MA797 Project Final Report: Image Super-resolution in Real-World

Jie Hu, Peiran Wang, Chanae Ottley

December 9th, 2021

1 INTRODUCTION

Single image super-resolution (SISR) is a classical problem in computer vision. It aims to obtain a high-resolution (HR) output from one of its low-resolution (LR) versions. This topic has many applications in the real world, e.g., video enhancement [11, 13], medical diagnosis [9, 6], etc. The super-resolution problem is inherently ill-posed since the super-resolution operator is a one-to-many mapping from LR to HR space, which can have multiple solutions [8]. One of the assumptions in the literature is that HR contains redundant information so that we can recover HR by utilizing the detailed information from the LR image itself. In other words, SISR mainly exploits the spatial redundancy within an image.

Prior to convoluted neural network (CNN) techniques, edge-based, image statistics-based, and patch-based methods have been widely studied [12]. Especially, the sparse-coding approach is one of the representative external patch-based SR methods, which involves several sequential steps to get the LR image to HR image mapping. [3] is one of the early works that connect this sparse coding approach with a CNN approach. Then, CNN models have been actively applied to the SISR application. In our project, we look at four different super-resolution methods for this project: Bicubic Interpolation, Super Resolution CNN [3], Sub-Pixel CNN [8] and Asymmetric CNN [10]. Since the bicubic interpolation is commonly mentioned in several publications and implemented into some methods, it is used as our baseline.

2 METHODS

In this section, we briefly introduce four super-resolution methods.

2.1 Bicubic Interpolation

Bicubic interpolation is a method of applying cubic interpolation to a dataset for interpolating data points on a two-dimensional regular grid. Since Bicubic interpolation can construct new data points within the range of a set of known data points, it is used for image scaling and has been widely used in photo editors, e.g., Adobe Photoshop. Bicubic utilizes the closest 4×4 neighborhood of known pixels to estimate one pixel. Closer pixels are usually given higher weights. In Figure 1, we give an example to interpolate a 2×2 matrix to 4×4 matrix and the scale factor is 2. We want to estimate the pixel value at index $(1, 2)$. First, we can add borders to the input matrix (we use a border replicate method here). Then, according to the position $(1, 2)$, we can find the column coefficient vector and row coefficient vector. These weights can be found in the OpenCV package. The last step is to do the normal matrix multiplication with highlight matrix/vector in Figure 1.

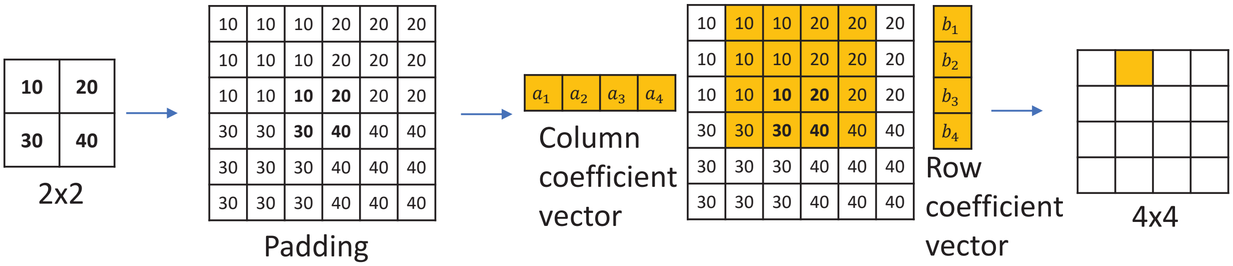


Figure 1: An example of Bicubic interpolation

2.2 Super Resolution CNN

Super Resolution CNN (SRCNN) [3] is the pioneer CNN model for SISR. Previous researches, e.g., dictionary learning [4] or random forest [7], tended to boost the super-resolution performance via complex optimization methods. These methods referred to manual tuning for parameters to obtain better results. As the pioneer, SRCNN first up-sampled LR image as input of the network through an end-to-end architecture to obtain the HR image. Although we can connect the traditional methods to CNN, not all operations have been considered in the optimization problem in the traditional methods. The advantage of this CNN method is that we do not need to manually solve the optimization problem to tune parameters and because the CNN would take all the possible operations into consideration.

