Single Image Super-Resolution: Analyze and Implementation

NGUYEN DANG DUY - 20210272
ICT, HUST, Hanoi, Vietnam

HOANG VAN KHANG - 20210466
ICT, HUST, Hanoi, Vietnam

AND

BUI DUC VIET - 20215254
ICT, HUST, Hanoi, Vietnam

Abstract Single Image Super-Resolution (SISR) is one of vital tasks in computer vision that aims to reconstruct a high-resolution (HR) image from a low-resolution (LR) counterpart. This process has significant applications in various fields such as medical imaging, satellite imagery, and surveillance systems. In this report, we explore and present some solutions to the problem of SISR, including traditional image processing techniques, Convolutional Neural Networks (CNNs), and Generative Adversarial Networks (GANs). We evaluate these methods using public datasets such as Set5, Set14, and BSD100, as well as a synthetic dataset we created to incorporate realistic noise and blur effects. Our analysis utilizes key metrics such as Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index (SSIM), and Mean Opinion Score (MOS) to measure the performance and quality of the generated HR images. Through comprehensive experiments, we assess the strengths and weaknesses of each approach, providing valuable insights into their practical applications and effectiveness.

Keywords: Computer Vision; Super-Resolution; Deep Learning

1. Introduction

Single-Image Super-Resolution (SISR) [10, 17] is a computer vision task that reconstructs a high-resolution (HR) image from a low-resolution (LR) image. It could be used in a variety of applications such as medical imaging, security, and surveillance imaging. The quality of the reconstructed HR image depends on how to extract and use the information from LR image. Since there are multiple HR images that can be downsampled to the same LR image and this is a one-to-many mapping relation to recover HR images from a LR image, SISR is an ill-posed and still challenging problem in the community In this project, we will explore and present prominent solutions for the problem of Single-image Super-resolution (SISR). These solutions include Image Processing techniques, CNNs, and GANs models. Additionally, we will experiment and evaluate the performance of these methods on the same datasets, thereby assessing the strengths and weaknesses of each approach in different scenarios.

2. Problem details

Single Image Super-Resolution (SISR) is a classical problem in the field of computer vision and image processing. It has many application in many aspect of our life such as improving quality of medical image, satellite image and security footage,...In this part, we will introduce some key points of this problem.

2.1. *Image Super-Resolution*

- **Low-resolution (LR) image:** An image with a lower pixel count, resulting in less detail and more visible artifacts.
- **High-resolution (HR) image:** The desired output image with a higher pixel count, greater detail and fewer artifacts.

The goal of SISR is to enhance the resolution of a given low-resolution (LR) image to produce a high-resolution (HR) image.

2.2. *Dataset*

2.2.1. Public test dataset: Set5, Set14, BSDS100 [1]

The Set5 dataset [2] is commonly used in the field of image super-resolution to benchmark and evaluate the performance of super-resolution algorithms. This dataset contains five high-resolution images, each of which is accompanied by its corresponding low-resolution versions. The images in the Set5 dataset are diverse, including natural scenes and human faces, providing a well-rounded test set for super-resolution tasks.

The Set14 dataset [18] is another widely used benchmark in the image super-resolution field. It includes 14 diverse images that provide a robust testing ground for evaluating the performance of super-resolution algorithms. Similar to the Set5 dataset, Set14 includes images of different types, such as natural scenes, urban landscapes, and human faces.

The BSD100 dataset [13], part of the Berkeley Segmentation Dataset (BSD), is another standard benchmark used in the image super-resolution field. It consists of 100 high-resolution images, which provide a diverse and comprehensive set for evaluating super-resolution algorithms. The BSD100 dataset is known for its variety and complexity, making it a challenging test set for super-resolution methods.

The report relies on these three datasets to accurately assess and compare the performance of various methods. They serve as a crucial foundation for evaluating the effectiveness and efficiency of the methods and algorithms employed in this study.

2.2.2. Public train dataset: DIV2K, Flickr2K

We used DIV2K [1] to train our CNNs models, and Flickr2K [11] were used to train GANs models by other researchers.

DIV2K (Diverse 2K Resolution) is a high-quality dataset specifically designed for image super-resolution tasks. It contains 1000 high resolution images, and often be used to train models for super resolution problems.

Flickr2K is often used to train GANs models. There are 2000 high resolution images in Flickr2k, twice as much as DIV2K. Because of the number of its data. Flickr2k is suitable for training big model like GANs.

2.2.3. Synthetic dataset

We contributed a new dataset[2] to this report by extracting over 1,200 high-resolution images with a size of 2040x1356 from various 4k super-resolution videos available on the Internet, predominantly from

---

[1] All of these datasets can be accessed in here

[2] The synthetic dataset can be accessed here

YouTube. These high-resolution images were then processed through a degraded pipeline to generate the low-resolution images for the dataset. We experimented with three main degraded algorithms:

- Downsampled using Bicubic interpolation
- Downsampled using Bicubic interpolation and apply an extra blur kernel
- Downsampled using Bicubic interpolation plus addictive white Gaussian noise

In addition to the commonly used Bicubic interpolation method, our proposal includes two additional challenging datasets. These datasets are affected by blur or noise, providing a greater diversity in training data. By incorporating these realistic elements such as noise and artifacts, our datasets offer a more accurate representation of images captured in the real life.

2.3. *Evaluation metric*

Evaluation metrics are essential tools for assessing the performance of machine learning models and image processing techniques. Choosing the right metric depends on the specific problem and the desired outcomes. Below are some commonly used evaluation metrics across different domains. In this problem, evaluation metrics is used to evaluate the similar of output image with the ground-truth high-resolution image.

2.3.1. Peak signal to noise(PSNR)

Peak Signal-to-Noise Ratio (PSNR): is a widely used metric to measure the quality of a reconstructed or compressed image compared to the original image. It is commonly used in the fields of image processing and video compression to assess the performance of compression algorithms and image enhancement techniques, such as single image super-resolution. The Peak Signal-to-Noise Ratio (PSNR) is defined as:

$$\text{PSNR} = 10 \cdot \log_{10} \left(\frac{\text{MAX}^2}{\text{MSE}} \right) \tag{2.1}$$

where:

- MAX: maximum possible value of pixel in the image. It can be 1 if value in range [0, 1] or 255 if value in range [0, 255]
- MSE: (Mean Squared Error) is the average of the squared differences between the pixel values of the original and the reconstructed images. It can be calculated by:

$$\text{MSE} = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [I(i,j) - K(i,j)]^2 \tag{2.2}$$

In these formulas:

- MAX is the maximum possible pixel value of the image.
- $I(i,j)$ is the pixel value at position (i,j) in the original image.
- $K(i,j)$ is the pixel value at position (i,j) in the reconstructed image.
- m and n are the dimensions of the images.

