. Ahuja, and M.-H. Yang, "Deep laplacian pyramid networks for fast and accurate super-resolution," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 624–632.

[49] P. Burt and E. Adelson, "The Laplacian Pyramid as a Compact Image Code," *IEEE Transactions on Communications*, vol. 31, no. 4, pp. 532–540, Apr. 1983, doi: 10.1109/TCOM.1983.1095851.

[50] B. Lim, S. Son, H. Kim, S. Nah, and K. M. Lee, "Enhanced Deep Residual Networks for Single Image Super-Resolution."

[51] X. Li and M. T. Orchard, "New Edge-Directed Interpolation," 2001.

[52] N. Asuni and A. Giachetti, "Accuracy Improvements and Artifacts Removal in Edge Based Image Interpolation," 2008.

[53] K. S. Reddy and D. K. R. L. Reddy, "Enlargement of image based upon Interpolation Techniques," *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 2, no. 12, p. 4631, 2013.

[54] A. Giachetti and N. Asuni, "Real-Time Artifact-Free Image Upscaling," *IEEE Transactions on Image Processing*, vol. 20, no. 10, pp. 2760–2768, Oct. 2011, doi: 10.1109/TIP.2011.2136352.

[55] R. Borse and P. Markad, "Competitive analysis of existing image quality assessment methods," in *2014 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, Sep. 2014, pp. 1440–1444, doi: 10.1109/ICACCI.2014.6968385.

[56] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image Quality Assessment: From Error Visibility to Structural Similarity," 2004. [Online]. Available: http://www.cns.nyu.edu/~lcv/ssim/.

[57] L. Zhang, L. Zhang, X. Mou, and D. Zhang, "FSIM: A feature similarity index for image quality assessment," *IEEE Transactions on Image Processing*, vol. 20, no. 8, pp. 2378–2386, Aug. 2011, doi: 10.1109/TIP.2011.2109730.

[58] R. Jain, R. Kasturi, and B. G. Schunck, *Machine vision*, vol. 5. McGraw-hill New York, 1995.

[59] B. Jähne, H. Haussecker, and P. Geissler, *Handbook of computer vision and applications*, vol. 2. Citeseer, 1999.

[60] R. Zeyde, M. Elad, and M. Protter, "On single image scale-up using sparse-representations," in *International conference on curves and surfaces*, 2010, pp. 711–730.

[61] J.-B. Huang, A. Singh, and N. Ahuja, "Single Image Super-Resolution From Transformed Self-Exemplars," Jun. 2015.

[62] J.-B. Huang, A. Singh, and N. Ahuja, "Single image super-resolution from transformed self-exemplars," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 5197–5206.

[63] J. Jung, S. Lee, M. S. Cho, and J. H. Kim, "Touch TT: Scene Text Extractor Using Touchscreen Interface," *ETRI Journal*, vol. 33, no. 1, pp. 78–88, 2011, doi: https://doi.org/10.4218/etrij.11.1510.0029.

[64] S. Lee, M. S. Cho, K. Jung, and J. H. Kim, "Scene Text Extraction with Edge Constraint and Text Collinearity," in *2010 20th International Conference on Pattern Recognition*, Aug. 2010, pp. 3983–3986, doi: 10.1109/ICPR.2010.969.

[65] M. Müller, N. Ekhtiari, R. M. Almeida, and C. Rieke, "Super-resolution of multispectral satellite images using convolutional neural networks," *ArXiv*, vol. abs/2002.00580, 2020, Accessed: Apr. 02, 2021. [Online]. Available: https://arxiv.org/pdf/2002.00580.pdf.

IX. Appendix

Table IX-1: List of program, algorithm and files used for this project

Filename/Algorithm/ Package	Supplier/Source/Author/website	Use/Modifications made/ Student written
EDSR_x#.pb (#=2, 3, 4)	Saafke – Github / TensorFlow model save file. (python) https://github.com/Saafke/EDSR_Tensorflow	EDSR pre-trained model for benchmark by loading with TensorFlow.
ESPCN_x#.pb (#=2, 3, 4)	Fannymonori – Github / TensorFlow model save file. (python) https://github.com/fannymonori/TF-ESPCN	ESPCN pre-trained model for benchmark by loading with TensorFlow.
FSRCNN_x#.pb (#=2, 3 ,4)	Saafke – Github / TensorFlow model save file. (python) https://github.com/fannymonori/TF-ESPCN	FSRCNN pre-trained model for benchmark by loading with TensorFlow.
LapSRN_x#.pb (#=2, 4, 8)	Fannymonori – Github / TensorFlow model save file. (python) https://github.com/fannymonori/TF-LapSRN	LapSRN pre-trained model for benchmark by loading with TensorFlow.
DRCN (Package)	Jiny2001 – Github / (TensorFlow - python) https://github.com/jiny2001/deeply-recursive-cnn-tf	DRCNN model implementation with full source codes and pre-trained model used for benchmark.
VDSR (Package)	twtygqyy – Github / (Pytorch – python) https://github.com/twtygqyy/pytorch-vdsr	VDSR model implementation with full source codes and pre-trained model used for benchmark.
iCBI (Package)	http://www.andreagiachetti.it/icbi/ (MATLAB)	iCBI implementation codes used for benchmark.
iNEDI (Package)	nicolaasuni – Github / (MATLAB) https://github.com/tecnickcom/inedi	iNEDI implementation codes used for benchmark.
PyQt5 (Library)	Python library	PyQt5 library is used for building GUI Front-End views.
PIL (Library)	Python library	PIL library is used to save image.
OpenCV2 (Library)	Python library	CV2 library is used for loading, processing image and loading TensorFlow saved Images.
Numpy (Library)	Python library	Numpy library provides efficient tools to process arrays, matrices and vectors.
Pandas (Library)	Python library	Pandas library is used to handle data in table format and ease to save in csv.
Image_similarity_measure (Library)	Python library	Image-similarity-measure library provides functions of Image Quality Assessment methods.
Matplotlib (Library)	Python library	Matplotlib provides data visualization tool almost identical to MATLAB. This library is used to check process of benchmark.
Plotly (Library)	Python library	Plotly is used to visualizing final IQA results along with PyQt5 on GUI.
Sisr_project/src/{backend_api, frontend_ui}	Written codes by Jongyoon Kim	This directory is written by Jongyoon Kim and non-code files are temporary results during code implementations.
.ui files in sisr_project/src/frontend_ui/ui	UI files designed by Jongyoon Kim	The ui files are produced by building GUI designs by Jongyoon Kim.

Table IX-2: IQA metrics of Set5, scale factor 2

image	iqa	DRCN	EDSR	ESPCN	FSRCNN	ICBI	INEDI	LapSRN	VDSR bicubic	VDSR bilinear	VDSR lanczos	VDSR nearest	bicubic	bilinear	lanczos	nearest	best
baby	FSIM	0.721	0.752	0.751	0.746	**0.767**	0.7	0.748	0.707	0.734	0.684	0.661	0.75	0.748	0.745	0.747	ICBI
	PSNR	53.904	55.842	55.793	55.51	**57.697**	54.089	55.709	48.794	49.306	48.485	47.781	55.755	55.94	55.647	53.991	ICBI
	RMSE	0.002	0.002	0.002	0.002	**0.001**	0.002	0.002	0.004	0.003	0.004	0.004	0.002	0.002	0.002	0.002	ICBI
	SSIM	0.996	0.997	0.997	0.997	**0.998**	0.996	0.997	0.991	0.992	0.99	0.987	0.997	0.997	0.997	0.996	ICBI
bird	FSIM	0.771	**0.797**	0.792	0.788	0.778	0.73	0.787	0.734	0.749	0.715	0.703	0.79	0.792	0.785	0.785	EDSR
	PSNR	52.439	54.273	54.114	53.898	**55.857**	51.887	53.874	47.511	47.854	47.162	46.783	54.046	53.967	54.048	51.846	ICBI
	RMSE	0.002	0.002	0.002	0.002	**0.002**	0.003	0.002	0.004	0.004	0.004	0.005	0.002	0.002	0.002	0.003	ICBI
	SSIM	0.994	0.996	0.996	0.995	**0.997**	0.992	0.996	0.977	0.978	0.976	0.975	0.996	0.996	0.996	0.993	ICBI
butterfly	FSIM	0.668	**0.738**	0.724	0.718	0.722	0.664	0.715	0.669	0.698	0.636	0.637	0.703	0.721	0.687	0.718	EDSR
	PSNR	44.751	47.85	47.942	47.676	**50.491**	44.597	47.656	43.115	44.241	42.66	42.043	47.329	47.284	47.19	45.232	ICBI
	RMSE	0.006	0.004	0.004	0.004	**0.003**	0.006	0.004	0.007	0.006	0.007	0.008	0.004	0.004	0.004	0.005	ICBI
	SSIM	0.973	0.987	0.987	0.986	**0.992**	0.971	0.986	0.958	0.968	0.955	0.949	0.985	0.984	0.984	0.976	ICBI
head	FSIM	0.629	0.683	0.686	0.679	**0.7**	0.647	0.676	0.618	0.656	0.594	0.574	0.682	0.695	0.669	0.677	ICBI
	PSNR	51.084	52.969	53.178	53.038	**53.932**	52.296	53.143	49.564	50.992	48.779	48.114	52.864	53.413	52.637	51.857	ICBI
	RMSE	0.003	0.002	0.002	0.002	**0.002**	0.002	0.002	0.003	0.003	0.004	0.004	0.002	0.002	0.002	0.003	ICBI
	SSIM	0.991	0.994	0.994	0.994	**0.995**	0.993	0.994	0.986	0.99	0.984	0.982	0.994	0.995	0.994	0.993	ICBI
woman	FSIM	0.716	0.765	0.763	0.758	**0.8**	0.731	0.755	0.717	0.745	0.691	0.682	0.753	0.759	0.744	0.755	ICBI
	PSNR	48.242	50.896	51.048	50.742	**55.212**	49.637	50.856	46.934	48.323	46.321	45.621	50.874	51.02	50.749	48.967	ICBI
	RMSE	0.004	0.003	0.003	0.003	**0.002**	0.003	0.003	0.005	0.004	0.005	0.005	0.003	0.003	0.003	0.004	ICBI
	SSIM	0.987	0.993	0.993	0.992	**0.997**	0.99	0.993	0.982	0.987	0.979	0.976	0.992	0.993	0.992	0.989	ICBI

Table IX-3: IQA metrics of Set5, scale factor 3

image_name	iqa	DRCN	EDSR	ESPCN	FSRCNN	ICBI	INEDI	LapSRN	VDSR_bicubic	VDSR_bilinear	VDSR_lanczos	VDSR_nearest	bicubic	bilinear	lanczos	nearest	best
baby	FSIM	0.638	0.635	0.634	0.63				0.613	0.627	0.59	0.534	**0.641**	0.639	0.638	0.545	bicubic
	PSNR	51.352	51.679	51.743	51.59				47.317	47.995	46.626	46.451	51.622	**51.883**	51.498	50.166	bilinear
	RMSE	0.003	0.003	0.003	0.003				0.004	0.004	0.005	0.005	0.003	**0.003**	0.003	0.003	bilinear
	SSIM	0.993	0.993	0.993	0.993				0.985	0.988	0.982	0.981	0.993	**0.993**	0.993	0.99	bilinear
bird	FSIM	0.678	**0.692**	0.682	0.682				0.635	0.653	0.613	0.554	0.682	0.686	0.677	0.579	EDSR
	PSNR	48.986	49.31	49.309	49.262				45.234	46.1	44.702	44.698	49.284	**49.483**	49.184	47.681	bilinear
	RMSE	0.004	0.003	0.003	0.003				0.005	0.005	0.006	0.006	0.003	**0.003**	0.003	0.004	bilinear
	SSIM	0.986	0.987	0.986	0.986				0.965	0.969	0.962	0.962	0.986	**0.987**	0.986	0.982	bilinear
butterfly	FSIM	0.601	**0.639**	0.633	0.637				0.556	0.609	0.505	0.514	0.613	0.627	0.598	0.507	EDSR
	PSNR	42.546	43.367	43.601	**43.702**				40.2	41.598	39.485	39.611	43.232	43.472	43.058	41.62	FSRCNN
	RMSE	0.007	0.007	0.007	**0.007**				0.01	0.008	0.011	0.01	0.007	0.007	0.007	0.008	FSRCNN
	SSIM	0.955	0.963	0.964	**0.965**				0.922	0.942	0.909	0.913	0.961	0.962	0.959	0.945	FSRCNN
head	FSIM	0.593	0.593	0.594	0.584				0.556	0.583	0.535	0.507	0.597	**0.603**	0.589	0.521	bilinear
	PSNR	50.279	50.808	50.935	50.728				48.068	49.389	47.433	47.553	50.543	**50.971**	50.35	49.549	bilinear
	RMSE	0.003	0.003	0.003	0.003				0.004	0.003	0.004	0.004	0.003	**0.003**	0.003	0.003	bilinear
	SSIM	0.989	0.99	0.99	0.989				0.98	0.985	0.978	0.979	0.989	**0.99**	0.989	0.987	bilinear
woman	FSIM	0.646	**0.663**	0.654	0.652				0.618	0.647	0.584	0.55	0.652	0.657	0.644	0.549	EDSR
	PSNR	46.117	46.546	46.74	46.645				43.567	45.471	42.656	43.186	46.627	**46.946**	46.452	45.105	bilinear
	RMSE	0.005	0.005	0.005	0.005				0.007	0.005	0.007	0.007	0.005	**0.004**	0.005	0.006	bilinear
	SSIM	0.978	0.98	0.981	0.98				0.962	0.974	0.954	0.958	0.98	**0.981**	0.979	0.973	bilinear

