EP 4130 : Data Science Analysis
Enhancing Astronomical Image Clarity: Superresolution Techniques and Satellite Mitigation

Sai Harish Madireddi
EE21BTECH11047
ee21btech11047@iith.ac.in

Abstract

In this project, we propose a method for improving the quality of astronomical images, which are often degraded, and satellite tracks that obscure critical information. We address this issue by firstly removing satellite tracks by employing Hough Transform. We then reconstruct high-resolution images using a CNN-based super-resolution reconstruction technique, which involves an SRCNN (Super Resolution Convolutional Neural Network) process to enhance the accuracy and visual quality of the reconstructed image. We finally, evaluate the effectiveness of our approach using various metrics such as PSNR, MSE and SSIM. Our work is motivated by the need to extract more information from low-resolution images obtained by space telescopes and the increasing incidence of satellite tracks due to private satellites. Our results demonstrate the effectiveness of our approach in producing high-quality astronomical images with improved contrast and better visibility of hidden features.

1. Introduction

The space program of the 1950s provided the first high-resolution images of Earth, Moon, and Mars, but technical issues degraded their quality. Since then, digital image restoration has emerged as a powerful tool for extracting more information from astronomical observations. Recent advancements in machine learning and deep learning have further expanded the image processing toolkit for researchers to explore and understand celestial objects in greater detail. Image processing is now an essential preprocessing and analytical phase in the astronomical observation pipeline.

The objective of this research paper is to present a method for constructing high-resolution astronomical images from low-resolution, blurred observations and removing satellite trails that obscure critical information.

This manuscript is structured as follows. The motivation behind the work is described in Sect. 2. The analysis methodology is described in Sect. 3. The results are in Sect. 4. The progress after the previous report is in Sect. 5. Finally the conclusions are in Sect. 6.

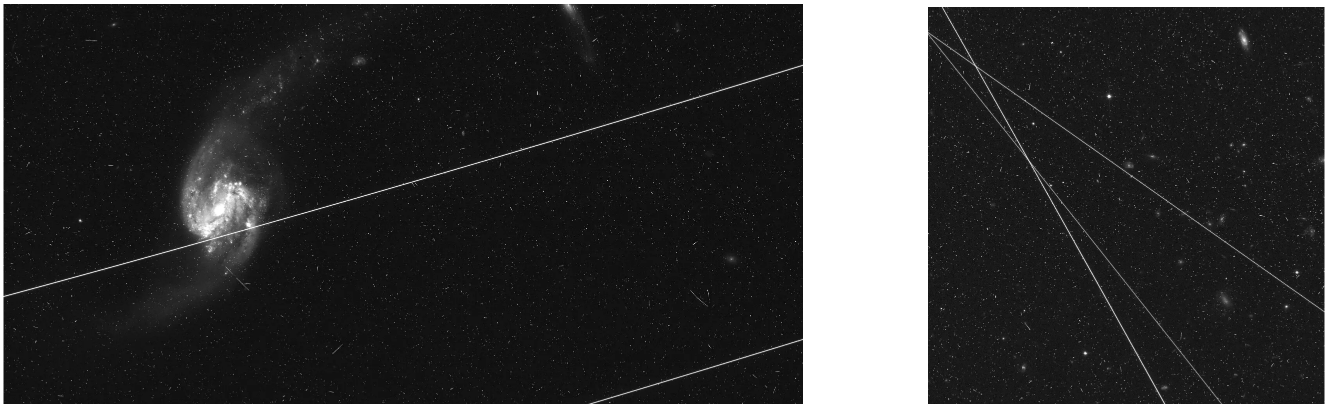
2. Motivation

A study [3] revealed that the number of Hubble images distorted by passing satellites has increased due to private satellites such as Starlink, producing unremovable long, bright streaks and curves of light. The likelihood of spotting a satellite in a Hubble image between 2009 and 2020 was only 3.7%, but it increased to 5.9% in 2021, an increase that experts claim is related to Starlink.

The first objective is thus, to use Hough Transform [4] to identify and remove satellite traces in images. The results obtained from these two methods are then compared after enhancing images to highlight hidden features.

Space telescopes are limited in their physical resolution by physical phenomena such as the atmosphere and the interstellar medium, as well as system features such as lens aperture and CCD array qualities. These systems gather images of celestial entities, often distorted by mechanical vibrations and the satellite's movement.

We then use Superresolution image reconstruction [2] to create a new, high-resolution image from the low-resolution image and noisy observation to identify image details and structures not readily discernible in the raw data.



