Super Resolution using Various Interpolation Techniques

Priyansh Agarwal (22116073), Yusuf Topiwala (22116111), Shashank Bhardwaj (22116099), Aman (21116013)
Department of Electronics & Communication Engineering,
Indian Institute of Technology (IIT) Roorkee.
Supervised by: Vinod Pankajakshan (VP), Associate Professor, IIT Roorkee.

I. Introduction-What is Interpolation

Interpolation is a method for adding new data points within a range of a set of known data points. Interpolation can be used to fill in gaps in data, smooth out data, make predictions .The process of interpolation involves figuring out the unknown values that lie between the known data points. For example in case of geographically related data points, parameters such as noise level, rainfall, elevation, and so forth, it is primarily used to forecast unknown values.

II. Different Types of Interpolation

We will be implementing the following algorithms:

- Nearest Neighbor Interpolation
- bilinear Interpolation
- Bi Cubic Interpolation
- Spline Interpolation
- Lancoz Resampling
- Edge Directed Interpolation
- K-Nearest Neighbor
- Wavelet Transform Interpolation

III. Nearest Neighbor Interpolation *by Aman*

Nearest neighbor interpolation is a technique used to estimate the value of a function at a new, unknown point by looking at the values of the function at surrounding, known points. It's essentially a way of filling in the gaps between existing data points .The nearest neighbor method finds the existing data point that's closest to your new point and is usually determined by calculating the distance between them. The value of the function at the new point is simply assigned the same value as its nearest neighbor.

A. Advantage

Nearest neighbor interpolation is very easy to understand and implement. It's computationally cheap, making it ideal for real-time applications.

B. Disadvantage

The blocky appearance creates a staircase effect, especially for diagonal lines or smooth curves. As it only considers the closest neighbor, it ignores potentially valuable data from other nearby points so there is loss of information.

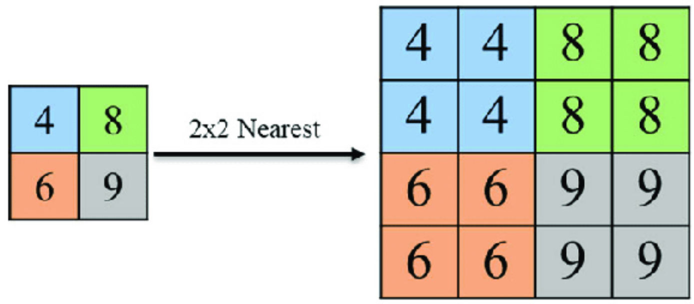

Fig. 1. Nearest neighbor interpolation

IV. Bilinear Interpolation *by Priyansh*

Bilinear interpolation is an extension of linear interpolation to a two-dimensional space. It is a method used to estimate the value of a function at a point within a grid of known values. The key idea behind bilinear interpolation is to interpolate both horizontally and vertically between the values of the pixels of an image.

A. Advantage

Bilinear interpolation is a simple and fast method for upscaling images, making it a popular choice in applications where computational efficiency is important.

B. Disadvantage

It assumes that the changes in the function across the grid are linear, leading to reasonably good results for smooth functions, but it may not be as accurate for complex or rapidly changing functions.

The bilinear interpolation formula combines two linear interpolations to compute an estimate in a 2D space.

- $f(x_1, y_1) = Q_{11}$
- $f(x_2, y_1) = Q_{21}$
- $f(x_1, y_2) = Q_{12}$
- $f(x_2, y_2) = Q_{22}$

The bilinear interpolation formula is:

$$f(x,y) = \frac{(x_2 - x)(y_2 - y)}{(x_2 - x_1)(y_2 - y_1)} Q_{11} + \frac{(x - x_1)(y_2 - y)}{(x_2 - x_1)(y_2 - y_1)} Q_{21} + \frac{(x_2 - x)(y - y_1)}{(x_2 - x_1)(y_2 - y_1)} Q_{12} + \frac{(x - x_1)(y - y_1)}{(x_2 - x_1)(y_2 - y_1)} Q_{22}$$

This formula involves a weighted average of the four surrounding points, ensuring that the closer a point is to the target, the more influence it has on the interpolated result.

V. Bicubic Interpolation *by Yusuf*

Bicubic takes in to consideration the intensity values of more neighboring pixels to create a smoother curve by using the additional information to create non linear functions which reduce the occurrence of shortcomings produced by the bilinear method as stated earlier. The additional data allows for a more precise estimation of pixel values, leading to higher-quality and better-detailed images.

A. Advantage

Produces smoother and more visually pleasing results compared to nearest-neighbor or bilinear interpolation.

B. Disadvantage

More complex than nearest-neighbor and bilinear interpolation due to its reliance on cubic polynomials, making it slower.It causes ringing artifacts around sharp edges, particularly in high-contrast images.