2.3.2. Structural similarity index measure (SSIM)

Structural Similarity Index (SSIM): is a metric used to measure the similarity between two images. It aims to model the perceived change in structural information, providing a more accurate assessment of image quality compared to traditional metrics such as mean squared error (MSE) or peak signal-to-noise ratio (PSNR). SSIM is particularly useful in image processing tasks such as image compression, transmission, and enhancement.

The Structural Similarity Index (SSIM) between two images x and y is given by:

$$\text{SSIM}(x,y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}$$

where:

μ_x	mean of x	σ_x^2	variance of x
μ_y	mean of y	σ_y^2	variance of y
σ_{xy}	covariance of x and y	C_1, C_2	constants

K_1 and K_2 are constants, and L is the dynamic range of the pixel values (typically $2^{\text{bits per pixel}} - 1$).

2.3.3. Mean opinion score (MOS)

Mean opinion score (MOS): The Mean Opinion Score (MOS) is a measure used to evaluate the quality of speech, audio, or video, based on subjective assessment by human participants. It is widely used in telecommunications and multimedia to assess the perceived quality of services such as voice calls, streaming audio, and video content.

The MOS is typically calculated by asking a group of users to rate the quality of a sample on a predefined scale, often from 1 to 5, where:

- 1 = Bad
- 2 = Poor
- 3 = Fair
- 4 = Good
- 5 = Excellent

The scores from all participants are then averaged to produce the MOS. The result is a single numerical value that represents the overall perceived quality of the sample.

3. Methods and algorithms

3.1. *Classical approaches - Image processing*

Years ago, people started using pure-math methods for making higher resolution images. Interpolation methods were considered since it reconstructs the new image based on the pixel values in the old one. Four most popular methods will be introduced, including Nearest Neighbor, Bilinear and Bicublic interpolation, from the order of increasing complexity, and quality.

3.1.1. Nearest Neighbor

The idea behind the Nearest Neighbor algorithm 1 is to find the nearest neighboring pixel in the original image and replicate its value to the corresponding pixel in the resized image.

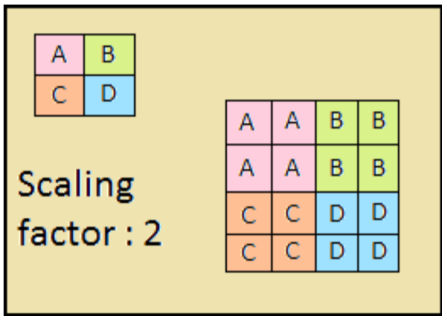
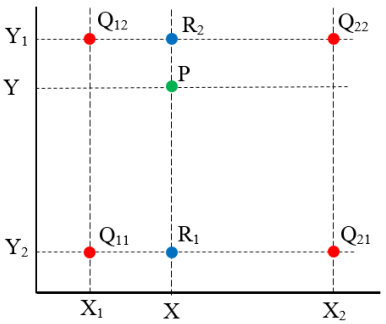


FIG. 1. Nearest Neighbor Explain

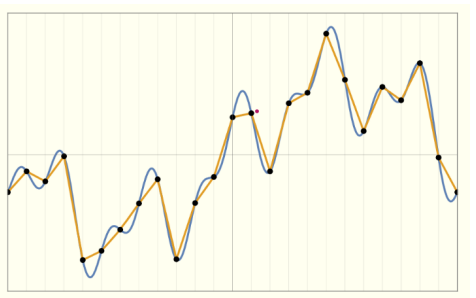
The algorithm is straightforward to implement and computationally efficient, making it suitable for real-time applications where speed is crucial. Due to its low computational complexity, nearest neighbor interpolation is very fast and requires minimal processing power and memory.
However, this method does not provide smooth transitions between pixels and can result in jagged edges and pronounced artifacts.

3.1.2. Bilinear Interpolation

To fill in the pixel values for a region, we will use 4 origin pixel in the corner to make interpolation. Let Q_{11}, Q_{12}, Q_{21}, Q_{22} are 4 original pixels 2a.



(a) Bilinear Interpolation explain [5]



(b) Interpolation using linear equations

FIG. 2. Bilinear Algorithms

First, we use a linear equation to describe the line $Q_{11}Q_{21}$. Every pixel value of point in $Q_{11}Q_{21}$ can be located by the equation:

$$R_1 = f(x_1, y_1)\frac{x_2 - x}{x_2 - x_1} + f(x2, y1)\frac{x - x_1}{x_2 - x_1}$$

Similarly, every pixel value of point in $Q_{12}Q_{22}$ can be located by the equation:

$$R_2 = f(x_1, y_2)\frac{x_2 - x}{x_2 - x_1} + f(x2, y2)\frac{x - x_1}{x_2 - x_1}$$

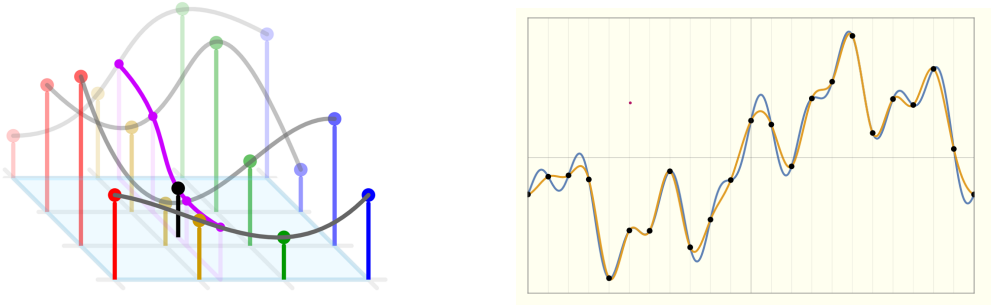
Therefore, every point that located in the line R_1R_2 has pixel value calculated by the equation:

$$f(x,y) = R_1 \frac{y_2 - y}{y_2 - y_1} + R_2 \frac{y - y_1}{y_2 - y_1}$$

Bilinear interpolation provides smoother images compared to nearest neighbor by reducing the blockiness and creating more natural transitions between pixels.
While more complex than nearest neighbor, bilinear interpolation is still relatively simple and fast, making it suitable for many real-time applications.
However, this method tends to blur the image slightly, which can lead to a loss of sharpness and detail, particularly in high-frequency regions.

3.1.3. Bicubic Interpolation

While Bilinear Interpolation uses only 4 origin pixels to interpolates, Bicubic uses 16, since we need 4 points to establish a cubic equation. The methods is as same as bilinear, except using cubic functions 3a.



(a) Bicubic Interpolation explain [5]

(b) Interpolation using cubic equations

FIG. 3. Bicubic Algorithms

Bicubic interpolation produces smoother and more visually appealing images with better edge preservation and fewer artifacts compared to nearest neighbor and bilinear methods. This method is particularly good at maintaining the sharpness and fine details in the image, making it suitable for applications where high-quality output is essential.
For that reason, Bicubic Interpolation becomes one of the most common methods for image rescaling problem. However, sometimes we want to reconstructed more realistic images due to the requirement of the project, or the job. That is why many other methods were suggested such as using Convolutional Neural Networks, or Generative Networks.