(a) A satellite track cutting an image of a galaxy.

(b) Three satellite trails in one Hubble image.

Figure 1. Examples of satellite trails observed. NASA, ESA, Kruk et al.

3. Methodology

3.1. Hough Transform

The Hough Transform is an image processing technique that can detect linear and other features in an image. It works by transforming the data from the point space to the Hough space, which represents lines [4]. In the Hough space, each point corresponds to a line in the original space that is perpendicular to the line passing through the center of the data space and inclined at a specific angle.

To apply the Hough Transform, the following steps are taken:

1. Compute discrete values for r_i and θ_i in the appropriate range for the image. The range of r is from -R to R (where R is the resultant distance), and the range of θ is from -$\pi/2$ to $\pi/2$.

2. Create an array of size r_n by θ_n, where r_n is the number of r values and θ_n is the number of θ values. This array is an accumulator or a three-dimensional histogram.

3. For each pixel in the image, loop through each θ value. Compute r for each θ value and select the nearest discrete r value.

4. Increment by one the appropriate [r,θ] element in the accumulator. The more counts an element has in the accumulator, the more points there are on that particular line.

5. Once all pixels have been processed, the accumulator will contain peaks at the [r,θ] values corresponding to the lines in the image.

6. If the number of counts in a particular element exceeds a certain value, then the [r,θ] value associated with that element is considered to be a line in the image.

The Hough Transform is particularly useful when traditional edge detection techniques fail, such as when the edges are not well-defined or broken. It can also be used to detect other features in an image, such as circles or ellipses, by modifying the accumulator to represent the shapes [4].

For the removal of the line detected using the algorithm, we use inpaint algorithm, which is a computer vision technique which fills in the missing pixels by using information from the surrounding pixels, this is done by solving a partial differential equation known as the Navier-Stokes equation, which is used to estimate the pixel values in the missing regions by taking into account the values of neighbouring pixels, as well as the structure and texture of the surrounding regions.

The implementation for the Hough Transform can be found in this repository.

3.2. Image Super-Resolution

Super-resolution refers to the task of increasing the spatial resolution of an image, typically by upsampling a low-resolution image to a higher resolution.

3.2.1 Wavelet Based Super-resolution (not implemented)

We had initially planned to create super-resolution image using Wavelet-Based Super Resolution [5], which involves creating a higher-resolution image from a set of low-resolution, rotated, shifted, and noisy observations. Wavelets and multiresolution analysis are particularly suitable for this task since they can provide accurate and sparse images of smooth regions with abrupt changes or singularities. This method utilizes the EM algorithm to address image deconvolution by combining the discrete wavelet transform's efficient image representation with the diagonalization of the convolution operator obtained in the

Fourier domain. The algorithm alternates between an E-step based on the Fast Fourier Transform (FFT) and a DWT(Discrete wavelet transform)-based M-step, which requires O(NlogN) operations per iteration, where N is the number of pixels in the super-resolution image.

Due to unforeseen circumstances and hard implementations, we switched to generating the super-resolution image using SRCNN.

3.2.2 Image Super-Resolution Using Deep Convolutional Networks

The proposed method uses deep learning and convolutional neural networks (CNNs) to perform image super-resolution [2]. The approach involves training a CNN to learn the mapping from low-resolution images to high-resolution images.

We use a deep CNN architecture consisting of multiple convolutional layers, activation functions, and normalization layers. The network takes a low-resolution image as input and outputs a high-resolution image. The paper experiments with different network architectures, including different numbers of layers, filter sizes, and activations.

The network is trained using a large dataset of high-resolution images and their corresponding low-resolution versions. We use a mean squared error (MSE) loss function as the objective function to train their CNN. The MSE loss measures the difference between the predicted high-resolution image and the ground truth high-resolution image. However, the authors also use perceptual loss functions to improve image quality. Perceptual loss functions measure the difference between high-resolution images in a feature space, rather than in pixel space, which allows the network to learn high-level features and produce more visually appealing results.

The training process involves iterating over the dataset multiple times, with each iteration updating the parameters of the CNN using gradient descent. We use the Adam optimizer to optimize the objective function. We also employ a technique called batch normalization, which normalizes the inputs to each layer of the CNN to improve training stability and convergence.

The paper experiments with different training strategies, including pre-training on a larger dataset, using data augmentation techniques, and using different optimization techniques. They also investigate the effect of adding adversarial loss functions to the objective function. Adversarial loss functions encourage the network to produce results that are not only perceptually similar to the ground truth high-resolution images but also realistic and plausible.