Table IX-4: IQA metrics of Set5, scale factor 4

image_name	iqa	DRCN	EDSR	ESPCN	FSRCNN	ICBI	INEDI	LapSRN	VDSR_bicubic	VDSR_bilinear	VDSR_lanczos	VDSR_nearest	bicubic	bilinear	lanczos	nearest	best
baby	FSIM	0.56	0.559	0.557	0.552	**0.646**	0.536	0.557	0.545	0.555	0.544	0.447	0.564	0.563	0.563	0.442	ICBI
	PSNR	49.331	49.417	49.526	49.377	**53.389**	48.295	49.433	46.397	47.152	46.324	45.052	49.338	49.783	49.195	48.028	ICBI
	RMSE	0.003	0.003	0.003	0.003	**0.002**	0.004	0.003	0.005	0.004	0.005	0.006	0.003	0.003	0.003	0.004	ICBI
	SSIM	0.989	0.989	0.989	0.989	**0.995**	0.986	0.989	0.981	0.985	0.981	0.974	0.989	0.99	0.988	0.985	ICBI
bird	FSIM	0.604	0.61	0.605	0.602	**0.686**	0.572	0.608	0.567	0.586	0.563	0.453	0.605	0.612	0.602	0.464	ICBI
	PSNR	46.686	46.611	46.807	46.742	**50.731**	45.579	46.773	44.224	44.878	44.185	43.114	46.687	47.074	46.585	45.357	ICBI
	RMSE	0.005	0.005	0.005	0.005	**0.003**	0.005	0.005	0.006	0.006	0.006	0.007	0.005	0.004	0.005	0.005	ICBI
	SSIM	0.975	0.975	0.975	0.975	**0.991**	0.967	0.975	0.954	0.959	0.954	0.947	0.975	0.977	0.975	0.968	ICBI
butterfly	FSIM	0.561	0.576	0.579	0.576	**0.621**	0.498	0.575	0.526	0.561	0.5	0.447	0.561	0.568	0.553	0.409	ICBI
	PSNR	41.216	41.148	41.708	41.574	**44.269**	38.67	41.493	39.274	40.547	38.981	38.248	41.265	41.68	41.097	39.838	ICBI
	RMSE	0.009	0.009	0.008	0.008	**0.006**	0.012	0.008	0.011	0.009	0.011	0.012	0.009	0.008	0.009	0.01	ICBI
	SSIM	0.938	0.939	0.945	0.943	**0.969**	0.894	0.942	0.903	0.926	0.896	0.884	0.939	0.943	0.937	0.918	ICBI
head	FSIM	0.535	0.527	0.528	0.521	**0.584**	0.503	0.527	0.518	0.531	0.513	0.461	0.536	0.54	0.534	0.448	ICBI
	PSNR	49.256	49.526	49.691	49.574	**51.489**	48.496	49.668	47.69	48.663	47.511	46.485	49.252	49.855	49.068	48.28	ICBI
	RMSE	0.003	0.003	0.003	0.003	**0.003**	0.004	0.003	0.004	0.004	0.004	0.005	0.003	0.003	0.004	0.004	ICBI
	SSIM	0.985	0.986	0.986	0.986	**0.991**	0.981	0.986	0.977	0.982	0.977	0.972	0.985	0.987	0.984	0.982	ICBI
woman	FSIM	0.589	0.599	0.589	0.58	**0.666**	0.559	0.587	0.563	0.592	0.552	0.467	0.589	0.597	0.584	0.443	ICBI
	PSNR	44.383	44.413	44.697	44.486	**48.238**	43.393	44.534	42.314	44.098	42.177	41.514	44.444	44.982	44.269	43.081	ICBI
	RMSE	0.006	0.006	0.006	0.006	**0.004**	0.007	0.006	0.008	0.006	0.008	0.008	0.006	0.006	0.006	0.007	ICBI
	SSIM	0.967	0.968	0.969	0.968	**0.987**	0.96	0.968	0.95	0.965	0.948	0.94	0.968	0.971	0.967	0.957	ICBI

Table IX-5: IQA metrics of Set5, scale factor 8

image_name	iqa	DRCN	EDSR	ESPCN	FSRCNN	ICBI	INEDI	LapSRN	VDSR_bicubic	VDSR_bilinear	VDSR_lanczos	VDSR_nearest	bicubic	bilinear	lanczos	nearest	best
baby	FSIM					**0.515**	0.438		0.453	0.445	0.453	0.353	0.454	0.448	0.453	0.305	ICBI
	PSNR					**48.518**	44.233		44.188	44.546	44.089	43.021	45.059	45.553	44.929	43.935	ICBI
	RMSE					**0.004**	0.006		0.006	0.006	0.006	0.007	0.006	0.005	0.006	0.006	ICBI
	SSIM					**0.985**	0.968		0.969	0.973	0.968	0.96	0.973	0.976	0.972	0.965	ICBI
bird	FSIM					**0.542**	0.435		0.445	0.44	0.443	0.344	0.46	0.456	0.458	0.315	ICBI
	PSNR					**45.503**	41.635		41.813	42.109	41.713	40.824	42.541	43.046	42.399	41.533	ICBI
	RMSE					**0.005**	0.008		0.008	0.008	0.008	0.009	0.007	0.007	0.008	0.008	ICBI
	SSIM					**0.969**	0.92		0.919	0.925	0.918	0.907	0.933	0.94	0.931	0.922	ICBI
butterfly	FSIM					**0.47**	0.384		0.422	0.422	0.416	0.364	0.427	0.427	0.42	0.28	ICBI
	PSNR					**39.162**	35.735		37.307	37.767	37.123	36.076	37.599	38.129	37.394	36.504	ICBI
	RMSE					**0.011**	0.016		0.014	0.013	0.014	0.016	0.013	0.012	0.013	0.015	ICBI
	SSIM					**0.908**	0.82		0.856	0.872	0.849	0.822	0.872	0.883	0.866	0.842	ICBI
head	FSIM					**0.478**	0.436		0.437	0.445	0.426	0.37	0.437	0.45	0.425	0.345	ICBI
	PSNR					**49.067**	45.43		45.781	46.227	45.648	44.692	46.343	46.907	46.185	45.447	ICBI
	RMSE					**0.004**	0.005		0.005	0.005	0.005	0.006	0.005	0.005	0.005	0.005	ICBI
	SSIM					**0.982**	0.96		0.961	0.967	0.96	0.955	0.967	0.972	0.965	0.962	ICBI
woman	FSIM					**0.547**	0.426		0.454	0.455	0.449	0.366	0.459	0.461	0.453	0.298	ICBI
	PSNR					**43.664**	39.844		40.261	40.822	40.111	38.993	40.579	41.184	40.411	39.446	ICBI
	RMSE					**0.007**	0.01		0.01	0.009	0.01	0.011	0.009	0.009	0.01	0.011	ICBI
	SSIM					**0.963**	0.918		0.922	0.931	0.919	0.901	0.928	0.936	0.926	0.911	ICBI

Table IX-6: IQA metrics of Set14, scale factor 2

image_name	iqa	DRCN	EDSR	ESPCN	FSRCNN	ICBI	INEDI	LapSRN	VDSR_bicubic	VDSR_bilinear	VDSR_lanczos	VDSR_nearest	bicubic	bilinear	lanczos	nearest	best
baboon	FSIM	0.593	0.636	0.641	0.637	**0.654**	0.597	0.639	0.592	0.621	0.58	0.552	0.634	0.636	0.624	0.631	ICBI
	PSNR	42.062	44.179	44.486	44.314	44.627	43.315	44.487	41.279	42.767	41.219	39.673	44.24	**44.81**	43.969	43.46	bilinear
	RMSE	0.008	0.006	0.006	0.006	0.006	0.007	0.006	0.009	0.007	0.009	0.01	0.006	**0.006**	0.006	0.007	bilinear
	SSIM	0.944	0.964	0.966	0.965	0.967	0.955	0.966	0.924	0.941	0.923	0.899	0.964	**0.968**	0.962	0.958	bilinear
barbara	FSIM	0.675	0.714	0.707	0.701	**0.72**	0.668	0.705	0.667	0.688	0.642	0.625	0.704	0.711	0.695	0.707	ICBI
	PSNR	44.911	47.49	47.699	47.385	48.083	46.827	47.889	44.208	45.87	43.648	42.731	47.829	**48.446**	47.493	46.88	bilinear
	RMSE	0.006	0.004	0.004	0.004	0.004	0.005	0.004	0.006	0.005	0.007	0.007	0.004	**0.004**	0.004	0.005	bilinear
	SSIM	0.971	0.983	0.983	0.982	0.985	0.98	0.984	0.962	0.973	0.958	0.95	0.984	**0.986**	0.983	0.98	bilinear
bridge	FSIM	0.627	0.685	0.679	0.671	**0.71**	0.623	0.674	0.634	0.671	0.605	0.58	0.676	0.673	0.662	0.679	ICBI
	PSNR	45.386	48.084	48.094	47.849	**49.128**	46.509	48.029	45.322	47.486	44.934	43.03	47.963	48.379	47.726	46.789	ICBI
	RMSE	0.005	0.004	0.004	0.004	**0.003**	0.005	0.004	0.005	0.004	0.006	0.007	0.004	0.004	0.004	0.005	ICBI
	SSIM	0.972	0.985	0.985	0.984	**0.988**	0.978	0.984	0.972	0.982	0.969	0.953	0.984	0.985	0.983	0.979	ICBI
coastguard	FSIM	0.626	0.673	0.676	0.67	**0.719**	0.596	0.667	0.639	0.655	0.597	0.561	0.669	0.654	0.647	0.682	ICBI
	PSNR	46.643	49.327	49.599	49.373	**51.477**	47.002	49.553	46.646	48.527	45.918	44.44	49.606	49.854	49.359	48.204	ICBI
	RMSE	0.005	0.003	0.003	0.003	**0.003**	0.004	0.003	0.005	0.004	0.005	0.006	0.003	0.003	0.003	0.004	ICBI
	SSIM	0.979	0.988	0.989	0.988	**0.993**	0.98	0.989	0.978	0.986	0.974	0.965	0.989	0.99	0.988	0.985	ICBI
comic	FSIM	0.651	0.723	0.715	0.708	**0.732**	0.666	0.708	0.654	0.69	0.63	0.612	0.707	0.712	0.693	0.707	ICBI
	PSNR	43.164	46.033	46.306	46.033	**47.582**	44.498	46.179	42.801	44.624	42.486	41.038	46.091	46.297	45.868	44.518	ICBI
	RMSE	0.007	0.005	0.005	0.005	**0.004**	0.006	0.005	0.007	0.006	0.008	0.009	0.005	0.005	0.005	0.006	ICBI
	SSIM	0.959	0.978	0.979	0.977	**0.984**	0.968	0.978	0.955	0.97	0.952	0.936	0.978	0.978	0.977	0.969	ICBI
face	FSIM	0.633	0.687	0.689	0.682	**0.702**	0.652	0.678	0.621	0.658	0.598	0.577	0.686	0.698	0.672	0.681	ICBI
	PSNR	51.048	52.939	53.142	53.002	**53.919**	52.255	53.106	49.532	50.965	48.747	48.082	52.829	53.377	52.603	51.821	ICBI
	RMSE	0.003	0.002	0.002	0.002	**0.002**	0.002	0.002	0.003	0.003	0.004	0.004	0.002	0.002	0.002	0.003	ICBI
	SSIM	0.991	0.994	0.994	0.994	**0.995**	0.993	0.994	0.986	0.99	0.984	0.982	0.994	0.995	0.994	0.993	ICBI
flowers	FSIM	0.669	**0.728**	0.722	0.716	0.723	0.66	0.718	0.663	0.695	0.639	0.62	0.716	0.721	0.703	0.714	EDSR
	PSNR	47.318	49.83	49.806	49.602	**50.915**	47.549	49.745	45.387	46.526	44.944	43.976	49.633	49.785	49.462	47.949	ICBI
	RMSE	0.004	0.003	0.003	0.003	**0.003**	0.004	0.003	0.005	0.005	0.006	0.006	0.003	0.003	0.003	0.004	ICBI
	SSIM	0.983	0.99	0.99	0.99	**0.992**	0.983	0.99	0.971	0.977	0.968	0.962	0.99	0.99	0.989	0.985	ICBI
foreman	FSIM	0.728	0.776	0.768	0.766	**0.781**	0.706	0.766	0.723	0.75	0.688	0.682	0.762	0.766	0.754	0.768	ICBI
	PSNR	49.644	52.12	52.012	51.841	**53.299**	49.764	51.918	48.905	50.402	48.05	47.819	51.941	51.917	51.893	50.562	ICBI
	RMSE	0.003	0.002	0.003	0.003	**0.002**	0.003	0.003	0.004	0.003	0.004	0.004	0.003	0.003	0.003	0.003	ICBI
	SSIM	0.993	0.996	0.996	0.996	**0.998**	0.993	0.996	0.991	0.994	0.989	0.988	0.996	0.996	0.996	0.994	ICBI
lenna	FSIM	0.654	0.707	0.705	0.698	**0.718**	0.665	0.701	0.638	0.679	0.598	0.587	0.698	0.708	0.687	0.699	ICBI
	PSNR	50.948	53.137	53.241	52.986	**54.729**	52.435	53.202	46.546	47.352	46.147	45.48	53.053	53.416	52.872	51.653	ICBI
	RMSE	0.003	0.002	0.002	0.002	**0.002**	0.002	0.002	0.005	0.004	0.005	0.005	0.002	0.002	0.002	0.003	ICBI
	SSIM	0.992	0.995	0.995	0.995	**0.996**	0.994	0.995	0.98	0.984	0.978	0.974	0.995	0.995	0.995	0.993	ICBI
man	FSIM	0.637	0.701	0.699	0.691	**0.732**	0.665	0.695	0.646	0.681	0.613	0.595	0.69	0.697	0.68	0.685	ICBI
	PSNR	46.534	49.302	49.399	49.131	**50.797**	48.416	49.318	46.631	48.715	46.012	44.725	49.239	49.621	49.023	47.986	ICBI
	RMSE	0.005	0.003	0.003	0.003	**0.003**	0.004	0.003	0.005	0.004	0.005	0.006	0.003	0.003	0.004	0.004	ICBI
	SSIM	0.978	0.988	0.988	0.987	**0.991**	0.985	0.988	0.978	0.986	0.975	0.968	0.987	0.988	0.987	0.984	ICBI
monarch	FSIM	0.726	**0.776**	0.768	0.764	0.762	0.705	0.762	0.715	0.743	0.679	0.666	0.756	0.767	0.746	0.758	EDSR
	PSNR	49.96	52.82	52.964	52.698	**55.479**	49.647	52.79	46.821	47.647	46.406	45.913	52.43	52.438	52.309	50.374	ICBI
	RMSE	0.003	0.002	0.002	0.002	**0.002**	0.003	0.002	0.005	0.004	0.005	0.005	0.002	0.002	0.002	0.003	ICBI
	SSIM	0.992	0.995	0.996	0.995	**0.997**	0.991	0.995	0.979	0.983	0.978	0.976	0.995	0.995	0.995	0.992	ICBI
pepper	FSIM	0.663	0.712	0.709	0.704	0.701	0.661	0.707	0.652	0.682	0.623	0.624	0.705	**0.715**	0.692	0.703	bilinear
	PSNR	49.757	51.534	51.577	51.431	**52.879**	50.389	51.679	45.53	45.943	45.199	45.038	51.461	51.807	51.342	50.07	ICBI
	RMSE	0.003	0.003	0.003	0.003	**0.002**	0.003	0.003	0.005	0.005	0.005	0.006	0.003	0.003	0.003	0.003	ICBI
	SSIM	0.992	0.994	0.994	0.994	**0.995**	0.992	0.994	0.96	0.962	0.958	0.958	0.994	0.995	0.994	0.992	ICBI
ppt3	FSIM	0.668	0.74	0.725	0.714	0.719	0.67	0.71	0.665	0.689	0.634	0.644	0.728	0.735	0.685	**0.747**	nearest
	PSNR	44.817	46.641	46.685	46.349	**49.244**	45.591	46.549	43.716	44.541	43.498	42.825	46.48	46.458	46.414	44.718	ICBI
	RMSE	0.006	0.005	0.005	0.005	**0.003**	0.005	0.005	0.007	0.006	0.007	0.007	0.005	0.005	0.005	0.006	ICBI
	SSIM	0.977	0.984	0.984	0.983	**0.991**	0.98	0.984	0.968	0.972	0.966	0.962	0.983	0.983	0.983	0.977	ICBI
zebra	FSIM	0.705	0.751	0.747	0.743	**0.785**	0.693	0.743	0.708	0.733	0.67	0.667	0.743	0.735	0.737	0.74	ICBI
	PSNR	47.039	49.179	49.142	48.88	**53.282**	47.127	49.179	45.45	46.538	44.474	44.188	49.204	49.183	49.134	47.218	ICBI
	RMSE	0.004	0.003	0.003	0.004	**0.002**	0.004	0.003	0.005	0.005	0.006	0.006	0.003	0.003	0.003	0.004	ICBI
	SSIM	0.982	0.988	0.988	0.988	**0.995**	0.981	0.989	0.975	0.98	0.968	0.967	0.988	0.988	0.988	0.983	ICBI