3.2. *Learning-based approaches - CNNs*

Initially, the Single-Image Super Resolution was typically solved using interpolation methods such as Nearest-neighbor Interpolation, Bilinear Interpolation, Bicubic Interpolation. When the scale factor between the HR image and its LR counterpart surpasses 2, this curve-fitting process results in very smooth images, devoid of sharp edges and sometimes, with “artifacts.” This is because an interpolation technique is not, in fact, adding any new information to the signal. This can be changed with the aid of

learning-based techniques. The idea is that if a neural network is provided with enough LR-HR pairs, it will be able to recognize the obscured entities in the LR images and reconstruct them based on the samples it's seen during training. Deep convolutional neural networks are an obvious candidate for the job, given their outstanding success in image processing problems.
In their 2014 publication, Chao Dong and his colleagues introduced the first learning-based method for SISR, a deep convolutional neural network that came to be known as **SRCNN** [3]. After that, there were many solutions based on CNN published. Super-resolution images generated by CNN achieved higher PSNR values than the Bicubic interpolation algorithm. In this report, we will introduce and implement 3 CNN: **SRCNN** [3], **ESPCN** [14] and **VDSR** [6].

3.2.1. SRCNN - Super resolution convolutional neural network

- **Idea:** First upscale the low-resolution image to the desired size using bicubic interpolation. We need to learn a mapping F to map the interpolated image with the ground-truth high-resolution image. This task conceptually consists of three operations:
  - Patch extraction and representation
  - Nonlinear mapping
  - Reconstruction

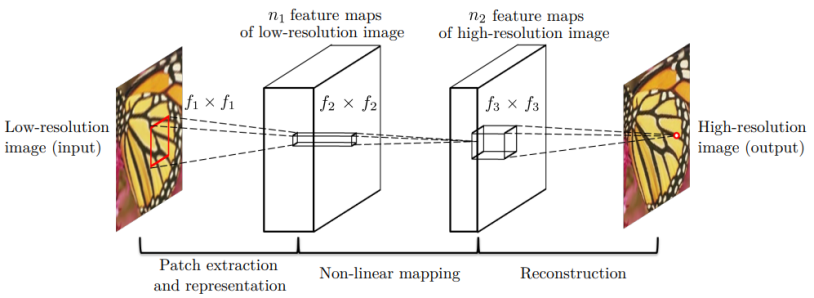


FIG. 4. Illustation of SRCNN architecture

- **Detail architecture:**
  - The first layer extracts n1-dimensional features for each part.
  - Applying n_2 filters which have a trivial spatial support 1×1 to map each of this n_1-dimensional vectors to n_2-dimensional vectors
  - Larger filter 3x3, 5x5 can be applied instead of 1x1 filter
  - Each of the output n2-dimensional vectors is conceptually a representation of a high-resolution patch that will be used for reconstruction.
- **Loss function:** MSE Loss - Mean Square Error The Mean Squared Error (MSE) is defined as:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(F(Y_i) - X_i)^2 \tag{3.1}$$

where $F(Y_i)$ is the predicted value and X_i is the ground truth value for the i-th sample.

- **Training details:** We divided the **DIV-2K** dataset into two parts: one part for training and the other for testing. The model was trained using the Adam optimization algorithm with a learning rate of 0.0001 and trained over 40 epochs.

3.2.2. ESPCN: Efficient Sub-pixel Convolutional Neural Network

- **Idea:** The proposed method avoids upscaling the LR image before feature extraction. Instead, it applies a multi-layer convolutional network directly to the LR image, followed by a sub-pixel convolution layer that upscales the LR feature maps to produce the HR image. This design reduces computational and memory complexity, enabling real-time super-resolution of HD videos .
- **Detail architecture:**

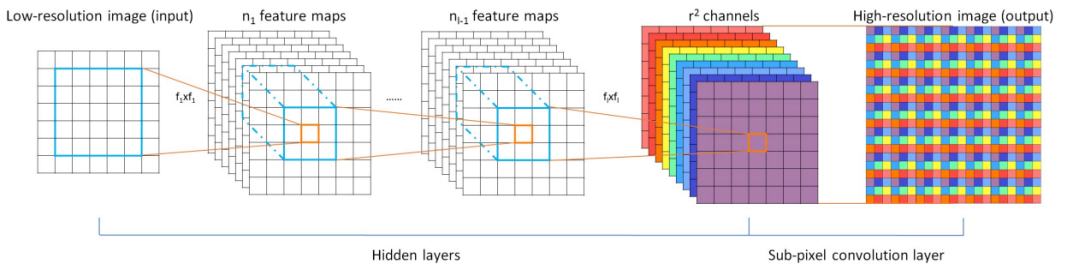


FIG. 5. Illustration architecture of ESPCN

(L-1) first layers are convolutional layers which obtain features map of the input LR images. The last layer is an efficient sub-pixel convolutional layer to recover the output image size with the specified upscale factor.

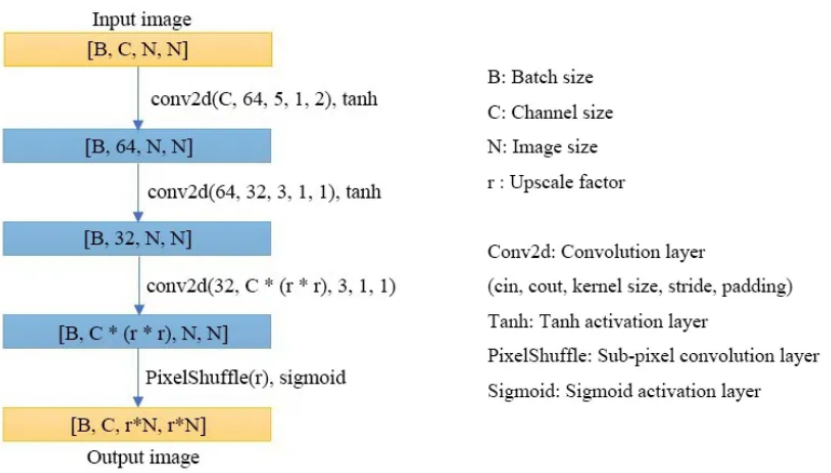


FIG. 6. ESPCN model

Sub-pixel convolutional Due to the limitation of the light sensor, images are limited to the original pixel resolution. In the digital image we saw, pixels and pixels are connected together, while in the microscopic world there are numbers of tiny pixels between the two physical pixels. Those tiny pixels are called sub-pixels.