In the learning process, SRCNN needs to first upscale LR images to the same size as the HR images by using the bicubic interpolation method.[1] Then, the first layer is to densely extract $f_1 \times f_1$ patch and map it onto n_1 features. Then, the second layer works as mapping from n_1 features of LR image to n_2 features of HR image with filter size $f_2 \times f_2$. The last layer works as an image reconstruction that 'flattened' vector by averaging the previous n_2 features with filter size $f_3 \times f_3$. All these three layers are convoluted 2d layers with additional bias. In this project, we use $f_1 = 9, f_2 = 3, f_3 = 5$ and $n_1 = 128, n_2 = 64$ as our model parameters.

2.3 Sub-Pixel CNN

We implemented the Efficient Sub-Pixel CNN [8], which is based on deep neural networks similar to the previously mentioned method. The novelty of this method is claimed to be the design of the CNN architecture that performs the super-resolution pipeline different from previous work. The Sub-Pixel CNN includes a 'sub-pixel convolution' layer that learns imaging filters. This layer allows the high-resolution image to be created by upscaling the final low-resolution feature maps. This method replaces the use of the bicubic filter for image prediction. Instead, it uses filters that were specifically trained for a feature map. The network has 3 convolution layers and 1 sub-pixel convolution layer, which is the network's last layer. The sub-pixel layer implements a fractional slide that activates different parts of the filters and aggregates the feature maps from the low-resolution

[1] This step arbitrarily chooses bicubic as a scaling method, resulting in suboptimality. In fact, the scaling process can also be integrated into the CNN structure and let CNN learn the best way to scale. This idea is studied in the following method Sub-Pixel CNN [8].

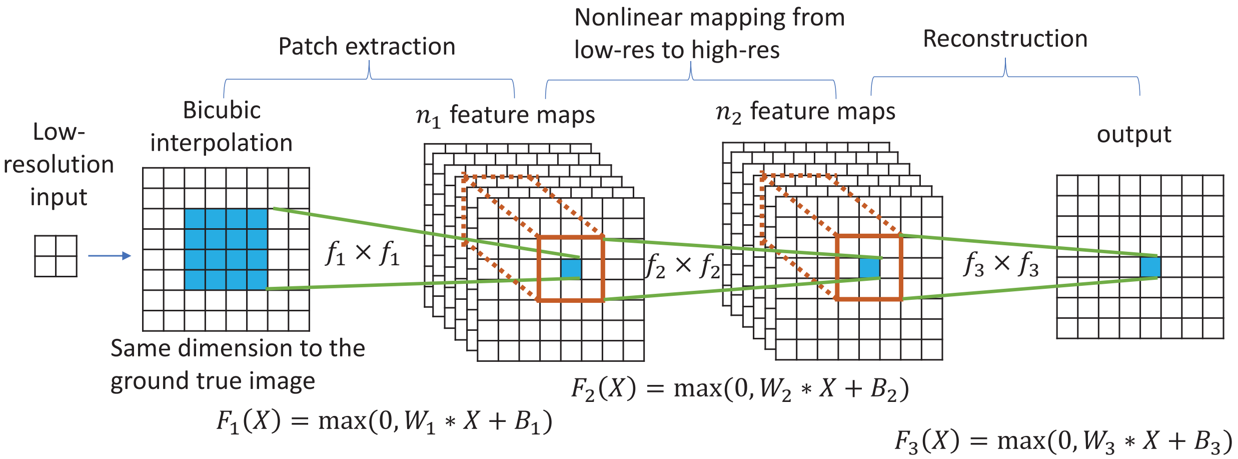

Patch extraction
Nonlinear mapping from low-res to high-res
Reconstruction
Low-resolution input
Bicubic interpolation
n_1 feature maps
n_2 feature maps
output
$f_1 \times f_1$
$f_2 \times f_2$
$f_3 \times f_3$
Same dimension to the ground true image
$F_1(X) = \max(0, W_1 * X + B_1)$
$F_2(X) = \max(0, W_2 * X + B_2)$
$F_3(X) = \max(0, W_3 * X + B_3)$

Figure 2: SRCNN structure

space to create the high-resolution image. Unlike previous work, this CNN only increases the image from low resolution to high resolution at the end of the sub-pixel layer. The overall computational complexity is reduced by performing the resolution change at the end.