For a function $f(x,y)$, the interpolated value at (x,y) is given by:

$$f(x,y)=\sum_{i=-1}^{2}\sum_{j=-1}^{2} w(i,j)\cdot f(x_i,y_j)$$

Where:

- $w(i,j)$: Weighting coefficient based on the cubic kernel.
- $f(x_i,y_j)$: Known pixel values of the surrounding grid.
- (x,y): Coordinates of the point being interpolated.

The summation is over the 4x4 grid of surrounding pixels.

VI. Spline Interpolation *by Shashank*

Spline interpolation is a form of interpolation where the interpolant is a special type of piecewise polynomial called a spline. That is, instead of fitting a single, high-degree polynomial to all of the values at once, spline interpolation fits low-degree polynomials to small subsets of the values. A spline is a smooth, flexible curve created by joining these polynomial segments at the knots, which are simply the points where two segments meet.

A. Advantage

By using low-degree polynomials for each segment, spline interpolation avoids the oscillatory behavior observed in high-degree polynomial interpolation.

B. Disadvantage

Spline interpolation requires solving a system of equations to determine the coefficients of the polynomials, which can be computationally intensive for very large datasets.If the data points are very sparse or unevenly spaced, the spline may not accurately capture the underlying function.

The spline interpolation formula is:

$$S(x)=\sum_{i=0}^{n} c_i\cdot N_{i,k}(x)$$

where c_i are the control points and $N_{i,k}(x)$ are the B-spline basis functions of degree k.

VII. Lanczos Interpolation *Coding by Aman, Theory by Priyansh*

Lanczos interpolation uses the Lanczos kernel to calculate the new pixel values during resampling. It applies the concept of convolution, where the value of the new pixel is determined by the weighted average of neighboring pixels, with the weights being provided by the Lanczos kernel. The Lanczos kernel is derived from the sinc function and is truncated with a windowing function to make it suitable for discrete data.

The Lanczos kernel is defined as:

$$\text{Lanczos}(x)=\frac{\sin(\pi x)}{\pi x}\cdot\frac{\sin\left(\frac{\pi x}{a}\right)}{\frac{\pi x}{a}}$$

where a is a parameter (usually 3 or 4) that determines the number of neighboring points considered for interpolation.

A. Advantage

Lanczos interpolation typically produces higher-quality results than simpler methods like bilinear or bicubic interpolation. The Lanczos method uses a windowed sinc function, which helps reduce aliasing artifacts in the image

B. Disadvantage

Lanczos interpolation requires more computational power compared to simpler methods For smaller images or when the rescaling factor is small, Lanczos interpolation may be overkill and lead to unnecessary complexity.

VIII. Edge Directed Interpolation *Coding by Yusuf, Theory by Priyansh*

This technique focuses on adjusting the interpolation process based on the detected edges, modifying the interpolation kernel dynamically to give more weight to pixels near edges.

A. Advantage

The primary advantage of edge-directed interpolation is its ability to preserve sharp edges and high-frequency details. Traditional interpolation techniques can cause noticeable blurring along edges.

B. Disadvantage

The quality of edge-directed interpolation depends heavily on accurate edge detection. If the edge detection algorithm fails to correctly identify edges, the interpolation may not preserve sharp edges as effectively.

For an image $f(x, y)$, where x and y are pixel coordinates, and $Q_{i,j}$ are the known pixel values in the neighborhood of the point being interpolated, the formula can be represented as:

$$f(x, y) = \sum_{i,j \in N} w_{i,j}(x, y) \cdot f(x_i, y_j)$$

Where:

- $f(x, y)$ is the interpolated pixel value at the position (x, y),
- N is the neighborhood of the pixel (e.g., 2x2 or 3x3 neighborhood),
- $f(x_i, y_j)$ are the known pixel values around the interpolation point,
- $w_{i,j}(x, y)$ are the interpolation weights, which are edge-directed weights determined by the edge information around the point.

IX. K-Nearest Neighbor *by Shashank*

K-Nearest Neighbors (KNN) can be used for image super-resolution by matching low-resolution image patches with similar high-resolution patches from a training dataset. The idea is to find the K nearest neighbors of a given low-resolution patch and use the corresponding high-resolution patches to improve the image quality.

A. Advantage

No need for explicit image models and it works well for various image types.

B. Disadvantage

Requires comparing every patch with the training set and has high computation complexity. Also requires specific training for each specified task, super-resolution for X-Rays requires a dedicated model and cannot be used for Cars.

X. Wavelet Tranform Interpolation *Coding by Shashank, Theory and implementation design by Yusuf and Priyansh*

Wavelet transform interpolation is a technique that uses wavelet transformations to enhance the resolution of an image during interpolation. Wavelet-based super-resolution is done using the Haar wavelet to upscale an image. It leverages the ability of wavelet transforms to capture both low-frequency and high-frequency components of the data.

A. Advantage

By focusing on high-frequency subbands, wavelet-based interpolation is good at preserving edges and sharp details in the image. It provides a multiresolution analysis, which helps capture both coarse and fine details

B. Disadvantage

When applied to finite-length images, wavelet transforms can produce boundary artifacts at the edges due to the limited overlap of wavelet basis functions. High-frequency subbands amplify noise along with signals.