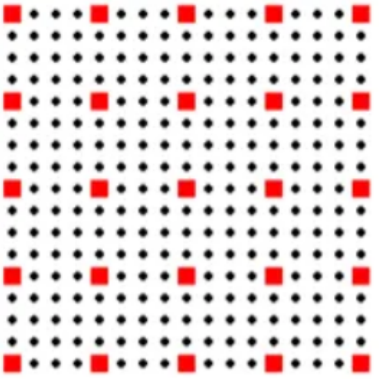


FIG. 7. Visualization of sub-pixel

Each square area surrounded by four little red squares is the pixel in the imaging plane of the camera, the black dots are sub-pixels. In this way, the mapping from small square areas to big square areas can be implemented through sub-pixel interpolation. Performing pixel-shuffle at the last layer of the network to recover the LR image does not need padding operation. As shown in Figure 8, combining each pixel on multiple-channel feature maps into one r × r square area in the output image. Thus, each pixel on feature maps is equivalent to the sub-pixel on the generated output image.

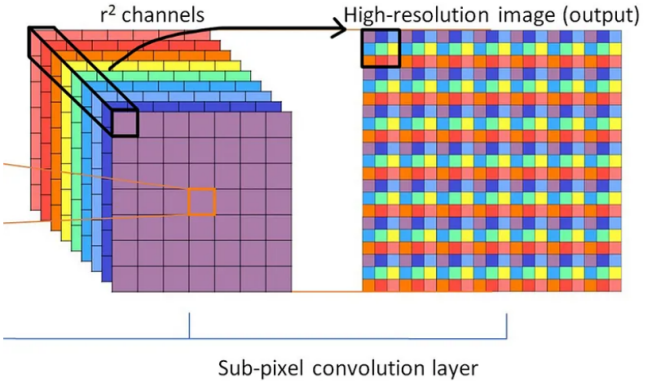


FIG. 8. Operation of pixel shuffle

- **Loss function:** Mean Square Error (MSE Loss)

- **Training details:** The training parameters of ESPCN is similar to the training set up of SRCNN. The DIV-2K dataset was used for training, 80% of the dataset for training and the remaining part for testing. Models were trained using Adam optimization algorithm with a learning rate of 0.0001 and trained over 100 epochs.

3.2.3. VDSR: Very Deep Super Resolution

- **Idea:** Inspired by the VGG-net used for ImageNet classification, the authors propose a 20-layer network that significantly improves super-resolution accuracy by leveraging contextual information over large image regions.
- **Detail architecture:** A 20-layer very deep convolutional network is used, with each layer employing 3x3 filters. The network predicts residual images (the difference between HR and LR images) rather than the HR images directly. Zero-padding is applied to maintain the same size for all feature maps, including the output image.

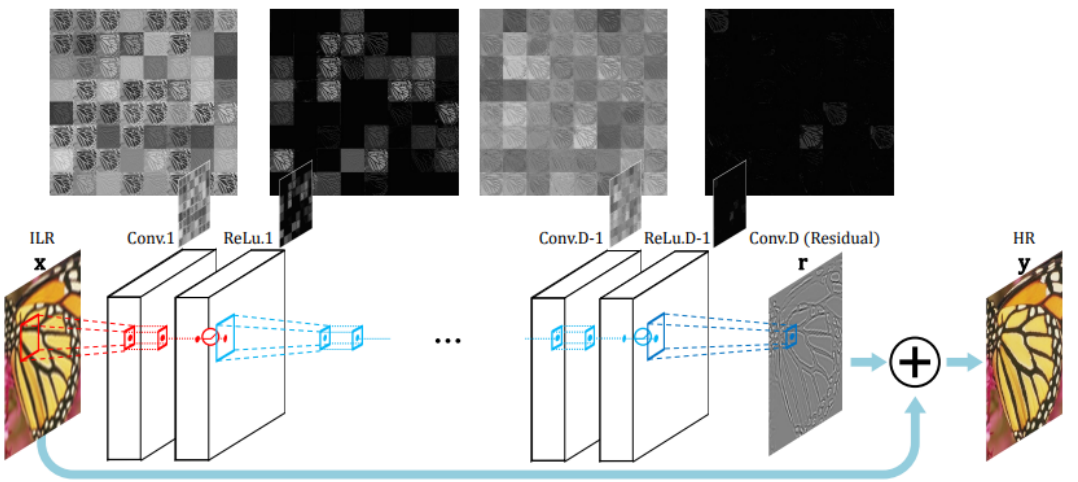


FIG. 9. Operation of pixel shuffle

- ***Residual-learning***
  In SRCNN, the exact copy of the input has to go through all layers until it reaches the output layer. With many weight layers, this becomes an end-to end relation requiring very long-term memory. For this reason, the vanishing/exploding gradients problem can be critical. VDSR can solve this problem simply with residual learning. Because the input and output image are largely similar, a residual image is defined as **r = y - x**.

- **Loss function:** Mean Square Error (MSE Loss)
  In VDSR, we want to predict this residual image. This loss function now becomes:

$$L = \frac{1}{2}\|r - F(x)\|^2, \tag{3.2}$$

In networks, this is reflected in the loss layer as follows. Our loss layer takes three inputs: residual estimate, network input (ILR image) and ground truth HR image.

- **With VDSR:** We devided image in T291 datasets into 41x41x3 patch for training and testing. Model was trained using Adam optimization algorithm with a high learning rate of 0.1 and then decreased by a factor of 10 every 20 epochs. Training uses batches of size 64 over 80 epochs.

3.3. *Modern approaches - Generative models*

Due to the extraordinary advancements in Deep Learning, generative models have emerged as a powerful and influential tool in the field of Machine Learning. These models offer a wide range of innovative solutions that effectively address various tasks. One of the most notable achievements of generative models is their ability to generate realistic images, which has revolutionized the field of computer vision. By learning the underlying patterns and structures of image datasets, generative models can generate new images that are visually indistinguishable from real ones. This has opened up a plethora of applications, ranging from art and design to data augmentation in machine learning pipelines. There have been countless proposals regarding Single Image Super-Resolution (SSIR), all of them leading to astonishing results. We thoroughly analyze a range of renowned and exemplary models from each class for our project. Specifically, we consider Denoise Super-Resolution Variational AutoEncoder or dSRVAE [12] as a representative VAE model, Super-Resolution Generative Adversarial Network - SRGAN [9] and its improved variation Real - Enhance Super-Resolution Generative Adversarial Network - Real-ESRGAN [15] as exceptional GAN models.

3.3.1. Denoise Super-Resolution Variational AutoEncoder - dSRVAE

A Variational AutoEncoder or VAEs [7, 8] consist of an encoder and a decoder. The encoder maps high-dimensional input data into a low-dimensional representation, while the decoder attempts to reconstruct the original high-dimensional input data by mapping this representation back to its original form. The encoder outputs the normal distribution of the latent code as a low-dimensional representation by predicting the mean and standard deviation vectors. The decoder uses the output of the encoder to generate synthetic data.