Table IX-7: IQA metrics of Set14, scale factor 3

image_name	iqa	DRCN	EDSR	ESPCN	FSRCNN	ICBI	INEDI	LapSRN	VDSR_bicubic	VDSR_bilinear	VDSR_lanczos	VDSR_nearest	bicubic	bilinear	lanczos	nearest	best
baboon	FSIM	0.535	0.52	0.518	0.517				0.515	0.524	0.503	0.481	0.534	0.523	**0.536**	0.455	lanczos
	PSNR	42.086	42.66	**42.874**	42.738				39.62	41.513	39.308	39.812	42.314	42.79	42.1	41.601	ESPCN
	RMSE	0.008	0.007	**0.007**	0.007				0.01	0.008	0.011	0.01	0.008	0.007	0.008	0.008	ESPCN
	SSIM	0.941	0.947	**0.95**	0.948				0.892	0.923	0.885	0.896	0.943	0.949	0.941	0.935	ESPCN
barbara	FSIM	0.586	0.583	0.584	0.582				0.566	0.577	0.546	0.506	**0.589**	0.588	0.587	0.506	bicubic
	PSNR	45.151	44.993	45.598	45.432				42.242	44.597	41.767	42.652	45.419	**45.851**	45.241	44.502	bilinear
	RMSE	0.006	0.006	0.005	0.005				0.008	0.006	0.008	0.007	0.005	**0.005**	0.005	0.006	bilinear
	SSIM	0.97	0.969	0.973	0.972				0.942	0.964	0.937	0.946	0.972	**0.974**	0.971	0.966	bilinear
bridge	FSIM	0.556	0.549	0.543	0.539				0.53	0.55	0.507	0.492	**0.558**	0.551	0.558	0.486	bicubic
	PSNR	44.825	45.402	**45.64**	45.539				42.811	45.059	42.06	42.635	45.211	45.607	45.006	44.221	ESPCN
	RMSE	0.006	0.005	**0.005**	0.005				0.007	0.006	0.008	0.007	0.005	0.005	0.006	0.006	ESPCN
	SSIM	0.968	0.972	**0.973**	0.973				0.951	0.97	0.942	0.948	0.971	0.973	0.969	0.964	ESPCN
coastguard	FSIM	0.521	0.497	0.501	0.501				0.523	0.517	0.514	0.476	0.531	0.513	**0.537**	0.473	lanczos
	PSNR	45.865	46.675	46.74	**46.752**				44.137	46.081	43.582	44.012	46.329	46.745	46.118	45.414	FSRCNN
	RMSE	0.005	0.005	0.005	**0.005**				0.006	0.005	0.007	0.006	0.005	0.005	0.005	0.005	FSRCNN
	SSIM	0.975	0.979	**0.979**	0.979				0.963	0.976	0.958	0.962	0.977	0.979	0.976	0.973	ESPCN
comic	FSIM	0.585	0.586	0.59	0.589				0.541	0.585	0.513	0.499	**0.595**	0.594	0.586	0.502	bicubic
	PSNR	42.118	42.471	42.87	42.853				39.875	41.964	39.357	39.758	42.538	**42.891**	42.346	41.343	bilinear
	RMSE	0.008	0.008	0.007	0.007				0.01	0.008	0.011	0.01	0.007	**0.007**	0.008	0.009	bilinear
	SSIM	0.947	0.951	0.954	0.954				0.915	0.945	0.905	0.913	0.951	**0.954**	0.949	0.937	bilinear
face	FSIM	0.589	0.591	0.592	0.582				0.554	0.575	0.532	0.5	0.596	**0.601**	0.588	0.516	bilinear
	PSNR	50.257	50.736	50.895	50.724				48.118	49.344	47.452	47.523	50.519	**50.953**	50.324	49.53	bilinear
	RMSE	0.003	0.003	0.003	0.003				0.004	0.003	0.004	0.004	0.003	**0.003**	0.003	0.003	bilinear
	SSIM	0.989	0.99	0.99	0.989				0.98	0.985	0.978	0.978	0.989	**0.99**	0.989	0.987	bilinear
flowers	FSIM	0.592	**0.606**	0.601	0.601				0.555	0.586	0.531	0.504	0.6	0.602	0.592	0.509	EDSR
	PSNR	45.439	45.881	46.137	46.092				42.829	44.314	42.215	42.405	45.871	**46.182**	45.692	44.554	bilinear
	RMSE	0.005	0.005	0.005	0.005				0.007	0.006	0.008	0.008	0.005	**0.005**	0.005	0.006	bilinear
	SSIM	0.973	0.976	0.977	0.976				0.95	0.963	0.944	0.946	0.976	**0.977**	0.975	0.968	bilinear
foreman	FSIM	0.649	**0.663**	0.655	0.657				0.617	0.651	0.589	0.546	0.658	0.662	0.653	0.537	EDSR
	PSNR	48.348	48.413	48.708	48.665				45.828	47.844	45.116	45.376	48.894	**49.002**	48.831	47.31	bilinear
	RMSE	0.004	0.004	0.004	0.004				0.005	0.004	0.006	0.005	0.004	**0.004**	0.004	0.004	bilinear
	SSIM	0.99	0.991	0.991	0.991				0.982	0.989	0.978	0.978	0.991	**0.992**	0.991	0.987	bilinear
lenna	FSIM	0.589	0.595	0.591	0.589				0.555	0.584	0.532	0.5	0.595	**0.6**	0.589	0.517	bilinear
	PSNR	49.383	49.744	49.911	49.76				45.006	46.193	44.499	44.761	49.732	**50.118**	49.558	48.529	bilinear
	RMSE	0.003	0.003	0.003	0.003				0.006	0.005	0.006	0.006	0.003	**0.003**	0.003	0.004	bilinear
	SSIM	0.988	0.989	0.99	0.989				0.971	0.978	0.968	0.969	0.989	**0.99**	0.989	0.986	bilinear
man	FSIM	0.578	0.587	0.586	0.583				0.548	0.582	0.52	0.493	0.587	**0.588**	0.581	0.501	bilinear
	PSNR	45.769	46.388	46.567	46.463				43.845	46.055	43.038	43.654	46.27	**46.639**	46.097	45.142	bilinear
	RMSE	0.005	0.005	0.005	0.005				0.006	0.005	0.007	0.007	0.005	**0.005**	0.005	0.006	bilinear
	SSIM	0.972	0.975	0.976	0.975				0.959	0.973	0.952	0.957	0.974	**0.976**	0.973	0.968	bilinear
monarch	FSIM	0.661	**0.684**	0.673	0.674				0.635	0.657	0.612	0.551	0.664	0.671	0.656	0.558	EDSR
	PSNR	47.635	48.36	48.541	**48.571**				44.568	45.7	43.954	43.884	48.212	48.468	48.047	46.63	FSRCN
	RMSE	0.004	0.004	0.004	**0.004**				0.006	0.005	0.006	0.006	0.004	0.004	0.004	0.005	FSRCN
	SSIM	0.985	0.988	0.988	**0.988**				0.968	0.974	0.964	0.964	0.987	0.988	0.987	0.982	FSRCN
pepper	FSIM	0.609	0.624	0.618	0.618				0.572	0.601	0.545	0.515	0.617	**0.625**	0.607	0.531	bilinear
	PSNR	47.788	48.218	48.266	48.259				44.13	44.699	43.601	43.758	48.155	**48.408**	48.045	46.992	bilinear
	RMSE	0.004	0.004	0.004	0.004				0.006	0.006	0.007	0.006	0.004	**0.004**	0.004	0.004	bilinear
	SSIM	0.987	0.988	0.988	0.988				0.951	0.956	0.948	0.95	0.988	**0.988**	0.987	0.984	bilinear
ppt3	FSIM	0.589	**0.625**	0.595	0.598				0.562	0.584	0.535	0.528	0.612	0.624	0.579	0.565	EDSR
	PSNR	41.717	42.179	42.442	42.479				40.483	41.465	40.266	40.077	42.395	**42.593**	42.311	41.178	bilinear
	RMSE	0.008	0.008	0.008	0.008				0.009	0.008	0.01	0.01	0.008	**0.007**	0.008	0.009	bilinear
	SSIM	0.954	0.958	0.959	0.959				0.937	0.947	0.934	0.933	0.958	**0.959**	0.958	0.949	bilinear
zebra	FSIM	0.626	0.629	0.616	0.614				0.601	0.62	0.578	0.529	**0.631**	0.623	0.631	0.537	bicubic
	PSNR	44.138	44.61	44.702	44.732				42.068	43.395	41.504	41.399	44.715	**44.898**	44.609	43.218	bilinear
	RMSE	0.006	0.006	0.006	0.006				0.008	0.007	0.008	0.009	0.006	**0.006**	0.006	0.007	bilinear
	SSIM	0.964	0.967	0.968	0.968				0.945	0.958	0.938	0.937	0.968	**0.968**	0.967	0.957	bilinear