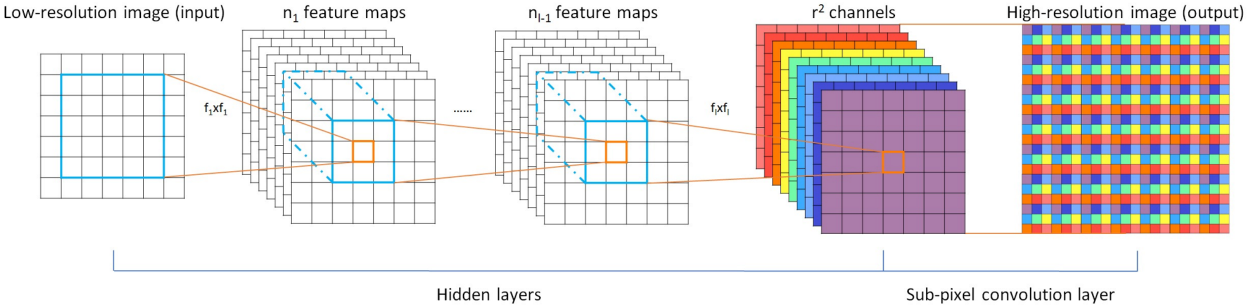


Figure 3: Sub-Pixel CNN structure

2.4 Asymmetric CNN

We also implemented the asymmetric CNN (ACNet) [10], which comprised of three blocks: an asymmetric block (AB), a memory enhancement block (MEB), and a high-frequency feature enhancement block (HFFEB).

A 17-layer asymmetric block (AB) utilizes one-dimensional asymmetric convolutions to intensify the square convolution kernels in the horizontal and vertical directions for improving the influences of local power feature points. It can accelerate the training for a super-resolution model and uses the lowest cost for computation. Then, we use a memory enhancement block (MEB) to resolve the long-term dependency problem and transforms obtained low-frequency features into high-frequency features. Memory enhancement block can be finished as three steps: the first step combines all hierarchical low-frequency features from AB via residues learning. The second step is using a sub-pixel convolutional layer to convert the low-frequency features into high-frequency features. The third step of MEB aims to avoid the loss of global input information. It uses sub-pixel convolutions to strengthen the output of the first layer in the asymmetric block. Finally, we use a 5-layer high-frequency feature enhancement block (HFFEB) to connect the gap between the predicted high-resolution image and the given high-resolution image.