- **Idea:** dSRVAE is a combination of Variational AutoEncoder (VAE) and Generative Adversarial Network (GAN) to jointly perform image denoising and super-resolution. This approach addresses the challenge of real-world images often containing noise and artifacts that degrade the quality of super-resolution.
- **Detail architecture:** The proposed dSRVAE model is composed of two main components: The Denoising AutoEncoder (DAE) is responsible for removing noise from the input low-resolution images. It uses a conditional variational autoencoder (CVAE) architecture to encode the noisy input into a latent space and then decode it to reconstruct a clean image. The encoder compresses the noisy image to learn latent variables, while the decoder uses these variables to estimate and remove the noise. After denoising, the clean image is upsampled to the desired high resolution using the Super-Resolution Sub-Network (SRSN). The SRSN consists of hierarchical residual blocks that refine the upsampled image to enhance its resolution. Bicubic interpolation is initially used to match the image dimensions, followed by the SRSN to add fine details and textures. A discriminator is incorporated to guide the SRSN in generating photo-realistic images. The GAN framework ensures that the super-resolved images have high perceptual quality by distinguishing between real high-resolution images and generated ones.

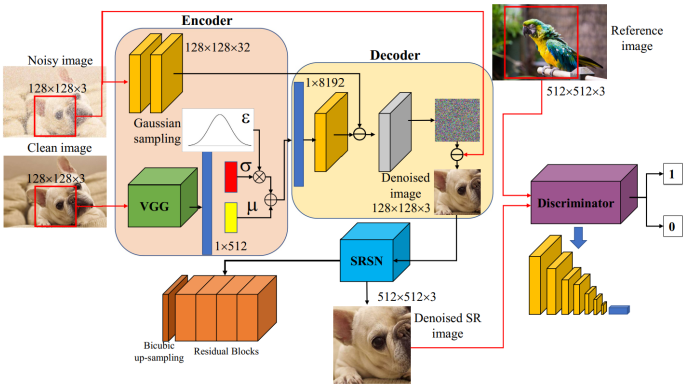


FIG. 10. Complete structure of the proposed dSRVAE model. It includes Denoising AutoEnocder (DAE) and Super-Resolution SubNetwork (SRSN). The discriminator is attached for photo-realistic SR generation

- **Training strategy:** To stabilize the adversarial training and balance the reference and super-resolved images, a cycle training strategy is employed. This strategy involves back-projecting the super-resolved image to the low-resolution space and comparing it with the original low-resolution image to refine the super-resolution process. For the training detail, the learning rate was set to 0.0001 for all layers. The batch size was set to 16 for $1x10^6$ iterations. For optimization, Adam optimizer is used with the momentum equal to 0.9 and the weight decay of 0.0001.
- **Loss function:** The training involves multiple loss functions to optimize the model:
  The reconstruction loss to ensure the denoised image is accurate

$$L_{MAE} = \sum_{c=1}^{C}\sum_{h=1}^{H}\sum_{w=1}^{W} \left|s(\mathbf{Y}_{c,h,w}) - g(\mathbf{X}_{c,h,w})\right| + \left|\mathbf{Y}_{c,h,w} - \bar{\mathbf{Y}}_{c,h,w}\right| \quad \text{where } \bar{\mathbf{Y}} = f(s(\mathbf{Y})), \mathbf{Y} = f(g(\mathbf{X})) \tag{3.3}$$

  where L_{MAE} is the pixel based Mean Absolute Errors MAE, f and g are the SRSN and DAE parameters, C, H and W are the size of SR images. s is the down-sampling operator with Bicubic process for simplicity. $\mathbf{Y}$ is the output SR image and $\bar{\mathbf{Y}}$ is the back projected SR image.
  The perceptual loss using pre-trained VGG networks to maintain high-level features and textures and the adversarial loss to encourage the generation of photo-realistic images

$$\mathbf{L} = \lambda \left\|\phi_i(f(g(\mathbf{X}))) - s(\phi_i(g(\mathbf{X})))\right\|_1^1 + \eta \log\left[1 - D_{\theta_D}(G_{\theta_G}(g(\mathbf{X})))\right] + L_{MAE} \tag{3.4}$$

  where λ and η are two weighting parameters to balance the VGG feature loss and adversarial loss. θ_G and θ_D are the learnable parameters of the generator and discriminator, respectively. θ_i represents the features from the i-th convolutional layer.

3.3.2. Generative Adversarial Network for Super-Resolution

GANs [4] learn to generate new data similar to a training dataset. It consists of two neural networks, a generator, and a discriminator, that play a two-player game. The generator takes in random values

sampled from a normal distribution and produces a synthetic sample, while the discriminator tries to distinguish between the real and generated sample. The generator is trained to produce realistic output that can fool the discriminator, while the discriminator is trained to correctly distinguish between the real and generated data.

- **Idea:** SRGAN is designed to generate high-resolution (HR) images from low-resolution (LR) inputs. It leverages the power of Generative Adversarial Networks (GANs) to create images that are perceptually more realistic compared to those produced by traditional methods which optimize for pixel-wise accuracy (e.g., MSE). SRGAN was the 1st GAN models proposed for SISR tasks. SRGAN is able to recover the finer texture details of the low-resolution images, results in a more realistic super-resolution images.
- **Detail architecture:**

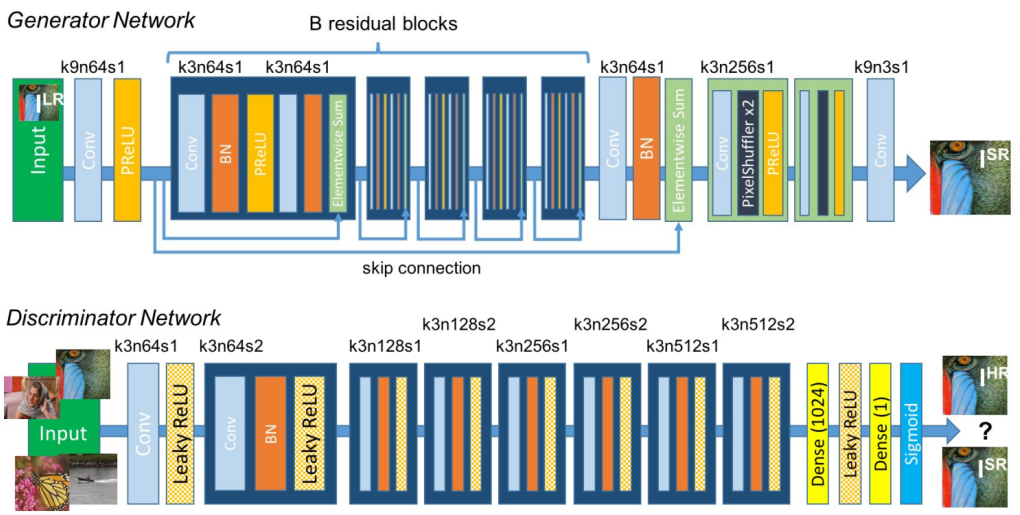


FIG. 11. Architecture of Generator and Discriminator Network with corresponding kernel size (k), number of feature maps (n) and stride (s) indicated for each convolutional layer.