Table IX-8: IQA metrics of Set14, scale factor 4

image_name	iqa	DRCN	EDSR	ESPCN	FSRCNN	ICBI	INEDI	LapSRN	VDSR_bicubic	VDSR_bilinear	VDSR_lanczos	VDSR_nearest	bicubic	bilinear	lanczos	nearest	best
baboon	FSIM	0.466	0.451	0.446	0.442	**0.492**	0.427	0.445	0.477	0.448	0.477	0.442	0.47	0.447	0.479	0.383	ICBI
	PSNR	41.565	41.967	42.171	42.068	42.105	41.113	42.139	39.244	41.287	39.279	39.338	41.524	**42.205**	41.305	40.834	bilinear
	RMSE	0.008	0.008	0.008	0.008	0.008	0.009	0.008	0.011	0.009	0.011	0.011	0.008	**0.008**	0.009	0.009	bilinear
	SSIM	0.932	0.937	0.94	0.939	0.94	0.923	0.94	0.88	0.918	0.881	0.884	0.931	**0.941**	0.928	0.922	bilinear
barbara	FSIM	0.521	0.518	0.522	0.519	**0.569**	0.493	0.522	0.507	0.516	0.506	0.443	0.52	0.521	0.52	0.424	ICBI
	PSNR	44.312	43.946	44.763	44.606	**45.46**	43.658	44.642	41.465	44.146	41.667	41.74	44.314	45.019	44.004	43.384	ICBI
	RMSE	0.006	0.006	0.006	0.006	**0.005**	0.007	0.006	0.008	0.006	0.008	0.008	0.006	0.006	0.006	0.007	ICBI
	SSIM	0.964	0.961	0.967	0.966	**0.973**	0.959	0.967	0.933	0.96	0.935	0.936	0.964	0.969	0.962	0.956	ICBI
bridge	FSIM	0.483	0.463	0.463	0.458	**0.527**	0.425	0.461	0.482	0.472	0.483	0.434	0.482	0.469	0.489	0.399	ICBI
	PSNR	43.672	43.921	44.231	44.101	**45.107**	42.316	44.158	41.946	44.124	41.935	41.53	43.685	44.293	43.492	42.746	ICBI
	RMSE	0.007	0.006	0.006	0.006	**0.006**	0.008	0.006	0.008	0.006	0.008	0.008	0.007	0.006	0.007	0.007	ICBI
	SSIM	0.96	0.962	0.964	0.963	**0.97**	0.947	0.964	0.941	0.963	0.941	0.935	0.96	0.965	0.958	0.951	ICBI
coastguard	FSIM	0.441	0.421	0.41	0.412	0.443	0.389	0.402	0.448	0.427	0.45	0.43	0.442	0.423	**0.454**	0.389	lanczos
	PSNR	45.009	45.071	45.391	45.413	45.503	43.694	45.251	43.556	45.359	43.611	43.424	45.039	**45.553**	44.904	44.287	bilinear
	RMSE	0.006	0.006	0.005	0.005	0.005	0.007	0.005	0.007	0.005	0.007	0.007	0.006	**0.005**	0.006	0.006	bilinear
	SSIM	0.97	0.971	0.973	0.973	**0.975**	0.959	0.972	0.958	0.972	0.958	0.957	0.97	0.974	0.969	0.966	ICBI
comic	FSIM	0.523	0.505	0.513	0.509	**0.579**	0.476	0.508	0.499	0.519	0.493	0.434	0.524	0.522	0.522	0.409	ICBI
	PSNR	40.908	40.927	41.34	41.179	**42.499**	39.773	41.24	39.004	41.012	38.945	38.611	40.918	41.501	40.717	39.858	ICBI
	RMSE	0.009	0.009	0.009	0.009	**0.007**	0.01	0.009	0.011	0.009	0.011	0.012	0.009	0.008	0.009	0.01	ICBI
	SSIM	0.932	0.932	0.937	0.935	**0.952**	0.915	0.936	0.899	0.933	0.897	0.89	0.932	0.939	0.929	0.915	ICBI
face	FSIM	0.543	0.533	0.533	0.525	**0.591**	0.511	0.531	0.522	0.537	0.516	0.458	0.544	0.549	0.541	0.449	ICBI
	PSNR	49.219	49.448	49.608	49.523	**51.446**	48.578	49.56	47.606	48.638	47.475	46.456	49.209	49.806	49.029	48.252	ICBI
	RMSE	0.003	0.003	0.003	0.003	**0.003**	0.004	0.003	0.004	0.004	0.004	0.005	0.003	0.003	0.004	0.004	ICBI
	SSIM	0.985	0.985	0.986	0.986	**0.991**	0.981	0.986	0.977	0.982	0.977	0.972	0.985	0.987	0.984	0.982	ICBI
flowers	FSIM	0.529	0.523	0.526	0.526	**0.595**	0.491	0.525	0.507	0.525	0.503	0.439	0.53	0.532	0.529	0.415	ICBI
	PSNR	44.068	43.996	44.403	44.298	**46.548**	42.696	44.349	41.968	43.386	41.87	41.249	44.07	44.599	43.895	42.932	ICBI
	RMSE	0.006	0.006	0.006	0.006	**0.005**	0.007	0.006	0.008	0.007	0.008	0.009	0.006	0.006	0.006	0.007	ICBI
	SSIM	0.964	0.963	0.966	0.965	**0.978**	0.952	0.965	0.94	0.955	0.939	0.932	0.964	0.967	0.962	0.954	ICBI
foreman	FSIM	0.594	0.603	0.592	0.592	**0.659**	0.531	0.597	0.585	0.605	0.571	0.472	0.599	0.611	0.598	0.445	ICBI
	PSNR	46.588	46.487	46.685	46.898	**49.859**	44.563	46.998	45.041	46.538	44.626	43.79	46.804	47.103	46.698	45.28	ICBI
	RMSE	0.005	0.005	0.005	0.005	**0.003**	0.006	0.004	0.006	0.005	0.006	0.006	0.005	0.004	0.005	0.005	ICBI
	SSIM	0.986	0.986	0.987	0.987	**0.993**	0.976	0.987	0.979	0.986	0.977	0.97	0.987	0.988	0.986	0.981	ICBI
lenna	FSIM	0.526	0.515	0.518	0.514	**0.604**	0.512	0.515	0.506	0.526	0.505	0.436	0.527	0.532	0.525	0.431	ICBI
	PSNR	47.918	47.847	48.098	47.961	**51.015**	47.8	48.017	44.378	45.567	44.346	43.676	47.921	48.514	47.736	46.802	ICBI
	RMSE	0.004	0.004	0.004	0.004	**0.003**	0.004	0.004	0.006	0.005	0.006	0.007	0.004	0.004	0.004	0.005	ICBI
	SSIM	0.984	0.984	0.985	0.984	**0.992**	0.984	0.984	0.966	0.975	0.966	0.961	0.984	0.986	0.983	0.98	ICBI
man	FSIM	0.518	0.511	0.514	0.512	**0.589**	0.489	0.515	0.504	0.522	0.498	0.435	0.52	0.523	0.52	0.413	ICBI
	PSNR	44.751	44.787	45.077	44.987	**46.86**	44.063	45.011	43.151	45.167	42.999	42.527	44.791	45.336	44.622	43.757	ICBI
	RMSE	0.006	0.006	0.006	0.006	**0.005**	0.006	0.006	0.007	0.006	0.007	0.007	0.006	0.005	0.006	0.006	ICBI
	SSIM	0.963	0.964	0.965	0.965	**0.978**	0.956	0.965	0.95	0.966	0.949	0.944	0.963	0.967	0.962	0.956	ICBI
monarch	FSIM	0.603	0.621	0.609	0.606	**0.674**	0.556	0.613	0.583	0.6	0.575	0.469	0.602	0.61	0.596	0.456	ICBI
	PSNR	46.141	46.133	46.56	46.427	**49.612**	43.858	46.428	43.65	44.735	43.48	42.52	46.177	46.611	46.017	44.785	ICBI
	RMSE	0.005	0.005	0.005	0.005	**0.003**	0.006	0.005	0.007	0.006	0.007	0.007	0.005	0.005	0.005	0.006	ICBI
	SSIM	0.979	0.98	0.981	0.98	**0.99**	0.966	0.981	0.961	0.969	0.959	0.951	0.979	0.981	0.979	0.972	ICBI
pepper	FSIM	0.552	0.555	0.552	0.548	**0.62**	0.527	0.55	0.526	0.548	0.517	0.436	0.555	0.564	0.552	0.442	ICBI
	PSNR	46.237	46.244	46.471	46.414	**49.905**	45.416	46.414	43.384	43.927	43.242	42.568	46.305	46.708	46.202	45.193	ICBI
	RMSE	0.005	0.005	0.005	0.005	**0.003**	0.005	0.005	0.007	0.006	0.007	0.007	0.005	0.005	0.005	0.005	ICBI
	SSIM	0.981	0.981	0.981	0.981	**0.992**	0.977	0.981	0.945	0.951	0.945	0.939	0.981	0.982	0.98	0.976	ICBI
ppt3	FSIM	0.54	0.55	0.525	0.508	**0.598**	0.501	0.515	0.511	0.538	0.488	0.476	0.545	0.56	0.518	0.492	ICBI
	PSNR	40.467	40.404	40.75	40.781	**43.38**	39.857	40.695	39.418	40.446	39.341	38.833	40.755	41.138	40.653	39.595	ICBI
	RMSE	0.009	0.01	0.009	0.009	**0.007**	0.01	0.009	0.011	0.009	0.011	0.011	0.009	0.009	0.009	0.01	ICBI
	SSIM	0.939	0.938	0.941	0.941	**0.967**	0.932	0.941	0.922	0.934	0.92	0.914	0.941	0.944	0.94	0.929	ICBI
zebra	FSIM	0.542	0.537	0.529	0.527	**0.617**	0.484	0.528	0.537	0.533	0.533	0.451	0.542	0.532	0.546	0.431	ICBI
	PSNR	42.228	42.14	42.441	42.331	**46.165**	41.136	42.365	40.474	41.943	40.469	39.664	42.375	42.852	42.238	41.04	ICBI
	RMSE	0.008	0.008	0.008	0.008	**0.005**	0.009	0.008	0.009	0.008	0.009	0.01	0.008	0.007	0.008	0.009	ICBI
	SSIM	0.944	0.943	0.946	0.945	**0.976**	0.928	0.945	0.921	0.94	0.921	0.909	0.945	0.949	0.944	0.93	ICBI

Table IX-9: IQA metrics of Set14, scale factor 8

image_name	iqa	DRCN	EDSR	ESPCN	FSRCNN	ICBI	INEDI	LapSRN	VDSR_bicubic	VDSR_bilinear	VDSR_lanczos	VDSR_nearest	bicubic	bilinear	lanczos	nearest	best
baboon	FSIM					**0.363**	0.35		0.332	0.331	0.328	0.314	0.332	0.33	0.328	0.253	ICBI
	PSNR					40.792	39.886		39.569	40.205	39.384	38.649	40.148	**40.831**	39.935	39.418	bilinear
	RMSE					0.009	0.01		0.011	0.01	0.011	0.012	0.01	**0.009**	0.01	0.011	bilinear
	SSIM					0.918	0.898		0.882	0.897	0.877	0.864	0.906	**0.919**	0.901	0.893	bilinear
barbara	FSIM					**0.461**	0.405		0.419	0.413	0.416	0.339	0.42	0.415	0.417	0.295	ICBI
	PSNR					**43.307**	41.389		41.667	42.251	41.538	40.729	42.059	42.698	41.909	41.309	ICBI
	RMSE					**0.007**	0.009		0.008	0.008	0.008	0.009	0.008	0.007	0.008	0.009	ICBI
	SSIM					**0.958**	0.937		0.936	0.944	0.934	0.923	0.945	0.952	0.943	0.935	ICBI
bridge	FSIM					**0.381**	0.353		0.354	0.352	0.349	0.326	0.353	0.351	0.349	0.268	ICBI
	PSNR					**42.501**	39.801		41.112	41.729	40.923	40.098	41.143	41.751	40.954	40.367	ICBI
	RMSE					**0.007**	0.01		0.009	0.008	0.009	0.01	0.009	0.008	0.009	0.01	ICBI
	SSIM					**0.95**	0.92		0.936	0.944	0.933	0.919	0.937	0.944	0.934	0.925	ICBI
coastguard	FSIM					**0.344**	0.33		0.325	0.325	0.323	0.308	0.323	0.322	0.322	0.261	ICBI
	PSNR					**43.176**	41.477		42.41	42.821	42.285	41.797	42.441	42.867	42.317	41.952	ICBI
	RMSE					**0.007**	0.008		0.008	0.007	0.008	0.008	0.008	0.007	0.008	0.008	ICBI
	SSIM					**0.962**	0.936		0.951	0.957	0.95	0.944	0.954	0.958	0.952	0.947	ICBI
comic	FSIM					**0.437**	0.389		0.396	0.4	0.389	0.346	0.398	0.401	0.391	0.282	ICBI
	PSNR					**39.811**	37.247		38.075	38.678	37.91	36.992	38.273	38.892	38.098	37.371	ICBI
	RMSE					**0.01**	0.014		0.012	0.012	0.013	0.014	0.012	0.011	0.012	0.014	ICBI
	SSIM					**0.919**	0.873		0.886	0.899	0.882	0.856	0.892	0.904	0.888	0.869	ICBI
face	FSIM					**0.488**	0.454		0.446	0.454	0.434	0.371	0.447	0.459	0.435	0.342	ICBI
	PSNR					**49.255**	45.592		45.874	46.316	45.756	44.793	46.445	47.0	46.303	45.549	ICBI
	RMSE					**0.003**	0.005		0.005	0.005	0.005	0.006	0.005	0.004	0.005	0.005	ICBI
	SSIM					**0.982**	0.96		0.961	0.967	0.96	0.955	0.967	0.972	0.966	0.962	ICBI
flowers	FSIM					**0.474**	0.395		0.419	0.417	0.414	0.348	0.424	0.421	0.418	0.291	ICBI
	PSNR					**43.064**	39.663		40.527	41.061	40.348	39.374	40.981	41.618	40.782	39.931	ICBI
	RMSE					**0.007**	0.01		0.009	0.009	0.01	0.011	0.009	0.008	0.009	0.01	ICBI
	SSIM					**0.954**	0.914		0.919	0.929	0.915	0.902	0.93	0.939	0.926	0.916	ICBI
foreman	FSIM					**0.544**	0.406		0.465	0.46	0.462	0.383	0.462	0.462	0.459	0.299	ICBI
	PSNR					**45.596**	40.623		41.703	42.011	41.627	40.36	41.85	42.165	41.769	40.613	ICBI
	RMSE					**0.005**	0.009		0.008	0.008	0.008	0.01	0.008	0.008	0.008	0.009	ICBI
	SSIM					**0.983**	0.952		0.964	0.967	0.963	0.946	0.966	0.968	0.965	0.951	ICBI
lenna	FSIM					**0.492**	0.418		0.417	0.415	0.411	0.351	0.417	0.416	0.411	0.31	ICBI
	PSNR					**47.605**	44.707		43.186	43.559	43.073	42.243	44.508	45.098	44.351	43.582	ICBI
	RMSE					**0.004**	0.006		0.007	0.007	0.007	0.008	0.006	0.006	0.006	0.007	ICBI
	SSIM					**0.982**	0.97		0.957	0.962	0.956	0.948	0.968	0.972	0.967	0.961	ICBI
man	FSIM					**0.464**	0.397		0.408	0.412	0.402	0.346	0.409	0.413	0.404	0.29	ICBI
	PSNR					**43.941**	41.205		41.825	42.46	41.648	40.625	41.864	42.484	41.684	40.946	ICBI
	RMSE					**0.006**	0.009		0.008	0.008	0.008	0.009	0.008	0.008	0.008	0.009	ICBI
	SSIM					**0.956**	0.919		0.927	0.936	0.925	0.911	0.928	0.936	0.925	0.916	ICBI
monarch	FSIM					**0.555**	0.428		0.475	0.465	0.475	0.38	0.479	0.472	0.478	0.317	ICBI
	PSNR					**44.526**	40.594		41.775	42.258	41.6	40.531	42.486	43.062	42.276	41.351	ICBI
	RMSE					**0.006**	0.009		0.008	0.008	0.008	0.009	0.008	0.007	0.008	0.009	ICBI
	SSIM					**0.972**	0.937		0.943	0.95	0.941	0.929	0.956	0.96	0.954	0.945	ICBI
pepper	FSIM					**0.521**	0.428		0.434	0.435	0.432	0.345	0.441	0.441	0.437	0.307	ICBI
	PSNR					**46.185**	41.543		41.119	41.431	41.041	40.296	42.329	42.788	42.214	41.383	ICBI
	RMSE					**0.005**	0.008		0.009	0.008	0.009	0.01	0.008	0.007	0.008	0.009	ICBI
	SSIM					**0.98**	0.946		0.92	0.927	0.92	0.911	0.953	0.957	0.951	0.942	ICBI
ppt3	FSIM					**0.489**	0.377		0.412	0.415	0.402	0.388	0.428	0.434	0.415	0.379	ICBI
	PSNR					**39.776**	36.791		37.353	37.818	37.287	36.374	37.74	38.165	37.665	36.7	ICBI
	RMSE					**0.01**	0.014		0.014	0.013	0.014	0.015	0.013	0.012	0.013	0.015	ICBI
	SSIM					**0.929**	0.887		0.887	0.895	0.886	0.87	0.899	0.905	0.898	0.883	ICBI
zebra	FSIM					**0.421**	0.363		0.365	0.364	0.362	0.345	0.366	0.363	0.364	0.283	ICBI
	PSNR					**40.77**	38.05		38.723	39.296	38.598	37.537	39.099	39.687	38.955	37.959	ICBI
	RMSE					**0.009**	0.013		0.012	0.011	0.012	0.013	0.011	0.01	0.011	0.013	ICBI
	SSIM					**0.922**	0.866		0.882	0.894	0.88	0.859	0.892	0.902	0.89	0.871	ICBI