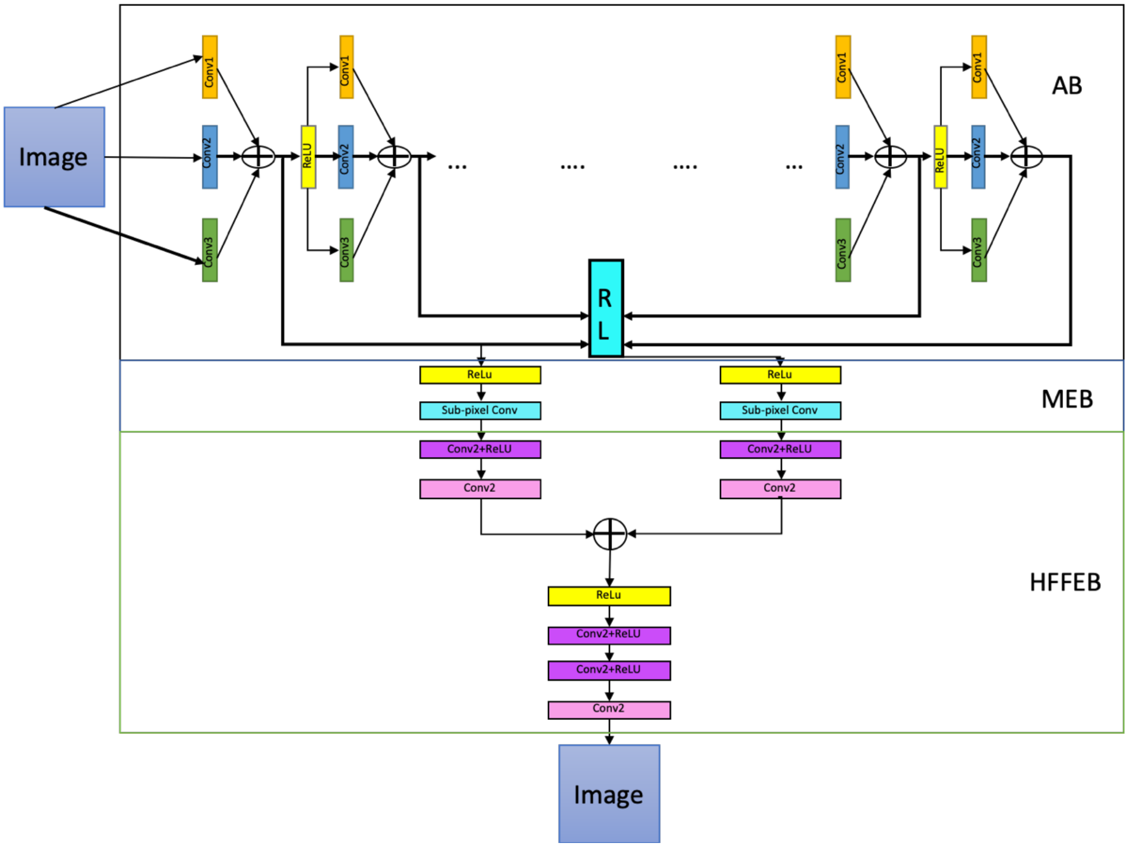


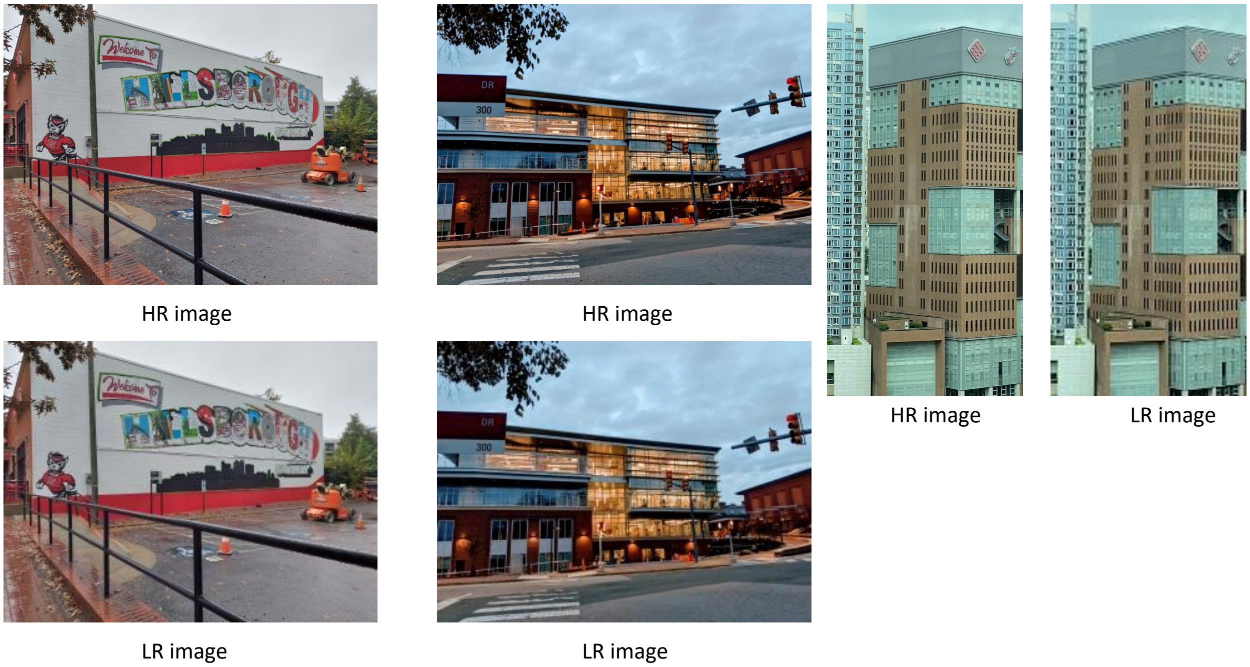
Figure 4: Asymmetric CNN structure

3 DATA

Previous works [3, 8, 13] tend to downsample the original image and add artificial noise (e.g., Gaussian noise) in the dataset, e.g., Set5, Set14 [1], ImageNet [2] and BSD300, BSD 500 [5], to generate the LR image and train the SR model. This hinders their performance in the real world, which also includes much more complicated degradations. [13] provided a real-world dataset with HR, LR images[2] captured from two cameras with 26mm-equivalent and 52-equivalent lens such that the LR image contains real-world noise. Besides, we create our own dataset as a test dataset to observe the performance of each model under a different dataset. Our dataset includes 10 images, and each is captured using a Samsumg Galaxy S21 Ultra phone with an ultra-wide lens and standard lens. The image taken using the ultra-wide lens serves as the LR image and the image taken using a standard lens serves as the HR image because the standard lens has a larger focal length that can capture scenes with finer details. The HR image is of size 1200 × 900 and LR image is of size 300 × 225. We borrow the post-process method in [13] to align the LR and HR images. Figure 5 shows some HR/LR examples that will be used in the next section to compare the performance of

[2] HR image with resolution 512 × 1024 and LR image with resolution 128 × 256.

different models.[3]


HR image

HR image

HR image

LR image

LR image

LR image

Figure 5: First two HR/LR pairs are from our own dataset and the last one is from Real-World dataset [13].