Generator (G): This is a deep residual network (ResNet) with 16 residual blocks. The generator takes a low-resolution image and generates a high-resolution counterpart. It employs skip-connections to ease the training of deeper networks and improve the quality of the generated images. **Discriminator** (D): This network differentiates between real high-resolution images and those generated by the generator. It is trained to distinguish between the super-resolved images and the real ones, pushing the generator to create more realistic images.

- **Adversarial Training:** The generator and discriminator are trained in an alternating fashion. The generator aims to minimize the perceptual loss, while the discriminator aims to maximize its ability to distinguish between real and generated images. The adversarial loss guides the generator towards producing images that reside on the manifold of natural images, thereby enhancing realism.
- **Perceptual Loss Function:** The loss function for SRGAN is the weighted sum of a **content loss** and an **adversarial loss** component.

Content Loss: Computed based on the differences in high-level feature maps extracted from a pre-trained VGG19 network, rather than just pixel-wise differences. This approach ensures that the generated images retain perceptually important features and textures.

$$I^{SR}_{VGG/i,j} = \frac{1}{W_{i,j}H_{i,j}} \sum_{x=1}^{W_{i,j}} \sum_{y=1}^{H_{i,j}} \left(\phi_{i,j}(I^{HR})_{x,y} - \phi_{i,j}(G_{\phi_G}(I^{LR}))_{x,y}\right)^2 \tag{3.5}$$

where $W_{i,j}$ and $H_{i,j}$ describe the dimensions of the respective feature maps within the VGG network.
Adversarial Loss: Encourages the generator to produce images that are indistinguishable from real images by the discriminator.

$$I^{SR}_{Gen} = \sum_{n=1}^{N} -\log D_{\theta_D}(G_{\theta_G}(I^{LR})) \tag{3.6}$$

where $D_{\theta_D}(G_{\theta_G}(I^{LR}))$ is the probability that the reconstructed image $G_{\theta_G}(I^{LR})$ is a natural HR image. For better gradient behavior, the author suggests minimizing $-\log D_{\theta_D}(G_{\theta_G}(I^{LR}))$ instead of $\log(1 - D_{\theta_D}(G_{\theta_G}(I^{LR})))$.

3.3.3. Improved version of SRGAN: Real-ESRGAN

Real-ESRGAN is a significant improved version of SRGAN, which is very popular for using in practical applications such as image restoration , Super-resolution for videos (Specially for cartoon and anime movies). Real -ESGRAN was trained on purely synthetic data, with an enhance high-order degradation process to generate the training images.

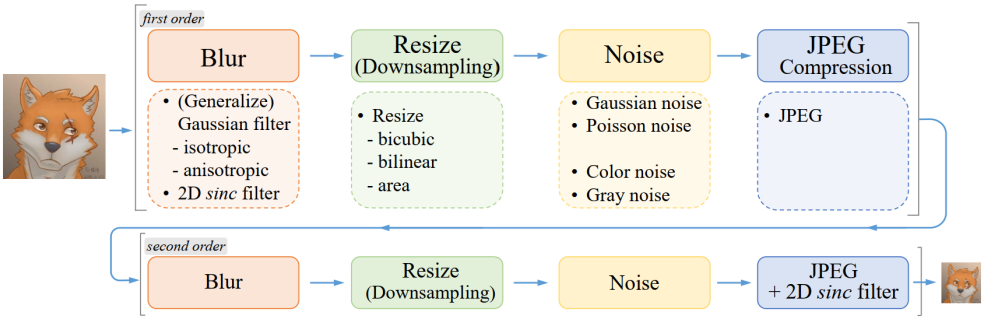


FIG. 12. Overview of the pure synthetic data generation adopted in Real-ESRGAN. It utilizes a second-order degradation process to model more practical degradations, where each degradation process adopts the classical degradation model. The detailed choices for blur, resize, noise and JPEG compression are showed in the figure.

The method models practical degradations through repeated degradation processes, making it more flexible and realistic compared to single-step degradation models. This process helps in better mimicking real-world degradation. Real-ESRGAN also incorporates of sinc filters during the degradation process to simulate common artifacts (ringing and overshoot) seen in real-world images.

For network architecture, Real-ESRGAN adopt the same generator (SR network) as ESRGAN [16], i.e., a deep network with several residual-in-residual dense blocks (RRDB), as shown in Fig. 13.

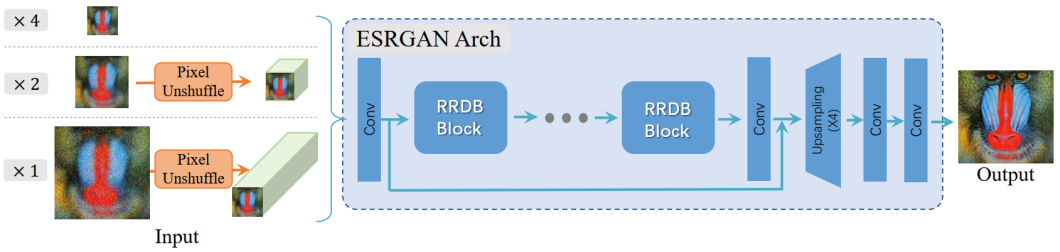


FIG. 13. Real-ESRGAN adopts the same generator network as that in ESRGAN. For the scale factor of ×2 and ×1, it first employs a pixel-unshuffle operation to reduce spatial size and re-arrange information to the channel dimension.

Real-ESRGAN is extended from the original ×4 ESRGAN architecture to perform super-resolution with a scale factor of ×2 and ×1. As ESRGAN is a heavy network, a pixel-unshuffle (an inverse operation of pixelshuffle [14]) is employed to reduce the spatial size and enlarge the channel size before feeding inputs into the main ESRGAN architecture. Thus, the most calculation is performed in a smaller resolution space, which can reduce the GPU memory and computational resources consumption.

Real-ESRGAN use an U-Net as the discriminator instead of a global styles discriminator model. The U-Net design increases the discriminator's ability to discern realness in images, provide detailed per-pixel feedback to the generator while spectral normalization helps stabilize the training dynamics, particularly important given the complex degradations being modeled. The U-Net structure and complicate degradation methods also increase the training instability.