Table IX-10: IQA metrics of Urban100, scale factor 2

image_name	iqa	DRCN	EDSR	ESPCN	FSRCNN	ICBI	INEDI	LapSRN	VDSR_bicubic	VDSR_bilinear	VDSR_lanczos	VDSR_nearest	bicubic	bilinear	lanczos	nearest	best
img_001	FSIM	0.61	**0.69**	0.677	0.67	0.685	0.637	0.672	0.62	0.653	0.599	0.578	0.665	0.673	0.65	0.664	EDSR
	PSNR	45.355	48.741	48.791	48.563	**49.488**	47.817	48.695	45.547	47.702	45.017	44.144	48.56	48.945	48.282	47.323	ICBI
	RMSE	0.005	0.004	0.004	0.004	**0.003**	0.004	0.004	0.005	0.004	0.006	0.006	0.004	0.004	0.004	0.004	ICBI
	SSIM	0.974	0.987	0.987	0.987	**0.989**	0.983	0.987	0.975	0.984	0.971	0.965	0.986	0.987	0.986	0.983	ICBI
img_002	FSIM	0.624	0.692	0.69	0.68	**0.736**	0.67	0.683	0.623	0.66	0.602	0.585	0.679	0.689	0.667	0.68	ICBI
	PSNR	43.836	47.002	47.526	47.172	**49.447**	47.078	47.467	44.045	46.122	43.864	42.19	47.38	47.861	47.136	46.089	ICBI
	RMSE	0.006	0.004	0.004	0.004	**0.003**	0.004	0.004	0.006	0.005	0.006	0.008	0.004	0.004	0.004	0.005	ICBI
	SSIM	0.963	0.981	0.983	0.982	**0.989**	0.981	0.983	0.964	0.977	0.963	0.948	0.982	0.984	0.981	0.977	ICBI
img_003	FSIM	0.597	**0.661**	0.65	0.644	0.655	0.608	0.646	0.603	0.63	0.582	0.566	0.641	0.645	0.631	0.639	EDSR
	PSNR	42.732	45.545	45.51	45.337	**45.811**	43.851	45.41	42.733	44.421	42.298	41.389	45.216	45.647	44.95	44.272	ICBI
	RMSE	0.007	0.005	0.005	0.005	**0.005**	0.006	0.005	0.007	0.006	0.008	0.009	0.005	0.005	0.006	0.006	ICBI
	SSIM	0.954	0.974	0.974	0.973	**0.975**	0.961	0.973	0.953	0.967	0.948	0.938	0.972	0.974	0.97	0.966	ICBI
img_004	FSIM	0.66	0.733	0.722	0.713	**0.736**	0.663	0.714	0.671	0.707	0.638	0.638	0.706	0.726	0.689	0.715	ICBI
	PSNR	43.412	46.162	46.456	46.098	**48.817**	43.157	46.242	43.974	45.609	42.9	42.5	46.161	46.202	46.013	44.467	ICBI
	RMSE	0.007	0.005	0.005	0.005	**0.004**	0.007	0.005	0.006	0.005	0.007	0.007	0.005	0.005	0.005	0.006	ICBI
	SSIM	0.962	0.978	0.979	0.977	**0.988**	0.954	0.978	0.965	0.975	0.957	0.953	0.977	0.977	0.977	0.968	ICBI
img_005	FSIM	0.679	**0.746**	0.73	0.725	0.726	0.671	0.723	0.679	0.704	0.648	0.648	0.716	0.724	0.695	0.719	EDSR
	PSNR	45.471	48.018	48.238	48.026	**50.842**	45.66	48.111	44.612	45.75	44.331	43.364	48.116	48.204	47.967	46.194	ICBI
	RMSE	0.005	0.004	0.004	0.004	**0.003**	0.005	0.004	0.006	0.005	0.006	0.007	0.004	0.004	0.004	0.005	ICBI
	SSIM	0.979	0.988	0.988	0.988	**0.993**	0.977	0.988	0.971	0.976	0.968	0.963	0.988	0.987	0.987	0.981	ICBI
img_006	FSIM	0.63	0.718	0.711	0.703	**0.732**	0.661	0.701	0.636	0.679	0.618	0.571	0.696	0.698	0.681	0.681	ICBI
	PSNR	41.25	44.008	44.179	43.968	**44.868**	42.366	44.155	41.463	43.148	41.385	40.092	44.069	44.435	43.802	42.918	ICBI
	RMSE	0.009	0.006	0.006	0.006	**0.006**	0.008	0.006	0.008	0.007	0.009	0.01	0.006	0.006	0.006	0.007	ICBI
	SSIM	0.94	0.966	0.967	0.965	**0.971**	0.95	0.966	0.941	0.958	0.94	0.922	0.966	0.968	0.964	0.957	ICBI
img_007	FSIM	0.701	0.75	0.745	0.739	**0.767**	0.697	0.739	0.693	0.721	0.665	0.66	0.738	0.744	0.728	0.736	ICBI
	PSNR	47.737	50.055	50.123	49.854	**52.962**	48.47	50.044	46.629	48.03	45.861	45.48	50.044	50.276	49.903	48.422	ICBI
	RMSE	0.004	0.003	0.003	0.003	**0.002**	0.004	0.003	0.005	0.004	0.005	0.005	0.003	0.003	0.003	0.004	ICBI
	SSIM	0.985	0.991	0.991	0.99	**0.995**	0.987	0.991	0.982	0.987	0.978	0.976	0.991	0.991	0.991	0.987	ICBI
img_008	FSIM	0.611	0.696	0.691	0.683	**0.706**	0.648	0.686	0.632	0.667	0.613	0.588	0.681	0.687	0.667	0.677	ICBI
	PSNR	40.54	43.652	43.978	43.764	**44.35**	41.782	43.84	40.951	42.709	40.791	39.329	43.691	44.035	43.413	42.387	ICBI
	RMSE	0.009	0.007	0.006	0.006	**0.006**	0.008	0.006	0.009	0.007	0.009	0.011	0.007	0.006	0.007	0.008	ICBI
	SSIM	0.931	0.963	0.965	0.964	**0.969**	0.944	0.964	0.936	0.955	0.933	0.912	0.963	0.964	0.961	0.952	ICBI
img_009	FSIM	0.643	**0.704**	0.688	0.684	0.663	0.601	0.689	0.633	0.663	0.607	0.593	0.681	0.68	0.672	0.699	EDSR
	PSNR	51.276	54.673	54.849	54.804	**55.618**	53.179	54.871	51.139	52.645	50.766	49.438	54.743	55.008	54.556	53.437	ICBI
	RMSE	0.003	0.002	0.002	0.002	**0.002**	0.002	0.002	0.003	0.002	0.003	0.003	0.002	0.002	0.002	0.002	ICBI
	SSIM	0.993	0.997	0.997	0.997	**0.997**	0.995	0.997	0.993	0.995	0.992	0.99	0.997	0.997	0.997	0.996	ICBI
img_010	FSIM	0.585	0.651	0.655	0.642	**0.663**	0.643	0.648	0.6	0.63	0.574	0.564	0.65	0.661	0.633	0.652	ICBI
	PSNR	43.224	45.182	45.94	45.649	**47.917**	47.113	46.097	43.707	44.979	43.454	42.569	46.012	46.672	45.737	44.787	ICBI
	RMSE	0.007	0.006	0.005	0.005	**0.004**	0.004	0.005	0.007	0.006	0.007	0.007	0.005	0.005	0.005	0.006	ICBI
	SSIM	0.964	0.975	0.979	0.978	**0.986**	0.984	0.98	0.967	0.975	0.965	0.959	0.979	0.981	0.978	0.973	ICBI

Table IX-11: IQA metrics of Urban100, scale factor 3

image_name	iqa	DRCN	EDSR	ESPCN	FSRCNN	ICBI	INEDI	LapSRN	VDSR_bicubic	VDSR_bilinear	VDSR_lanczos	VDSR_nearest	bicubic	bilinear	lanczos	nearest	best
img_001	FSIM	0.558	**0.581**	0.574	0.567				0.532	0.56	0.507	0.488	0.565	0.566	0.56	0.472	EDSR
	PSNR	45.4	45.865	**46.173**	46.015				43.543	45.587	42.593	43.412	45.77	46.124	45.613	44.743	ESPCN
	RMSE	0.005	0.005	**0.005**	0.005				0.007	0.005	0.007	0.007	0.005	0.005	0.005	0.006	ESPCN
	SSIM	0.972	0.975	**0.976**	0.975				0.959	0.973	0.95	0.958	0.974	0.976	0.973	0.968	ESPCN
img_002	FSIM	0.543	0.544	0.552	0.546				0.496	0.544	0.471	0.46	0.551	**0.559**	0.541	0.488	bilinear
	PSNR	43.878	43.712	44.534	44.37				41.7	44.086	41.01	41.655	44.306	**44.753**	44.118	43.25	bilinear
	RMSE	0.006	0.007	0.006	0.006				0.008	0.006	0.009	0.008	0.006	**0.006**	0.006	0.007	bilinear
	SSIM	0.962	0.96	0.967	0.966				0.94	0.963	0.93	0.939	0.965	**0.968**	0.964	0.956	bilinear
img_003	FSIM	0.54	**0.558**	0.549	0.543				0.515	0.536	0.49	0.486	0.542	0.538	0.538	0.456	EDSR
	PSNR	42.353	43.02	**43.305**	43.118				40.502	42.484	39.831	40.632	42.673	43.069	42.497	41.871	ESPCN
	RMSE	0.008	0.007	**0.007**	0.007				0.009	0.008	0.01	0.009	0.007	0.007	0.008	0.008	ESPCN
	SSIM	0.946	0.953	**0.955**	0.953				0.921	0.946	0.909	0.924	0.949	0.953	0.947	0.941	ESPCN
img_004	FSIM	0.586	**0.615**	0.608	0.61				0.552	0.597	0.525	0.504	0.6	0.613	0.591	0.511	EDSR
	PSNR	41.561	41.85	42.432	42.364				40.138	41.799	39.671	39.979	42.197	**42.435**	42.065	40.974	bilinear
	RMSE	0.008	0.008	0.008	0.008				0.01	0.008	0.01	0.01	0.008	**0.008**	0.008	0.009	bilinear
	SSIM	0.935	0.94	**0.944**	0.944				0.916	0.937	0.908	0.913	0.942	0.944	0.941	0.928	ESPCN
img_005	FSIM	0.613	**0.662**	0.636	0.642				0.601	0.618	0.57	0.54	0.62	0.628	0.605	0.515	EDSR
	PSNR	43.622	43.992	44.632	**44.66**				41.866	43.198	41.259	41.293	44.41	44.638	44.256	42.868	FSRCNN
	RMSE	0.007	0.006	0.006	**0.006**				0.008	0.007	0.009	0.009	0.006	0.006	0.006	0.007	FSRCNN
	SSIM	0.966	0.969	0.972	**0.972**				0.948	0.958	0.94	0.941	0.971	0.971	0.97	0.96	FSRCNN
img_006	FSIM	0.568	**0.581**	0.58	0.577				0.534	0.567	0.503	0.48	0.576	0.573	0.569	0.464	EDSR
	PSNR	41.065	41.464	41.818	41.683				39.345	41.362	39.042	39.437	41.491	**41.865**	41.332	40.423	bilinear
	RMSE	0.009	0.008	0.008	0.008				0.011	0.009	0.011	0.011	0.008	**0.008**	0.009	0.01	bilinear
	SSIM	0.935	0.941	**0.943**	0.942				0.907	0.937	0.9	0.909	0.94	0.943	0.938	0.926	ESPCN
img_007	FSIM	0.614	**0.627**	0.62	0.616				0.584	0.609	0.558	0.518	0.622	0.626	0.615	0.535	EDSR
	PSNR	45.549	45.7	45.987	45.911				43.585	45.127	42.74	43.042	45.961	**46.263**	45.838	44.686	bilinear
	RMSE	0.005	0.005	0.005	0.005				0.007	0.006	0.007	0.007	0.005	**0.005**	0.005	0.006	bilinear
	SSIM	0.975	0.976	0.977	0.976				0.962	0.973	0.954	0.957	0.977	**0.978**	0.976	0.969	bilinear
img_008	FSIM	0.57	0.577	0.57	0.569				0.538	0.572	0.518	0.501	**0.577**	0.572	0.571	0.472	bicubic
	PSNR	40.537	40.992	**41.474**	41.337				38.548	40.745	38.28	38.552	40.951	41.362	40.734	39.894	ESPCN
	RMSE	0.009	0.009	**0.008**	0.009				0.012	0.009	0.012	0.012	0.009	0.009	0.009	0.01	ESPCN
	SSIM	0.926	0.932	**0.938**	0.936				0.892	0.928	0.884	0.891	0.932	0.936	0.929	0.916	ESPCN
img_009	FSIM	0.537	**0.563**	0.535	0.528				0.522	0.532	0.502	0.489	0.529	0.534	0.523	0.488	EDSR
	PSNR	51.023	51.102	51.647	51.55				48.631	50.593	48.103	48.77	51.389	**51.728**	51.194	50.41	bilinear
	RMSE	0.003	0.003	0.003	0.003				0.004	0.003	0.004	0.004	0.003	**0.003**	0.003	0.003	bilinear
	SSIM	0.993	0.993	0.993	0.993				0.988	0.992	0.986	0.988	0.993	**0.994**	0.993	0.991	bilinear
img_010	FSIM	0.528	0.533	0.526	0.516				0.502	**0.534**	0.482	0.475	0.526	0.534	0.517	0.488	VDSR_ bilinear
	PSNR	43.122	42.3	43.714	43.49				41.726	43.57	41.137	41.461	43.589	**44.102**	43.383	42.574	bilinear
	RMSE	0.007	0.008	0.007	0.007				0.008	0.007	0.009	0.008	0.007	**0.006**	0.007	0.007	bilinear
	SSIM	0.962	0.953	0.966	0.965				0.95	0.965	0.942	0.947	0.965	**0.968**	0.964	0.957	bilinear