4 RESULTS

We first introduce three metrices used for our performance analysis. For our project, we use peak signal-to-noise ratio(PSNR), structural similarity index measure (SSIM), and mean squared error (MSE) as the metrics to evaluate the performance of each experiment.

The peak signal-to-noise ratio (PSNR) is commonly used in the field of image quality assessment. It is defined by the maximum possible pixel value (denote as L) and the mean squared error (MSE) between images. Given the ground truth X with a total of N pixels and its corresponding constructed image X_{SR}, the MSE and the PSNR can be calculated by the following equations:

$$MSE = \frac{1}{N}\sum_{i=1}^{N} ||X(i) - X_{SR}(i)||_2$$

$$PSNR = 10\log_{10}\frac{L^2}{MSE}$$

The structural similarity index measure (SSIM) focuses on measuring the structural similarity between images. It incorporates three relatively independent elements, including luminance, contrast, and structure. The definition of SSIM is as follows:

$$SSIM(X, X_{SR}) = \frac{(2\mu_X\mu_{SR} + C_1)(\sigma_{XX_{SR}+C_2})}{(\mu_X^2 + \mu_{X_{SR}}^2 + C_1)(\sigma_X^2 + \sigma_{X_{SR}}^2 + C_1)}$$

[3] The difference between HR and LR images is more evident if we zoom in a bit and we see that LR is more blurry than the HR image.

Where $C_1 = (k_1 L)^2$ and $C_2 = (k_2 L)^2$ are constants to avoid instability. The mean and the standard deviation of the ground truth X are denoted as μ_X and σ_X, respectively, and the mean and standard deviation of the constructed image X_{SR} are denoted as $\mu_{X_{SR}}$ and $\sigma_{X_{SR}}$. $\mu_{XX_{SR}}$ is the covariance between X and X_{SR}.