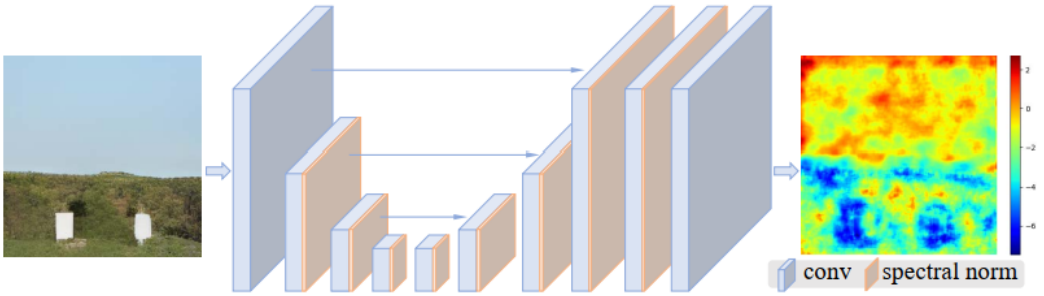


FIG. 14. Architecture of the U-Net discriminator.

The training process is divided into two stages. First, a PSNR - oriented model was trained with the L1 loss. The obtained model is named by Real-ESRNet. Then the trained PSNR-oriented model is used as an initialization of the generator, and train the Real-ESRGAN with a combination of L1 loss, perceptual loss using the first 5 convolution layers of VGG19 network and common adversarial loss for GAN training process, with weights $\{1, 1, 0.1\}$. Real-ESRNet was trained for 1000K iterations with learning rate 2×10^{-4} while Real-ESRGAN was trained for 400K iterations with learning rate 1×10^{-4}.

4. Experimental results

All 3 approaches are evaluated using Set5, Set14 and BSD100 as described in section 2.2. Using 3 metrics PSNR, SSIM and MOS, we conduct the evaluation for each approaches, record the evaluation results and save all output super-resolution images to local[3].

4.1. *Results for classical approaches*

(a) Nearest Neighbor (b) Bilinear

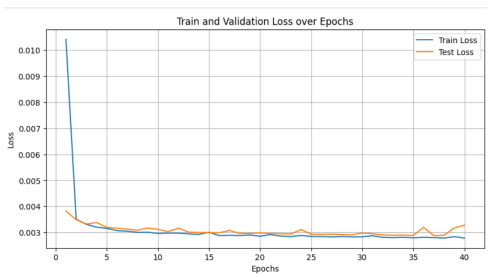
(c) Bicubic (d) GT image

FIG. 15. Result for image processing methods

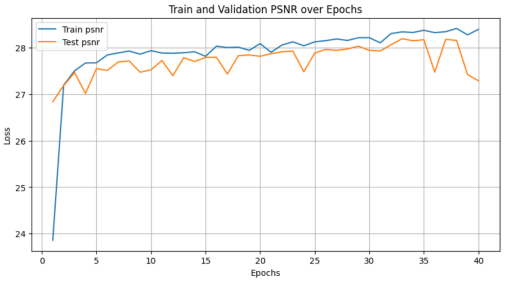
TABLE 1 *Performance of Image processing method*

	Set5		Set14		BSD100	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Nearest	24.56	0.73	22.96	0.65	21.97	0.61
Bilinear	25.82	0.76	23.70	0.66	23.55	0.65
Bicubic	26.85	0.79	24.35	0.69	23.11	0.65D

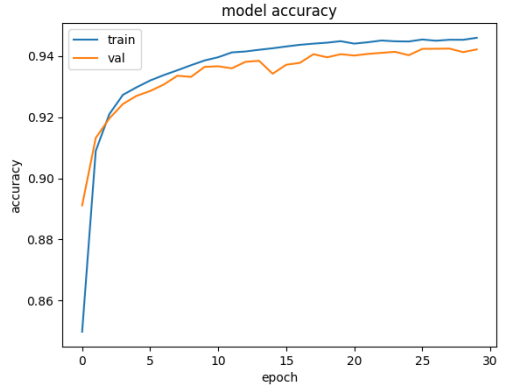
4.2. *Results for learning-based approaches*



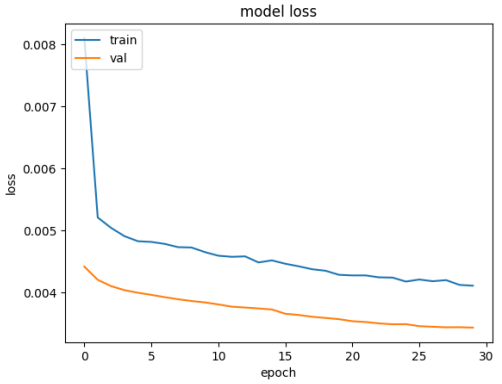
(a) SRCNN loss over 40 epochs



(b) SRCNN PSNR evaluation over 40 epochs



(c) VDSR accuracy over first 30 epochs



(d) VDSR lost over first 30 epochs

FIG. 16. SRCNN & VDSR training details

[3] See here for more info

(a) SRCNN (b) VDSR

(c) ESPCN (d) GT image

FIG. 17. Result images for different CNN models

TABLE 2 *Performance of CNN models*

	Set5		Set14		BSD100	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
SRCNN	23.87	0.79	22.48	0.69	18.94	0.57
VDSR	25.98	0.77	23.85	0.66	24.03	0.66
ESPCN	27.07	0.79	24.59	0.70	21.30	0.56

4.3. *Results for modern generative approaches*

(a) dSRVAE (b) SRGAN

(c) Real-ESRGAN (d) GT image

FIG. 18. Result images for different generative models

TABLE 3 *Performance of Generative models*

	Set5		Set14		BSD100	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
dSRVAE	24.93	0.73	23.68	0.65	21.54	0.55
SRGAN	21.35	0.61	18.50	0.44	23.20	0.61
Real-ESRGAN	24.30	0.73	23.27	0.65	21.96	0.59

4.4. *Comparison*

Out of the three mentioned methods, implementing the image processing method is not only the simplest, but also the most efficient in terms of memory and execution time. By solely relying on mathematical formulas, it calculates pixel values based on existing ones. However, it's important to note that the effectiveness of this method isn't exceptionally high, as the resulting images may exhibit jagged edges or excessive smoothness. It is ideal for situations where a low scale factor is needed or when high accuracy isn't a priority. Nevertheless, this method remains significant as it assists in preprocessing images for the CNN-based method, serving a crucial role in the overall process.