Table IX-12: IQA metrics of Urban100, scale factor 4

image_name	iqa	DRCN	EDSR	ESPCN	FSRCNN	ICBI	INEDI	LapSRN	VDSR_bicubic	VDSR_bilinear	VDSR_lanczos	VDSR_nearest	bicubic	bilinear	lanczos	nearest	best
img_001	FSIM	0.505	0.512	0.513	0.506	**0.559**	0.467	0.511	0.489	0.506	0.485	0.433	0.505	0.507	0.506	0.384	ICBI
	PSNR	44.409	44.335	44.793	44.733	**45.996**	43.504	44.747	42.928	44.712	42.698	42.311	44.442	44.948	44.287	43.463	ICBI
	RMSE	0.006	0.006	0.006	0.006	**0.005**	0.007	0.006	0.007	0.006	0.007	0.008	0.006	0.006	0.006	0.007	ICBI
	SSIM	0.964	0.964	0.967	0.966	**0.975**	0.953	0.966	0.952	0.966	0.949	0.946	0.964	0.967	0.962	0.957	ICBI
img_002	FSIM	0.485	0.453	0.475	0.473	**0.579**	0.481	0.477	0.454	0.489	0.446	0.394	0.488	0.495	0.483	0.403	ICBI
	PSNR	42.811	42.235	43.106	42.965	**45.137**	42.543	42.996	41.033	43.21	40.96	40.546	42.819	43.485	42.634	41.856	ICBI
	RMSE	0.007	0.008	0.007	0.007	**0.006**	0.007	0.007	0.009	0.007	0.009	0.009	0.007	0.007	0.007	0.008	ICBI
	SSIM	0.952	0.945	0.955	0.954	**0.972**	0.949	0.954	0.931	0.956	0.929	0.922	0.952	0.958	0.95	0.941	ICBI
img_003	FSIM	0.485	0.492	0.488	0.484	**0.524**	0.457	0.488	0.476	0.478	0.473	0.438	0.488	0.48	0.491	0.375	ICBI
	PSNR	41.651	41.75	42.241	42.129	**42.583**	40.63	42.157	39.932	41.976	39.813	39.778	41.636	42.22	41.446	40.839	ICBI
	RMSE	0.008	0.008	0.008	0.008	**0.007**	0.009	0.008	0.01	0.008	0.01	0.01	0.008	0.008	0.008	0.009	ICBI
	SSIM	0.935	0.937	0.943	0.941	**0.947**	0.918	0.942	0.909	0.939	0.906	0.907	0.935	0.942	0.932	0.925	ICBI
img_004	FSIM	0.536	0.542	0.539	0.531	**0.603**	0.481	0.534	0.507	0.544	0.501	0.446	0.538	0.547	0.535	0.417	ICBI
	PSNR	40.264	40.111	40.611	40.513	**42.45**	38.296	40.451	38.925	40.622	38.962	38.569	40.393	40.819	40.279	39.32	ICBI
	RMSE	0.01	0.01	0.009	0.009	**0.008**	0.012	0.009	0.011	0.009	0.011	0.012	0.01	0.009	0.01	0.011	ICBI
	SSIM	0.908	0.907	0.914	0.912	**0.946**	0.852	0.911	0.885	0.914	0.885	0.877	0.91	0.917	0.908	0.892	ICBI
img_005	FSIM	0.558	0.601	0.574	0.573	**0.619**	0.51	0.582	0.54	0.56	0.526	0.478	0.554	0.562	0.549	0.424	ICBI
	PSNR	42.537	42.255	42.95	42.809	**45.109**	40.475	42.976	40.651	42.226	40.445	39.977	42.649	43.107	42.486	41.287	ICBI
	RMSE	0.007	0.008	0.007	0.007	**0.006**	0.009	0.007	0.009	0.008	0.01	0.01	0.007	0.007	0.008	0.009	ICBI
	SSIM	0.956	0.954	0.959	0.958	**0.975**	0.926	0.959	0.933	0.947	0.929	0.923	0.956	0.959	0.955	0.944	ICBI
img_006	FSIM	0.508	0.51	0.508	0.507	**0.568**	0.459	0.51	0.496	0.502	0.487	0.426	0.51	0.502	0.511	0.374	ICBI
	PSNR	40.447	40.3	41.006	40.822	**41.254**	39.151	40.971	38.636	40.888	38.691	38.602	40.521	41.154	40.301	39.525	ICBI
	RMSE	0.009	0.01	0.009	0.009	**0.009**	0.011	0.009	0.012	0.009	0.012	0.012	0.009	0.009	0.01	0.011	ICBI
	SSIM	0.923	0.922	0.93	0.928	**0.936**	0.893	0.93	0.89	0.927	0.89	0.891	0.924	0.932	0.92	0.909	ICBI
img_007	FSIM	0.544	0.537	0.538	0.532	**0.639**	0.516	0.536	0.526	0.543	0.518	0.437	0.546	0.552	0.543	0.434	ICBI
	PSNR	43.811	43.428	43.88	43.703	**47.624**	42.523	43.753	42.318	43.733	42.176	41.331	43.824	44.35	43.689	42.708	ICBI
	RMSE	0.006	0.007	0.006	0.007	**0.004**	0.007	0.006	0.008	0.007	0.008	0.009	0.006	0.006	0.007	0.007	ICBI
	SSIM	0.962	0.959	0.963	0.962	**0.984**	0.95	0.962	0.949	0.962	0.947	0.937	0.962	0.966	0.961	0.953	ICBI
img_008	FSIM	0.51	0.499	0.495	0.492	**0.548**	0.472	0.495	0.499	0.503	0.495	0.443	0.511	0.5	0.513	0.379	ICBI
	PSNR	39.724	39.744	40.329	40.121	**40.518**	38.433	40.23	37.775	40.06	37.824	37.593	39.763	40.386	39.554	38.775	ICBI
	RMSE	0.01	0.01	0.01	0.01	**0.009**	0.012	0.01	0.013	0.01	0.013	0.013	0.01	0.01	0.011	0.012	ICBI
	SSIM	0.912	0.911	0.921	0.918	**0.927**	0.887	0.92	0.872	0.917	0.873	0.867	0.913	0.922	0.909	0.894	ICBI
img_009	FSIM	0.467	0.484	0.471	0.469	**0.509**	0.431	0.475	0.465	0.47	0.456	0.42	0.463	0.468	0.46	0.4	ICBI
	PSNR	50.045	49.665	50.404	50.338	**51.321**	49.244	50.358	48.189	49.887	48.278	48.02	50.049	50.557	49.875	49.18	ICBI
	RMSE	0.003	0.003	0.003	0.003	**0.003**	0.003	0.003	0.004	0.003	0.004	0.004	0.003	0.003	0.003	0.003	ICBI
	SSIM	0.991	0.99	0.992	0.992	**0.993**	0.989	0.992	0.987	0.991	0.987	0.986	0.991	0.992	0.991	0.989	ICBI
img_010	FSIM	0.454	0.456	0.455	0.444	**0.512**	0.479	0.458	0.439	0.462	0.432	0.412	0.452	0.46	0.448	0.403	ICBI
	PSNR	42.407	41.657	42.729	42.492	**44.838**	43.43	42.728	40.91	42.916	40.751	40.623	42.45	43.221	42.24	41.508	ICBI
	RMSE	0.008	0.008	0.007	0.008	**0.006**	0.007	0.007	0.009	0.007	0.009	0.009	0.008	0.007	0.008	0.008	ICBI
	SSIM	0.956	0.947	0.958	0.957	**0.974**	0.965	0.958	0.941	0.96	0.938	0.938	0.957	0.962	0.955	0.948	ICBI

Table IX-13: IQA metrics of Urban100, scale factor 8

image_name	iqa	DRCN	EDSR	ESPCN	FSRCNN	ICBI	INEDI	LapSRN	VDSR_bicubic	VDSR_bilinear	VDSR_lanczos	VDSR_nearest	bicubic	bilinear	lanczos	nearest	best
img_001	FSIM					**0.457**	0.379		0.404	0.409	0.4	0.349	0.405	0.409	0.401	0.273	ICBI
	PSNR					**43.737**	40.649		41.745	42.374	41.591	40.487	41.907	42.467	41.751	40.899	ICBI
	RMSE					**0.007**	0.009		0.008	0.008	0.008	0.009	0.008	0.008	0.008	0.009	ICBI
	SSIM					**0.956**	0.911		0.931	0.939	0.928	0.914	0.932	0.94	0.93	0.92	ICBI
img_002	FSIM					**0.457**	0.388		0.383	0.385	0.378	0.335	0.384	0.387	0.379	0.304	ICBI
	PSNR					**42.552**	39.673		39.975	40.688	39.802	38.823	40.062	40.761	39.884	39.2	ICBI
	RMSE					**0.007**	0.01		0.01	0.009	0.01	0.011	0.01	0.009	0.01	0.011	ICBI
	SSIM					**0.95**	0.91		0.917	0.929	0.914	0.894	0.919	0.93	0.916	0.902	ICBI
img_003	FSIM					**0.412**	0.383		0.384	0.385	0.379	0.338	0.385	0.385	0.38	0.258	ICBI
	PSNR					**40.598**	38.7		39.54	40.199	39.354	38.495	39.671	40.303	39.483	38.851	ICBI
	RMSE					**0.009**	0.012		0.011	0.01	0.011	0.012	0.01	0.01	0.011	0.011	ICBI
	SSIM					**0.915**	0.875		0.894	0.909	0.89	0.875	0.899	0.912	0.895	0.885	ICBI
img_004	FSIM					**0.455**	0.38		0.405	0.409	0.401	0.354	0.409	0.411	0.405	0.286	ICBI
	PSNR					**38.848**	36.446		37.472	37.994	37.381	36.498	37.525	38.027	37.426	36.609	ICBI
	RMSE					**0.011**	0.015		0.013	0.013	0.014	0.015	0.013	0.013	0.013	0.015	ICBI
	SSIM					**0.868**	0.779		0.824	0.841	0.821	0.796	0.827	0.843	0.824	0.8	ICBI
img_005	FSIM					**0.505**	0.391		0.441	0.442	0.438	0.383	0.443	0.445	0.439	0.305	ICBI
	PSNR					**41.048**	38.173		39.205	39.732	39.038	38.006	39.536	40.09	39.36	38.452	ICBI
	RMSE					**0.009**	0.012		0.011	0.01	0.011	0.013	0.011	0.01	0.011	0.012	ICBI
	SSIM					**0.937**	0.877		0.907	0.911	0.903	0.885	0.915	0.921	0.912	0.899	ICBI
img_006	FSIM					**0.428**	0.367		0.389	0.391	0.385	0.342	0.392	0.393	0.388	0.261	ICBI
	PSNR					**39.501**	37.498		38.778	39.434	38.583	37.66	38.849	39.496	38.651	37.918	ICBI
	RMSE					**0.011**	0.013		0.012	0.011	0.012	0.013	0.011	0.011	0.012	0.013	ICBI
	SSIM					**0.905**	0.847		0.888	0.901	0.883	0.863	0.892	0.903	0.887	0.872	ICBI
img_007	FSIM					**0.502**	0.406		0.419	0.414	0.415	0.348	0.423	0.419	0.419	0.301	ICBI
	PSNR					**43.096**	39.045		39.911	40.502	39.76	38.82	40.045	40.678	39.881	39.123	ICBI
	RMSE					**0.007**	0.011		0.01	0.009	0.01	0.011	0.01	0.009	0.01	0.011	ICBI
	SSIM					**0.955**	0.901		0.915	0.925	0.912	0.896	0.918	0.928	0.915	0.903	ICBI
img_008	FSIM					**0.402**	0.378		0.373	0.373	0.369	0.355	0.374	0.372	0.37	0.262	ICBI
	PSNR					**38.372**	37.016		37.7	38.445	37.485	36.613	37.896	38.589	37.683	37.085	bilinear
	RMSE					**0.012**	0.014		0.013	0.012	0.013	0.015	0.013	0.012	0.013	0.014	bilinear
	SSIM					**0.891**	0.861		0.875	0.89	0.869	0.844	0.879	0.893	0.874	0.858	bilinear
img_009	FSIM					**0.438**	0.384		0.398	0.395	0.398	0.343	0.401	0.401	0.399	0.311	ICBI
	PSNR					**48.983**	47.288		47.382	47.88	47.242	46.558	47.696	48.236	47.539	46.989	ICBI
	RMSE					**0.004**	0.004		0.004	0.004	0.004	0.005	0.004	0.004	0.004	0.004	ICBI
	SSIM					**0.99**	0.985		0.986	0.987	0.985	0.982	0.986	0.988	0.986	0.984	ICBI
img_010	FSIM					**0.435**	0.394		0.371	0.374	0.372	0.327	0.375	0.377	0.375	0.294	ICBI
	PSNR					**42.714**	41.227		40.472	41.356	40.252	39.431	40.628	41.443	40.396	39.756	ICBI
	RMSE					**0.007**	0.009		0.009	0.009	0.01	0.011	0.009	0.008	0.01	0.01	ICBI
	SSIM					**0.962**	0.949		0.942	0.95	0.939	0.927	0.943	0.95	0.941	0.932	ICBI