	Bicubic	SRCNN	Sub-Pixel CNN	ACNet
PSNR	21.7211	21.7875	22.945	22.2436
MSE	1473.5177	1443.4275	1180.6676	1540.4094
SSIM	0.6468	0.6463	0.6284	0.6725

Table 1: The average results of PSNR (dB), MSE and SSIM on the Real-World dataset [13].

The PSNR score is a variation of the MSE and concentrates on the pixel-by-pixel comparison. Concerning the PSNR and MSE metrics, the Sub-Pixel CNN outperformed the other methods. Instead of using traditional error summation methods, the SSIM models image distortion as a combination of three factors: loss of correlation, luminance distortion, and contrast distortion. Based on Table1, we got PSNR scores around 22. The results demonstrate that the ACNet performed because it has a higher SSIM score than the other models.

Next we show the prediction results of 3 test images from Figure 5 among four models. In Figure 6, 7 and 8 we zoom in the details of each image. They show that ACNet always has a better reconstruction of details in the image like windows or edges. It achieves sharper image compared to other 3 models. Although Sub-Pixel CNN has good performance result from table 1, the sub-pixel layer that aggregates pixel onto the final image leads to inconsistency and we feel the image generated by Sub-Pixel CNN looks like mosaic if we zoom in a bit. Bicubic interpolation creates blurry feeling and performs the worst because it doesn't learn any new information from the LR image itself.

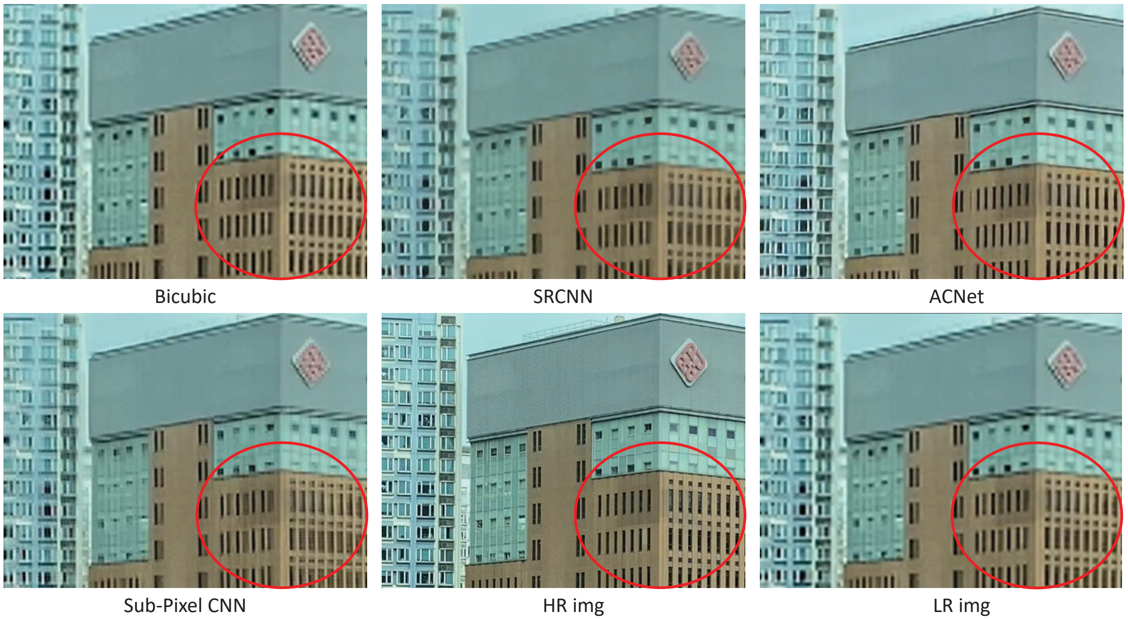


Figure 6: Super-resolution example 1 for four SR models from the Real-World dataset [13] with an upscaling factor of 4.