The CNN-based method yields more promising results than the basic image processing method. Images generated by CNNs have higher PSNR, SSIM, and MOS scores compared to traditional image processing methods. However, CNN-based methods require longer training times and resources for training. Training a CNN can be performed with a GPU, and the output images from the CNN are good enough for various purposes. Some methods, such as SRCNN and VDSR, have the drawback that the input images need to be upscaled using bicubic interpolation before passing through the network layers. This can increase both the complexity and the processing time.
The results from the experiments conducted on the three mentioned datasets demonstrate that GAN models consistently achieve the highest MOS scores (as shown in table 4). This indicates that GAN models excel in the task of recreating high-resolution images from low-resolution ones. While image processing techniques and CNNs may yield high peak signal-to-noise ratios, they often lack high-frequency details and fail to satisfy the human visual perception by not matching the expected fidelity at higher resolutions. Further studies have revealed that common quantitative measures such as PSNR and SSIM are inadequate in accurately assessing image quality from the perspective of the human visual system. Another reason why generative models may receive lower PSNR and SSIM scores is due to their ability to create entirely new pixels rather than relying on existing ones, rendering pixel-based evaluation less effective. The new generation of generative models has the impressive ability to handle even the most challenging scenarios, such as input images that contain noise or artifacts. This advanced capability makes generative models incredibly powerful, not only for super-resolution tasks but also for denoising and sharpening problems. However, training generative models can be a challenging and time-consuming task. It demands a significant amount of data and considerable training time. Moreover, these models can sometimes encounter instability issues during the training procedure. Moreover, generative models such as VAE or GAN often possess large network sizes and extreme complexity, leading to significantly slower performance than image processing methods or CNNs. As a result, they are not practical for deployment in real-life applications.

TABLE 4 *MOS evaluation for every method/algorithm*

	Nearest	Bilinear	Bicubic	SRCNN	VDSR
MOS	1.53	1.76	2.08	2.36	2.41
	ESPCN	dSRVAE	SRGAN	Real-ESRGAN	GT
MOS	2.45	3.23	3.69	4.03	4.82

5. Discussion

5.1. *Important findings*

After extensive analysis and experimentation, we have gathered important findings and insights regarding Single Image Super-Resolution (SISR) methods:

- Image processing methods are not only cost-effective but also easily implemented. However, it's worth noting that these methods may not yield highly realistic results.

- Models such as VAE, GAN, and diffusion have demonstrated their incredible ability to significantly enhance the visual quality of images, especially when it comes to reducing noise or correcting artifacts. However, due to their high complexity, heavyweight and extremely slow performance, most of these models are not practical for real world implementation.
- CNNs may offer the best of both worlds compared to traditional image processing methods and generative models, but they do not consistently excel in practical implementation or output accuracy.
- Enhancing the degradation algorithm used in the training data can effectively capture realistic images, thus leading to a more comprehensive and authentic model training process, result in better more nature and realistic images.
- PSNR and SSIM do not capture the complexity of human visual perception, which involves not just structural fidelity but also semantic content and subjective aesthetics. This can be proved by the evaluation results between generative models and the other methods. Also, these methods would fail if the input is affected by noise or contain artifacts.

5.2. *Recommendation*

Based on our comprehensive analysis and experimental results, we offer the following recommendations for practitioners and researchers working on Single Image Super-Resolution (SISR):

- To apply into real-time processing, consider employing image processing techniques such as Bicubic interpolation or compact Convolutional Neural Networks like SRCNN or ESPCN, which offer lower computational complexity and easy for pratical implementation.
- To achieve the highest possible image quality without being restricted by time, it's crucial to explore advanced techniques such as generative models(GAN, Diffusion model).
- By combining datasets like DIV2K with synthetic ones that include realistic noise and blur effects, we can create a powerful simulation of real-world conditions. This enhancement greatly increases the practicality and applicability of the model across various fields of application.
- Some of State-of-the-art methods like Super-Resolution using Diffusion model or Vision Transformer should be consider for study (Due to the complex nature and significant resource demands of these (SOTA) methods, we have made the decision to exclude them from our report).

6. Conclusion

In this report, we have comprehensively investigated Single Image Super-Resolution (SISR) through an in-depth exploration of traditional image processing techniques, Convolutional Neural Networks (CNNs), and Generative Adversarial Networks (GANs). Our evaluation of these methods was conducted utilizing public datasets such as Set5, Set14, and BSD100. Our findings indicate that while traditional image processing methods are straightforward, CNNs and generative models significantly enhance image resolution with higher quality outputs. Each approach has its strengths, with CNNs offering a good balance between performance and computational cost, and generative models excelling in visual quality despite their complexity. Looking ahead, future work should concentrate on developing hybrid models that combine the strengths of these approaches and enhancing the training efficiency for deep learning models. Moreover, it will be crucial to expand and diversify datasets to further advance SISR capabilities in real-world applications.

REFERENCES

1. Eirikur Agustsson and Radu Timofte. Ntire 2017 challenge on single image super-resolution: Dataset and study. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 126–135, 2017.
2. Marco Bevilacqua, Aline Roumy, Christine M. Guillemot, and Marie-Line Alberi-Morel. Low-complexity single-image super-resolution based on nonnegative neighbor embedding. In *British Machine Vision Conference*, 2012.
3. Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Image super-resolution using deep convolutional networks, 2015.
4. Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks, 2014.
5. Bilal Himite. Image processing: Image scaling algorithms, 2021. Accessed: 2024-05-29.
6. Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. Accurate image super-resolution using very deep convolutional networks, 2016.
7. Diederik P. Kingma and Max Welling. An introduction to variational autoencoders. *Foundations and Trends® in Machine Learning*, 12(4):307–392, 2019.
8. Diederik P Kingma and Max Welling. Auto-encoding variational bayes, 2022.
9. Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, and Wenzhe Shi. Photo-realistic single image super-resolution using a generative adversarial network, 2017.
10. Juncheng Li, Zehua Pei, Wenjie Li, Guangwei Gao, Longguang Wang, Yingqian Wang, and Tieyong Zeng. A systematic survey of deep learning-based single-image super-resolution, 2024.
11. Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced deep residual networks for single image super-resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 136–144, 2017.
12. Zhi-Song Liu, Wan-Chi Siu, Li-Wen Wang, Chu-Tak Li, Marie-Paule Cani, and Yui-Lam Chan. Unsupervised real image super-resolution via generative variational autoencoder, 2020.
13. D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proc. 8th Int'l Conf. Computer Vision*, volume 2, pages 416–423, July 2001.
14. 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, 2016.
15. Xintao Wang, Liangbin Xie, Chao Dong, and Ying Shan. Real-esrgan: Training real-world blind super-resolution with pure synthetic data, 2021.
16. Xintao Wang, Ke Yu, Shixiang Wu, Jinjin Gu, Yihao Liu, Chao Dong, Chen Change Loy, Yu Qiao, and Xiaoou Tang. Esrgan: Enhanced super-resolution generative adversarial networks, 2018.
17. Zhihao Wang, Jian Chen, and Steven C. H. Hoi. Deep learning for image super-resolution: A survey, 2020.
18. Roman Zeyde, Michael Elad, and Matan Protter. On single image scale-up using sparse-representations. volume 6920, pages 711–730, 06 2010.