We tried to train the neural model, which was time consuming to get the dataset and train the model. Thus, to save us the time it takes to train a deep neural network, we will be loading a pre-trained SRCNN model which can be found here.

The paper demonstrates the effectiveness of deep learning methods for image super-resolution. The proposed method provides a useful benchmark for future research in this area. The method has many potential applications, such as in medical imaging, satellite imaging, and video processing.

3.3. Performance Analysis

The performance of the different approaches are measured using PSNR, MSE and SSIM.

3.3.1 MSE

MSE is the most common estimator of image quality measurement metric. It is a full reference metric and the values closer to zero are the better.

It is the second moment of the error. The variance of the estimator and its bias are both incorporated with mean squared error. The MSE is the variance of the estimator in case of unbiased estimator. It is a function of risk, considering the expected value of the squared error loss or quadratic loss.

3.3.2 PSNR

PSNR is used to calculate the ratio between the maximum possible signal power and the power of the distorting noise which affects the quality of its representation. This ratio between two images is computed in decibel form. The PSNR is usually calculated as the logarithm term of decibel scale because of the signals having a very wide dynamic range. This dynamic range varies between the largest and the smallest possible values which are changeable by their quality.

The Peak signal-to-noise ratio is the most commonly used quality assessment technique to measure the quality of reconstruction of lossy image compression codecs. The signal is considered as the original data and the noise is the error yielded by the compression or distortion. The PSNR is the approximate estimation to human perception of reconstruction quality compared to the compression codecs.

3.3.3 SSIM

The Structural Similarity Index (SSIM) is a perceptual metric that quantifies image quality degradation* caused by processing such as data compression or by losses in data transmission. It is a full reference metric that requires two images from the same image capture— a reference image and a processed image. Unlike PSNR (Peak Signal-to-Noise Ratio), SSIM is based on visible structures in the image.

For our analysis, we will be using the inbuilt function from the module `skimage.metrics`.

4. Results

For the analysis, we selected an image from the NASA archive and resized it to 256 x 256. To check the robustness of all the algorithms, we tried to mimic a real-life astronomical image by adding a satellite trail.The code can be found here.

4.0.1 Hough Transform

Hough transform is a technique which is used to detect specific shapes in an image. Here we used Hough transform to detect satellite tracks (obscuring straight lines) in an image. We first read the image into a numpy array, then we applied Gaussian blur to remove the unwanted noise using `cv2.GaussianBlur` function. Hough transform takes an edge map of the image to perform its operation, hence we applied `cv2.Canny` to get a Canny image map of our image. Now, this canny edge map is given as input to our hough transform function, which returns the accumulator, the algorithmic working of this function is given in 3.2.2, the written code for this function is developed upon from here.

Now to determine the strongest line, we have used `np.argmax` function, this is used to retrieve the (r,θ) pair, which has the highest value in the accumulator. Now to show the detected line, we find both the (x,y) coordinates on either side of the image and connect both using `cv2.line` to highlight the detected line. Now to remove the line, we have used `cv2.inpaint` to replace the detected line pixels with nearby pixels information. Then, we have shown the SSIM map image between our actual image and the image with detected line removed. Also, the SSIM index turns out to be 0.958 for 128x128 image. The code for Hough Transform can be found here.

4.0.2 Image Super-Resolution Using Deep Convolutional Networks

The proposed method achieves state-of-the-art performance in terms of both objective and subjective quality measures. In the below figure-4 we can see that SSIM value has improved from 0.66 to 0.69 when the blurred image has undergone Hough Transform and the super resoluted.

5. Further Modifications Of The Previous Report

We have scraped the Wavelet-based super-resolution [5] and implemented the CNN-based method instead. We have scraped the RAST algorithm [1] as the Hough transform based method was working as expected.

6. Conclusion

Astronomical images are crucial for understanding the universe, but they often are severally blurred and contain noise and obscuring lines called satellite tracks, which can interfere with identifying important objects. To address this, we first use Hough Transform [4] to remove the satellite trails, which is used to detect linear and other features in an image by transforming the data from point space to Hough space. It involves creating an accumulator, a three-dimensional histogram, and computing r and θ values in the appropriate range. This method is particularly useful in cases where traditional edge detection techniques fail and can be adapted to detect other features like circles or ellipses and then re-construct High-resolution images using super-resolution convolutional network [2] and Hough Transform

We then use a novel approach to show the effectiveness of deep learning methods such as SRCNN for image super-resolution. The proposed method provides a useful benchmark for future research in this area.