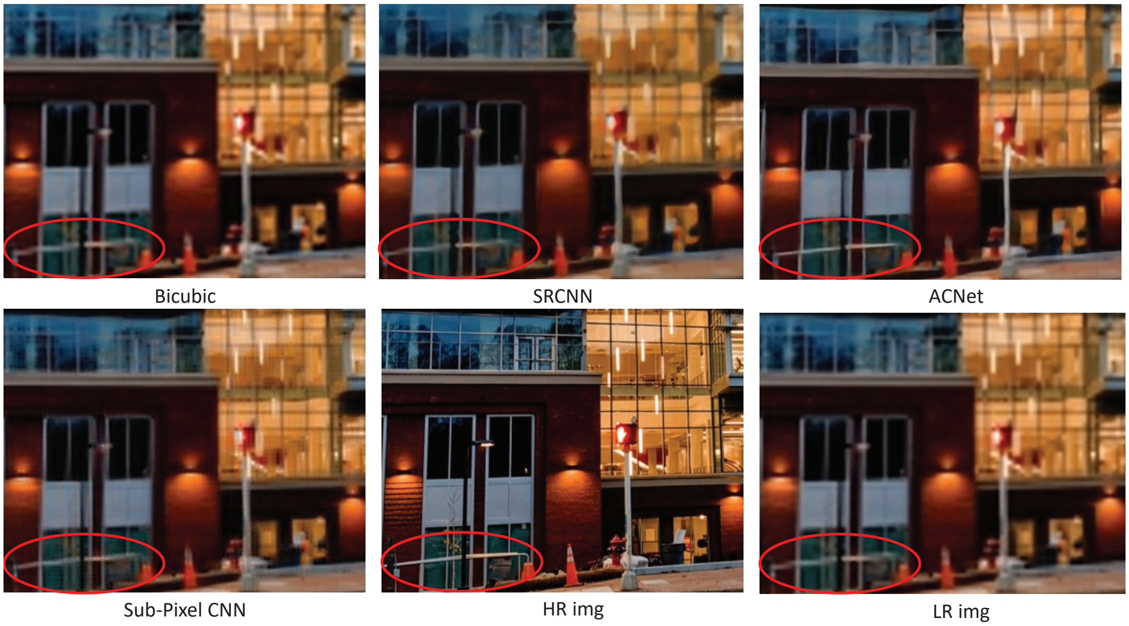


Figure 7: Super-resolution example 2 for four SR models with image taken in front of Carmichael Gym and an upscaling factor of 4.