Table IX-14: IQA metrics of tc100, scale factor 2

image_name	iqa	DRCN	EDSR	ESPCN	FSRCNN	ICBI	INEDI	LapSRN	VDSR_bicubic	VDSR_bilinear	VDSR_lanczos	VDSR_nearest	bicubic	bilinear	lanczos	nearest	best
DSC02725	FSIM	0.44	0.462	0.472	0.469	0.441	0.437	0.46	0.508	**0.513**	0.503	0.501	0.435	0.435	0.433	0.435	VDSR_ bilinear
	PSNR	59.317	60.409	60.484	60.385	**61.054**	59.389	60.474	44.232	44.265	44.212	44.237	60.209	60.284	60.168	59.592	ICBI
	RMSE	0.001	0.001	0.001	0.001	**0.001**	0.001	0.001	0.006	0.006	0.006	0.006	0.001	0.001	0.001	0.001	ICBI
	SSIM	0.999	0.999	0.999	0.999	**0.999**	0.999	0.999	0.92	0.921	0.92	0.921	0.999	0.999	0.999	0.999	ICBI
DSC02842	FSIM	0.615	0.633	0.632	0.63	**0.647**	0.626	0.632	0.6	0.621	0.581	0.582	0.627	0.631	0.626	0.619	ICBI
	PSNR	53.515	54.273	54.141	54.032	**55.15**	53.578	54.151	45.642	45.695	45.48	45.48	54.03	54.211	54.007	53.134	ICBI
	RMSE	0.002	0.002	0.002	0.002	**0.002**	0.002	0.002	0.005	0.005	0.005	0.005	0.002	0.002	0.002	0.002	ICBI
	SSIM	0.993	0.994	0.994	0.994	**0.995**	0.993	0.994	0.958	0.958	0.957	0.957	0.994	0.994	0.994	0.993	ICBI
DSC02861	FSIM	0.664	0.673	0.674	0.672	**0.706**	0.644	0.668	0.657	0.656	0.644	0.624	0.67	0.655	0.672	0.659	ICBI
	PSNR	52.174	54.236	54.337	54.114	**55.611**	53.849	54.283	50.919	52.29	48.7	49.084	54.185	54.42	54.01	53.031	ICBI
	RMSE	0.002	0.002	0.002	0.002	**0.002**	0.002	0.002	0.003	0.002	0.004	0.004	0.002	0.002	0.002	0.002	ICBI
	SSIM	0.994	0.996	0.996	0.996	**0.997**	0.996	0.996	0.991	0.994	0.986	0.987	0.996	0.996	0.996	0.995	ICBI
DSC03071	FSIM	0.627	0.646	0.648	0.644	**0.659**	0.628	0.648	0.625	0.651	0.607	0.603	0.637	0.646	0.634	0.63	ICBI
	PSNR	57.353	58.811	58.72	58.652	**60.594**	56.887	58.809	56.156	57.617	55.011	54.989	58.564	58.715	58.512	57.155	ICBI
	RMSE	0.001	0.001	0.001	0.001	**0.001**	0.001	0.001	0.002	0.001	0.002	0.002	0.001	0.001	0.001	0.001	ICBI
	SSIM	0.998	0.999	0.999	0.999	**0.999**	0.998	0.999	0.998	0.998	0.997	0.997	0.999	0.999	0.999	0.998	ICBI
DSC03083	FSIM	0.579	0.598	0.602	0.6	**0.611**	0.582	0.598	0.584	0.592	0.575	0.568	0.594	0.594	0.593	0.586	ICBI
	PSNR	56.426	57.421	57.389	57.304	**58.403**	56.265	57.429	50.061	50.292	49.855	49.898	57.214	57.284	57.178	56.334	ICBI
	RMSE	0.002	0.001	0.001	0.001	**0.001**	0.002	0.001	0.003	0.003	0.003	0.003	0.001	0.001	0.001	0.002	ICBI
	SSIM	0.997	0.998	0.998	0.998	**0.998**	0.997	0.998	0.982	0.982	0.982	0.982	0.998	0.998	0.998	0.997	ICBI
DSC03091	FSIM	0.584	0.61	0.621	0.612	0.614	0.594	0.61	0.601	**0.635**	0.578	0.586	0.603	0.612	0.595	0.593	VDSR_ bilinear
	PSNR	53.72	55.135	55.023	54.829	**58.226**	52.766	55.092	50.534	51.263	49.98	50.196	54.947	55.226	54.9	53.206	ICBI
	RMSE	0.002	0.002	0.002	0.002	**0.001**	0.002	0.002	0.003	0.003	0.003	0.003	0.002	0.002	0.002	0.002	ICBI
	SSIM	0.996	0.998	0.997	0.997	**0.999**	0.996	0.998	0.992	0.993	0.991	0.991	0.997	0.998	0.997	0.996	ICBI
DSC03144	FSIM	0.647	0.676	0.676	0.673	**0.703**	0.657	0.676	0.637	0.662	0.613	0.606	0.672	0.679	0.667	0.657	ICBI
	PSNR	50.394	51.846	51.906	51.694	**53.856**	50.18	51.887	48.967	50.364	47.943	47.638	51.744	51.929	51.646	50.278	ICBI
	RMSE	0.003	0.003	0.003	0.003	**0.002**	0.003	0.003	0.004	0.003	0.004	0.004	0.003	0.003	0.003	0.003	ICBI
	SSIM	0.991	0.994	0.994	0.994	**0.996**	0.991	0.994	0.989	0.992	0.986	0.984	0.994	0.994	0.994	0.991	ICBI
DSC03345	FSIM	0.623	0.663	0.662	0.656	**0.668**	0.643	0.663	0.622	0.656	0.6	0.604	0.65	0.666	0.643	0.646	ICBI
	PSNR	52.763	54.863	54.987	54.62	**57.528**	52.863	54.965	50.393	51.359	49.652	49.704	54.556	54.861	54.423	53.05	ICBI
	RMSE	0.002	0.002	0.002	0.002	**0.001**	0.002	0.002	0.003	0.003	0.003	0.003	0.002	0.002	0.002	0.002	ICBI
	SSIM	0.995	0.997	0.997	0.997	**0.998**	0.995	0.997	0.992	0.994	0.991	0.991	0.997	0.997	0.997	0.995	ICBI
DSC03366	FSIM	0.607	0.628	0.63	0.626	0.632	0.61	0.628	0.614	**0.638**	0.594	0.602	0.618	0.625	0.614	0.614	VDSR_ bilinear
	PSNR	57.336	58.682	58.555	58.441	**61.157**	56.446	58.649	48.275	48.412	48.068	48.153	58.481	58.618	58.434	56.701	ICBI
	RMSE	0.001	0.001	0.001	0.001	**0.001**	0.002	0.001	0.004	0.004	0.004	0.004	0.001	0.001	0.001	0.001	ICBI
	SSIM	0.998	0.998	0.998	0.998	**0.999**	0.997	0.998	0.977	0.978	0.976	0.977	0.998	0.998	0.998	0.998	ICBI
DSC03376	FSIM	0.554	0.578	0.577	0.582	**0.591**	0.538	0.578	0.573	0.591	0.556	0.56	0.568	0.567	0.567	0.563	ICBI
	PSNR	56.783	58.887	58.949	58.871	**60.723**	56.152	59.031	47.932	48.207	47.64	47.834	58.835	59.103	58.757	57.484	ICBI
	RMSE	0.001	0.001	0.001	0.001	**0.001**	0.002	0.001	0.004	0.004	0.004	0.004	0.001	0.001	0.001	0.001	ICBI
	SSIM	0.998	0.999	0.999	0.999	**0.999**	0.998	0.999	0.981	0.982	0.98	0.981	0.999	0.999	0.999	0.998	ICBI

Table IX-15: IQA metrics of tc100, scale factor 3

image_name	iqa	DRCN	EDSR	ESPCN	FSRCNN	ICBI	INEDI	LapSRN	VDSR_bicubic	VDSR_bilinear	VDSR_lanczos	VDSR_nearest	bicubic	bilinear	lanczos	nearest	best
DSC02725	FSIM	0.426	0.46	0.458	0.466				0.482	**0.485**	0.474	0.475	0.427	0.429	0.424	0.426	VDSR_ bilinear
	PSNR	56.816	57.304	**57.454**	57.387				44.127	44.142	44.047	44.091	57.299	57.383	57.229	56.542	ESPCN
	RMSE	0.001	0.001	**0.001**	0.001				0.006	0.006	0.006	0.006	0.001	0.001	0.001	0.001	ESPCN
	SSIM	0.998	0.998	**0.998**	0.998				0.92	0.92	0.919	0.919	0.998	0.998	0.998	0.998	ESPCN
DSC02842	FSIM	0.546	0.561	**0.563**	0.556				0.533	0.544	0.52	0.486	0.549	0.552	0.547	0.513	ESPCN
	PSNR	50.443	50.698	50.718	50.634				44.942	45.114	44.7	44.659	50.632	**50.826**	50.565	49.686	bilinear
	RMSE	0.003	0.003	0.003	0.003				0.006	0.006	0.006	0.006	0.003	**0.003**	0.003	0.003	bilinear
	SSIM	0.986	0.987	0.987	0.986				0.953	0.954	0.951	0.951	0.987	**0.987**	0.986	0.985	bilinear
DSC02861	FSIM	0.532	0.495	0.506	0.503				0.543	0.52	**0.546**	0.501	0.527	0.512	0.535	0.51	VDSR_ lanczos
	PSNR	50.886	50.879	**51.746**	51.565				48.404	50.067	47.687	47.777	51.301	51.588	51.207	50.399	ESPCN
	RMSE	0.003	0.003	**0.003**	0.003				0.004	0.003	0.004	0.004	0.003	0.003	0.003	0.003	ESPCN
	SSIM	0.991	0.991	**0.993**	0.993				0.985	0.99	0.982	0.982	0.992	0.993	0.992	0.99	ESPCN
DSC03071	FSIM	0.571	0.585	0.592	**0.593**				0.568	0.581	0.549	0.518	0.574	0.582	0.569	0.533	FSRCNN
	PSNR	54.483	54.989	**55.104**	55.026				53.032	54.302	52.094	52.083	54.845	55.102	54.748	53.577	ESPCN
	RMSE	0.002	0.002	**0.002**	0.002				0.002	0.002	0.002	0.002	0.002	0.002	0.002	0.002	ESPCN
	SSIM	0.997	0.997	**0.997**	0.997				0.995	0.996	0.994	0.994	0.997	0.997	0.997	0.996	ESPCN
DSC03083	FSIM	0.504	0.514	0.514	0.517				0.508	**0.519**	0.503	0.479	0.511	0.511	0.511	0.49	VDSR_ bilinear
	PSNR	53.852	54.1	**54.277**	54.251				49.031	49.421	48.758	48.709	54.139	54.267	54.066	53.235	ESPCN
	RMSE	0.002	0.002	**0.002**	0.002				0.004	0.003	0.004	0.004	0.002	0.002	0.002	0.002	ESPCN
	SSIM	0.995	0.995	0.995	**0.995**				0.979	0.98	0.978	0.978	0.995	0.995	0.995	0.994	FSRCNN
DSC03091	FSIM	0.544	0.558	0.559	0.569				0.547	**0.581**	0.521	0.511	0.548	0.554	0.544	0.51	VDSR_ bilinear
	PSNR	50.075	50.26	50.559	50.732				47.941	48.724	47.329	46.909	50.653	**50.821**	50.58	49.083	bilinear
	RMSE	0.003	0.003	0.003	0.003				0.004	0.004	0.004	0.005	0.003	**0.003**	0.003	0.004	bilinear
	SSIM	0.993	0.993	0.993	0.994				0.986	0.988	0.984	0.983	0.994	**0.994**	0.993	0.99	bilinear
DSC03144	FSIM	0.576	**0.59**	0.587	0.586				0.555	0.574	0.536	0.498	0.582	0.586	0.576	0.534	EDSR
	PSNR	47.711	48.138	**48.361**	48.291				45.816	47.29	44.937	44.792	48.084	48.314	47.944	46.753	ESPCN
	RMSE	0.004	0.004	**0.004**	0.004				0.005	0.004	0.006	0.006	0.004	0.004	0.004	0.005	ESPCN
	SSIM	0.985	0.986	**0.987**	0.987				0.977	0.984	0.972	0.971	0.986	0.987	0.986	0.981	ESPCN
DSC03345	FSIM	0.571	**0.597**	0.589	0.59				0.557	0.586	0.534	0.514	0.576	0.59	0.569	0.541	EDSR
	PSNR	50.16	50.899	50.799	50.924				48.044	49.132	47.183	47.528	50.789	**51.055**	50.686	49.522	bilinear
	RMSE	0.003	0.003	0.003	0.003				0.004	0.003	0.004	0.004	0.003	**0.003**	0.003	0.003	bilinear
	SSIM	0.991	0.993	0.992	0.993				0.987	0.99	0.984	0.985	0.992	**0.993**	0.992	0.99	bilinear
DSC03366	FSIM	0.55	0.575	0.575	0.577				0.56	**0.583**	0.533	0.516	0.555	0.566	0.549	0.52	VDSR_ bilinear
	PSNR	53.875	54.642	54.649	**54.667**				47.624	47.818	47.2	47.291	54.513	54.633	54.433	52.916	FSRCNN
	RMSE	0.002	0.002	0.002	**0.002**				0.004	0.004	0.004	0.004	0.002	0.002	0.002	0.002	FSRCNN
	SSIM	0.995	**0.996**	0.996	0.996				0.975	0.976	0.973	0.974	0.996	0.996	0.996	0.994	EDSR
DSC03376	FSIM	0.488	0.49	0.494	0.506				0.494	**0.521**	0.489	0.466	0.482	0.489	0.479	0.447	VDSR_ bilinear
	PSNR	54.817	55.14	55.041	55.36				47.049	47.687	46.906	47.2	55.263	**55.565**	55.097	53.979	bilinear
	RMSE	0.002	0.002	0.002	0.002				0.004	0.004	0.005	0.004	0.002	**0.002**	0.002	0.002	bilinear
	SSIM	0.997	0.997	0.997	0.997				0.978	0.981	0.978	0.979	0.997	**0.997**	0.997	0.996	bilinear