We then will finally compare the images obtained using MSE, PSNR and SSIM to evaluate the methods used.

References

[1] Haider Ali, Christoph H. Lampert, and Thomas M. Breuel. Satellite tracks removal in astronomical images. In José Francisco Martínez-Trinidad, Jesús Ariel Carrasco Ochoa, and Josef Kittler, editors, *Progress in Pattern Recognition, Image Analysis and Applications*, pages 892–901, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg. 4

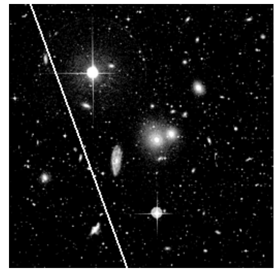
[2] Kaiming He Xiaoou Tang Chao Dong, Chen Change Loy. Image super-resolution using deep convolutional networks. 1, 3, 4

[3] Kruk S. et al. The impact of satellite trails on hubble space telescope observations. *Nat Astron (2023)*. 1

[4] Owen D Giersch and John A Kennewell. Automated analysis of satellite trails in astronomical images. 1, 2, 4

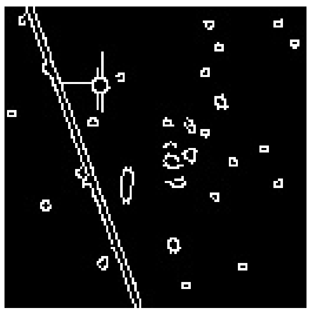
[5] R. M. Willett, I. Jermyn, R. D. Nowak, and J. Zerubia. Wavelet-Based Superresolution in Astronomy. In Francois Ochsenbein, Mark G. Allen, and Daniel Egret, editors, *Astronomical Data Analysis Software and Systems (ADASS) XIII*, volume 314 of *Astronomical Society of the Pacific Conference Series*, page 107, july 2004. 2, 4

(a) Original Image

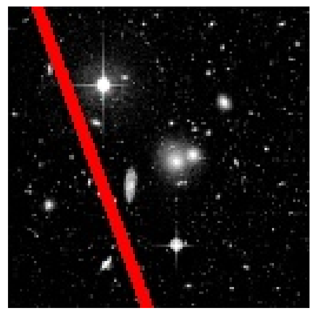


(b) Artificial image with satellite trail

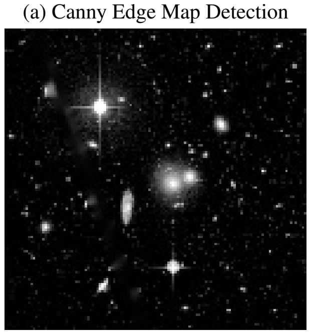
Figure 2. Generated Images



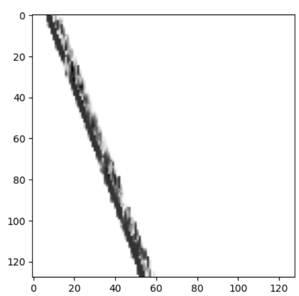
(a) Canny Edge Map Detection



(b) Line Detection



(c) After removing the detected line



(d) SSIM Map for comparision

Figure 3. Hough Transform Results.

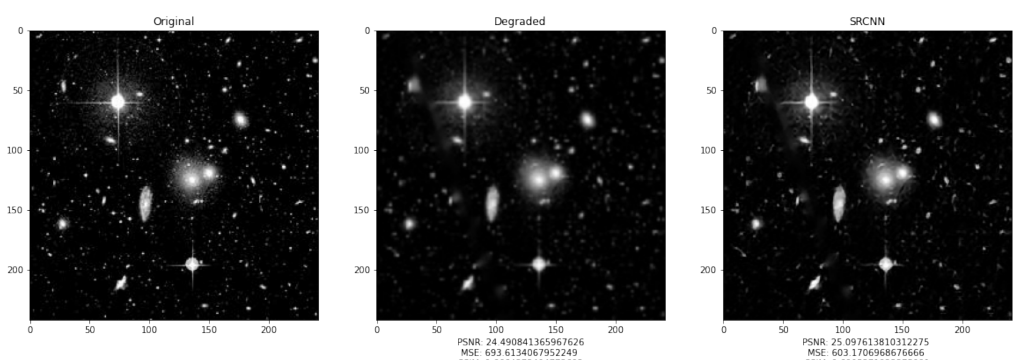


Figure 4. Generated Images