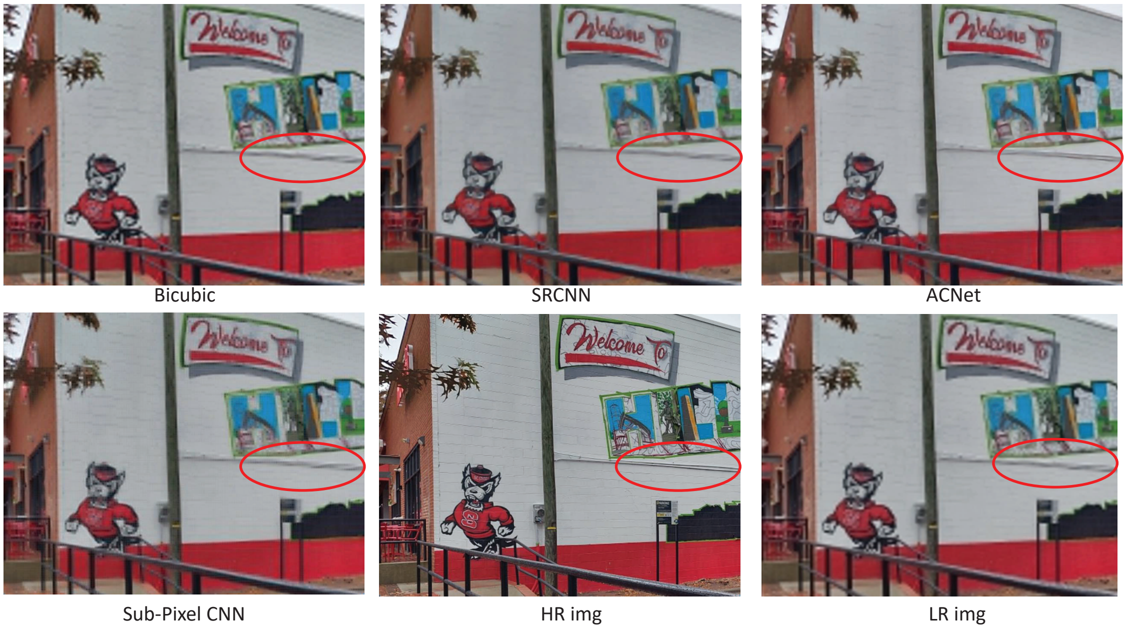


Figure 8: Super-resolution example 3 for four SR models with image taken on Hillsborough St. and an upscaling factor of 4.

5 CONCLUSIONS

In this project, we have built up Bicubic interpolation, SRCNN, Sub-Pixel CNN and the state-of-the-art ACNet models. These models were tested on the Real-World dataset and our dataset. Due to the advantage of residual learning and sub-pixel layer for reconstruction, ACNet has achieved the best results so far. However, we also find inconsistencies between the theoretical metrics and our visual interpretation. For example, the ACNet has a lower PSNR and higher MSE compared to SRCNN and Sub-Pixel CNN, but according to our experience, the ACNet provides a clearer image and the details are close to the HR image to the human eye. We have two future directions in mind: The first possible work is to design the layer that specifically deals with the real-world noise; The other direction is to analyze a better metric that measures the good performance of predicted image and HR image while also reflects users' preference.

References

[1] Yunjin Chen and Thomas Pock. Trainable nonlinear reaction diffusion: A flexible framework for fast and effective image restoration. *IEEE transactions on pattern analysis and machine intelligence*, 39(6):1256–1272, 2016.

[2] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[3] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Image super-resolution using deep convolutional networks. *IEEE transactions on pattern analysis and machine intelligence*, 38(2):295–307, 2015.

[4] Honglak Lee, Alexis Battle, Rajat Raina, and Andrew Y Ng. Efficient sparse coding algorithms. In *Advances in neural information processing systems*, pages 801–808, 2007.

[5] David Martin, Charless Fowlkes, Doron Tal, and Jitendra Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, volume 2, pages 416–423. IEEE, 2001.

[6] Sharon Peled and Yehezkel Yeshurun. Superresolution in mri: application to human white matter fiber tract visualization by diffusion tensor imaging. *Magnetic Resonance in Medicine: An Official Journal of the International Society for Magnetic Resonance in Medicine*, 45(1):29–35, 2001.

[7] Samuel Schulter, Christian Leistner, and Horst Bischof. Fast and accurate image upscaling with super-resolution forests. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3791–3799, 2015.

[8] Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1874–1883, 2016.

[9] Wenzhe Shi, Jose Caballero, Christian Ledig, Xiahai Zhuang, Wenjia Bai, Kanwal Bhatia, Antonio M Simoes Monteiro de Marvao, Tim Dawes, Declan O'Regan, and Daniel Rueckert. Cardiac image super-resolution with global correspondence using multi-atlas patchmatch. In *International conference on medical image computing and computer-assisted intervention*, pages 9–16. Springer, 2013.

[10] Chunwei Tian, Yong Xu, Wangmeng Zuo, Chia-Wen Lin, and David Zhang. Asymmetric cnn for image superresolution. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2021.

[11] Tianfan Xue, Baian Chen, Jiajun Wu, Donglai Wei, and William T Freeman. Video enhancement with task-oriented flow. *International Journal of Computer Vision*, 127(8):1106–1125, 2019.

[12] Chih-Yuan Yang, Chao Ma, and Ming-Hsuan Yang. Single-image super-resolution: A benchmark. In *European conference on computer vision*, pages 372–386. Springer, 2014.

[13] Xi Yang, Wangmeng Xiang, Hui Zeng, and Lei Zhang. Real-world video super-resolution: A benchmark dataset and a decomposition based learning scheme. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4781–4790, 2021.