XI. Evaluation Metrics *Researhced by Aman, implemented by Shashank*

Comparison of PSNR, SSIM, and LPIPS

1. PSNR (Peak Signal-to-Noise Ratio)

PSNR measures the ratio of the maximum possible power of a signal (image) to the power of noise (difference between two images). It is commonly expressed in decibels (dB) and used for evaluating image quality after compression or restoration.

Working:

1) Compute the Mean Squared Error (MSE) between two images:

$$\text{MSE} = \frac{1}{N} \sum_{i=1}^{N} \left(I_1(i) - I_2(i)\right)^2$$

2) Calculate PSNR using the formula:

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

where MAX is the maximum possible pixel value (e.g., 255 for 8-bit images).

Strengths:

- Simple and fast to compute.
- Works well for small, uniform distortions.

Limitations:

- Does not align well with human visual perception.
- Sensitive to pixel-by-pixel differences, even if they are visually imperceptible.

2. SSIM (Structural Similarity Index Measure)

SSIM measures the structural similarity between two images by comparing luminance, contrast, and structure.

Working:

1) Divide the image into small windows (e.g., 7×7 pixels).
2) For each window, compute:
   - Luminance similarity: Measures brightness similarity.
   - Contrast similarity: Measures intensity variation.
   - Structural similarity: Compares local patterns.
3) Combine these measures into the SSIM formula:

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

where:

- μ_x, μ_y: Mean intensities.
- σ_x, σ_y: Variances.
- σ_{xy}: Covariance.
- C_1, C_2: Small constants for numerical stability.

Strengths:

- Captures perceptual features like structure and contrast.
- Better alignment with human perception than PSNR.

Limitations:

- Sensitive to alignment and small distortions.
- Struggles with highly complex visual differences.

3. LPIPS (Learned Perceptual Image Patch Similarity)

LPIPS is a deep learning-based metric that measures perceptual similarity between two images using features from a neural network trained on human similarity judgments.

Working:

1) Pass the two images through a pre-trained neural network (e.g., VGG, AlexNet).
2) Extract intermediate feature maps from multiple layers.
3) Compute the distance between feature maps using an appropriate distance metric (e.g., L2 norm).
4) Aggregate the distances to produce the LPIPS score.

Strengths:

- Highly aligned with human perception.
- Captures complex visual differences, such as textures and semantics.

Limitations:

- Computationally expensive.
- Dependent on the pre-trained network and its architecture.

Summary of Use Cases

Metric	Use Case	Strengths	Weaknesses
PSNR	Simple quality measurement	Fast, easy to compute	Poor perceptual alignment
SSIM	Quality with perceptual focus	Captures structure and contrast	Sensitive to small distortions
LPIPS	High perceptual fidelity	Aligns well with human vision	Computationally intensive

XII. Conclusion

In this report, we explored multiple interpolation techniques to enhance image resolution, each with distinct strengths and limitations.

Simpler methods like K Nearest Neighbor (KNN) and Bilinear Interpolation offer computational efficiency but often compromise on quality. Advanced techniques such as Bicubic Interpolation and Lanczos Resampling achieve smoother results and better detail preservation, making them suitable for applications requiring higher visual fidelity. Meanwhile, Edge-Directed Interpolation effectively retains edges and high-frequency details, Wavelet Transform Interpolation introduces a unique approach leveraging frequency-domain information, further enhancing super-resolution.

Through quantitative metrics like PSNR and SSIM, this study underscores the importance of choosing the appropriate interpolation method based on application requirements.

Future work could explore hybrid techniques combining the strengths of these methods for even better results.

XIII. Extra Work: DL experiments by Yusuf and Shashank

We have also tried various architectures of DL models to see performance. However every model we tried stagnates at pixel-wise accuracy of 0.9627. Regardless of model architecture, the learning capacity of the model is less when considering super-resolution from 128x128 to 256x256. But we also could not try implementing deeper models for lack of access to GPU.

Various models were attempted, CNNs, U-Net inspired networks, and ResNet inspired Networks, but due to lack of GPU, deeper models were not succesfully attempted, and none of the experiments were fruitful.

References

[1] Gonzalez, R. C., Woods, R. E. (2008). Digital Image Processing. Prentice Hall.

[2] Cubic Convolution Interpolation for Digital Image Processing. IEEE Transactions on Acoustics, Speech, and Signal Processing, 29(6), 1153–1160.

[3] Mallat, S. (2009). A Wavelet Tour of Signal Processing. Academic Press.

[4] ResearchGate, "The Nearest Neighbor Interpolation for Up-sampling," Available at: https://www.researchgate.net/figure/The-nearest-neighbor-interpolation-for-Up-sampling_fig5_358610282

[5] K. Xie and L. Lu, "Lanczos algorithm based L-curve and curvature in image resolution enhancement," in *2009 9th International Conference on Electronic Measurement & Instruments*, Beijing, China, 2009, pp. 4-373–4-377, doi: 10.1109/ICEMI.2009.5274067.