Table IX-16: IQA metrics of tc-100, scale factor 4

image_name	iqa	DRCN	EDSR	ESPCN	FSRCNN	ICBI	INEDI	LapSRN	VDSR_bicubic	VDSR_bilinear	VDSR_lanczos	VDSR_nearest	bicubic	bilinear	lanczos	nearest	best
DSC02725	FSIM	0.505	0.512	0.513	0.506	**0.559**	0.467	0.511	0.489	0.506	0.485	0.433	0.505	0.507	0.506	0.384	ICBI
	PSNR	44.409	44.335	44.793	44.733	**45.996**	43.504	44.747	42.928	44.712	42.698	42.311	44.442	44.948	44.287	43.463	ICBI
	RMSE	0.006	0.006	0.006	0.006	**0.005**	0.007	0.006	0.007	0.006	0.007	0.008	0.006	0.006	0.006	0.007	ICBI
	SSIM	0.964	0.964	0.967	0.966	**0.975**	0.953	0.966	0.952	0.966	0.949	0.946	0.964	0.967	0.962	0.957	ICBI
DSC02842	FSIM	0.485	0.453	0.475	0.473	**0.579**	0.481	0.477	0.454	0.489	0.446	0.394	0.488	0.495	0.483	0.403	ICBI
	PSNR	42.811	42.235	43.106	42.965	**45.137**	42.543	42.996	41.033	43.21	40.96	40.546	42.819	43.485	42.634	41.856	ICBI
	RMSE	0.007	0.008	0.007	0.007	**0.006**	0.007	0.007	0.009	0.007	0.009	0.009	0.007	0.007	0.007	0.008	ICBI
	SSIM	0.952	0.945	0.955	0.954	**0.972**	0.949	0.954	0.931	0.956	0.929	0.922	0.952	0.958	0.95	0.941	ICBI
DSC02861	FSIM	0.485	0.492	0.488	0.484	**0.524**	0.457	0.488	0.476	0.478	0.473	0.438	0.488	0.48	0.491	0.375	ICBI
	PSNR	41.651	41.75	42.241	42.129	**42.583**	40.63	42.157	39.932	41.976	39.813	39.778	41.636	42.22	41.446	40.839	ICBI
	RMSE	0.008	0.008	0.008	0.008	**0.007**	0.009	0.008	0.01	0.008	0.01	0.01	0.008	0.008	0.008	0.009	ICBI
	SSIM	0.935	0.937	0.943	0.941	**0.947**	0.918	0.942	0.909	0.939	0.906	0.907	0.935	0.942	0.932	0.925	ICBI
DSC03071	FSIM	0.536	0.542	0.539	0.531	**0.603**	0.481	0.534	0.507	0.544	0.501	0.446	0.538	0.547	0.535	0.417	ICBI
	PSNR	40.264	40.111	40.611	40.513	**42.45**	38.296	40.451	38.925	40.622	38.962	38.569	40.393	40.819	40.279	39.32	ICBI
	RMSE	0.01	0.01	0.009	0.009	**0.008**	0.012	0.009	0.011	0.009	0.011	0.012	0.01	0.009	0.01	0.011	ICBI
	SSIM	0.908	0.907	0.914	0.912	**0.946**	0.852	0.911	0.885	0.914	0.885	0.877	0.91	0.917	0.908	0.892	ICBI
DSC03083	FSIM	0.558	0.601	0.574	0.573	**0.619**	0.51	0.582	0.54	0.56	0.526	0.478	0.554	0.562	0.549	0.424	ICBI
	PSNR	42.537	42.255	42.95	42.809	**45.109**	40.475	42.976	40.651	42.226	40.445	39.977	42.649	43.107	42.486	41.287	ICBI
	RMSE	0.007	0.008	0.007	0.007	**0.006**	0.009	0.007	0.009	0.008	0.01	0.01	0.007	0.007	0.008	0.009	ICBI
	SSIM	0.956	0.954	0.959	0.958	**0.975**	0.926	0.959	0.933	0.947	0.929	0.923	0.956	0.959	0.955	0.944	ICBI
DSC03091	FSIM	0.508	0.51	0.508	0.507	**0.568**	0.459	0.51	0.496	0.502	0.487	0.426	0.51	0.502	0.511	0.374	ICBI
	PSNR	40.447	40.3	41.006	40.822	**41.254**	39.151	40.971	38.636	40.888	38.691	38.602	40.521	41.154	40.301	39.525	ICBI
	RMSE	0.009	0.01	0.009	0.009	**0.009**	0.011	0.009	0.012	0.009	0.012	0.012	0.009	0.009	0.01	0.011	ICBI
	SSIM	0.923	0.922	0.93	0.928	**0.936**	0.893	0.93	0.89	0.927	0.89	0.891	0.924	0.932	0.92	0.909	ICBI
DSC03144	FSIM	0.544	0.537	0.538	0.532	**0.639**	0.516	0.536	0.526	0.543	0.518	0.437	0.546	0.552	0.543	0.434	ICBI
	PSNR	43.811	43.428	43.88	43.703	**47.624**	42.523	43.753	42.318	43.733	42.176	41.331	43.824	44.35	43.689	42.708	ICBI
	RMSE	0.006	0.007	0.006	0.007	**0.004**	0.007	0.006	0.008	0.007	0.008	0.009	0.006	0.006	0.007	0.007	ICBI
	SSIM	0.962	0.959	0.963	0.962	**0.984**	0.95	0.962	0.949	0.962	0.947	0.937	0.962	0.966	0.961	0.953	ICBI
DSC03345	FSIM	0.51	0.499	0.495	0.492	**0.548**	0.472	0.495	0.499	0.503	0.495	0.443	0.511	0.5	0.513	0.379	ICBI
	PSNR	39.724	39.744	40.329	40.121	**40.518**	38.433	40.23	37.775	40.06	37.824	37.593	39.763	40.386	39.554	38.775	ICBI
	RMSE	0.01	0.01	0.01	0.01	**0.009**	0.012	0.01	0.013	0.01	0.013	0.013	0.01	0.01	0.011	0.012	ICBI
	SSIM	0.912	0.911	0.921	0.918	**0.927**	0.887	0.92	0.872	0.917	0.873	0.867	0.913	0.922	0.909	0.894	ICBI
DSC03366	FSIM	0.467	0.484	0.471	0.469	**0.509**	0.431	0.475	0.465	0.47	0.456	0.42	0.463	0.468	0.46	0.4	ICBI
	PSNR	50.045	49.665	50.404	50.338	**51.321**	49.244	50.358	48.189	49.887	48.278	48.02	50.049	50.557	49.875	49.18	ICBI
	RMSE	0.003	0.003	0.003	0.003	**0.003**	0.003	0.003	0.004	0.003	0.004	0.004	0.003	0.003	0.003	0.003	ICBI
	SSIM	0.991	0.99	0.992	0.992	**0.993**	0.989	0.992	0.987	0.991	0.987	0.986	0.991	0.992	0.991	0.989	ICBI
DSC03376	FSIM	0.454	0.456	0.455	0.444	**0.512**	0.479	0.458	0.439	0.462	0.432	0.412	0.452	0.46	0.448	0.403	ICBI
	PSNR	42.407	41.657	42.729	42.492	**44.838**	43.43	42.728	40.91	42.916	40.751	40.623	42.45	43.221	42.24	41.508	ICBI
	RMSE	0.008	0.008	0.007	0.008	**0.006**	0.007	0.007	0.009	0.007	0.009	0.009	0.008	0.007	0.008	0.008	ICBI
	SSIM	0.956	0.947	0.958	0.957	**0.974**	0.965	0.958	0.941	0.96	0.938	0.938	0.957	0.962	0.955	0.948	ICBI

Table IX-17: IQA metrics of tc-100, scale factor 8

image_name	iqa	DRCN	EDSR	ESPCN	FSRCNN	ICBI	INEDI	LapSRN	VDSR_bicubic	VDSR_bilinear	VDSR_lanczos	VDSR_nearest	bicubic	bilinear	lanczos	nearest	best
DSC02725	FSIM					**0.457**	0.379		0.404	0.409	0.4	0.349	0.405	0.409	0.401	0.273	ICBI
	PSNR					**43.737**	40.649		41.745	42.374	41.591	40.487	41.907	42.467	41.751	40.899	ICBI
	RMSE					**0.007**	0.009		0.008	0.008	0.008	0.009	0.008	0.008	0.008	0.009	ICBI
	SSIM					**0.956**	0.911		0.931	0.939	0.928	0.914	0.932	0.94	0.93	0.92	ICBI
DSC02842	FSIM					**0.457**	0.388		0.383	0.385	0.378	0.335	0.384	0.387	0.379	0.304	ICBI
	PSNR					**42.552**	39.673		39.975	40.688	39.802	38.823	40.062	40.761	39.884	39.2	ICBI
	RMSE					**0.007**	0.01		0.01	0.009	0.01	0.011	0.01	0.009	0.01	0.011	ICBI
	SSIM					**0.95**	0.91		0.917	0.929	0.914	0.894	0.919	0.93	0.916	0.902	ICBI
DSC02861	FSIM					**0.412**	0.383		0.384	0.385	0.379	0.338	0.385	0.385	0.38	0.258	ICBI
	PSNR					**40.598**	38.7		39.54	40.199	39.354	38.495	39.671	40.303	39.483	38.851	ICBI
	RMSE					**0.009**	0.012		0.011	0.01	0.011	0.012	0.01	0.01	0.011	0.011	ICBI
	SSIM					**0.915**	0.875		0.894	0.909	0.89	0.875	0.899	0.912	0.895	0.885	ICBI
DSC03071	FSIM					**0.455**	0.38		0.405	0.409	0.401	0.354	0.409	0.411	0.405	0.286	ICBI
	PSNR					**38.848**	36.446		37.472	37.994	37.381	36.498	37.525	38.027	37.426	36.609	ICBI
	RMSE					**0.011**	0.015		0.013	0.013	0.014	0.015	0.013	0.013	0.013	0.015	ICBI
	SSIM					**0.868**	0.779		0.824	0.841	0.821	0.796	0.827	0.843	0.824	0.8	ICBI
DSC03083	FSIM					**0.505**	0.391		0.441	0.442	0.438	0.383	0.443	0.445	0.439	0.305	ICBI
	PSNR					**41.048**	38.173		39.205	39.732	39.038	38.006	39.536	40.09	39.36	38.452	ICBI
	RMSE					**0.009**	0.012		0.011	0.01	0.011	0.013	0.011	0.01	0.011	0.012	ICBI
	SSIM					**0.937**	0.877		0.907	0.911	0.903	0.885	0.915	0.921	0.912	0.899	ICBI
DSC03091	FSIM					**0.428**	0.367		0.389	0.391	0.385	0.342	0.392	0.393	0.388	0.261	ICBI
	PSNR					**39.501**	37.498		38.778	39.434	38.583	37.66	38.849	39.496	38.651	37.918	ICBI
	RMSE					**0.011**	0.013		0.012	0.011	0.012	0.013	0.011	0.011	0.012	0.013	ICBI
	SSIM					**0.905**	0.847		0.888	0.901	0.883	0.863	0.892	0.903	0.887	0.872	ICBI
DSC03144	FSIM					**0.502**	0.406		0.419	0.414	0.415	0.348	0.423	0.419	0.419	0.301	ICBI
	PSNR					**43.096**	39.045		39.911	40.502	39.76	38.82	40.045	40.678	39.881	39.123	ICBI
	RMSE					**0.007**	0.011		0.01	0.009	0.01	0.011	0.01	0.009	0.01	0.011	ICBI
	SSIM					**0.955**	0.901		0.915	0.925	0.912	0.896	0.918	0.928	0.915	0.903	ICBI
DSC03345	FSIM					**0.402**	0.378		0.373	0.373	0.369	0.355	0.374	0.372	0.37	0.262	ICBI
	PSNR					38.372	37.016		37.7	38.445	37.485	36.613	37.896	**38.589**	37.683	37.085	bilinear
	RMSE					0.012	0.014		0.013	0.012	0.013	0.015	0.013	**0.012**	0.013	0.014	bilinear
	SSIM					0.891	0.861		0.875	0.89	0.869	0.844	0.879	**0.893**	0.874	0.858	bilinear
DSC03366	FSIM					**0.438**	0.384		0.398	0.395	0.398	0.343	0.401	0.401	0.399	0.311	ICBI
	PSNR					**48.983**	47.288		47.382	47.88	47.242	46.558	47.696	48.236	47.539	46.989	ICBI
	RMSE					**0.004**	0.004		0.004	0.004	0.004	0.005	0.004	0.004	0.004	0.004	ICBI
	SSIM					**0.99**	0.985		0.986	0.987	0.985	0.982	0.986	0.988	0.986	0.984	ICBI
DSC03376	FSIM					**0.435**	0.394		0.371	0.374	0.372	0.327	0.375	0.377	0.375	0.294	ICBI
	PSNR					**42.714**	41.227		40.472	41.356	40.252	39.431	40.628	41.443	40.396	39.756	ICBI
	RMSE					**0.007**	0.009		0.009	0.009	0.01	0.011	0.009	0.008	0.01	0.01	ICBI
	SSIM					**0.962**	0.949		0.942	0.95	0.939	0.927	0.943	0.95	0.941	0.932	ICBI