491 Project: Sparse Representation for Color Image Restoration

Tabitha Weinbrenner
Jonah Wehner

Research Paper Summary

Sparse Representation for Color Image Restoration

Topics Covered Within Research Paper

- Denoising Grayscale
- Denoising RGB
- Inpainting
- Demosaicing

Methods/Algorithms From Paper

- K-SVD
- OMP
- Dictionary Learning/Sparse Coding

Python Packages We Used

- Scipy
- Scikit-learn
- Numpy
- CV2
- Sklearn
- PIL
- Time
- Matplotlib

Our Implementation of Research Paper:

- We used methods, such as dictionary learning, to produce similar results from the paper for Denoising Grayscale, Denoising RGB, and Inpainting, but not their exact algorithms.
- Note: We did not get to Part D, Demosaicing

Methods

Part A

Denoising Grayscale

- Dictionary Learning/Sparse Coding
- Data Transformation method Orthogonal Matching Pursuit

Part B

Denoising RGB

- CV2, non-local means algorithm.
- SVD

Part C

Inpainting

- CV2, Telea's Fast Marching Method
- CV2, Navier-Stokes Method

Denoising Grayscale Image Results

Steps:

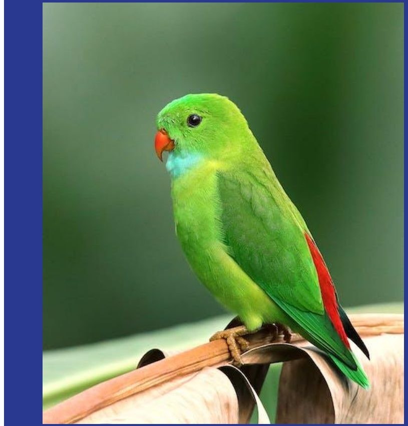
1. Image was uploaded, converted to 2D array, distortion applied to right side.

2. Using sklearn method, extract_patches_2d, patches of the distorted image were extracted.
   (Reshape a 2D image into a collection of patches. The resulting patches are allocated in a dedicated array)

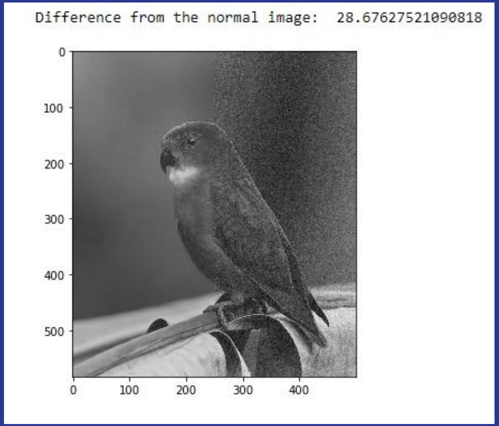
3. Dictionary was generated using sklearn, MiniBatchDictionaryLearning, then the referenced patch data was fit to the model.
   (Finds a dictionary (a set of atoms) that performs well at sparsely encoding the fitted data)

4. Noisy patches were then extracted and reconstructed using the sparse representation from the dictionary, using the transform algorithm Orthogonal Matching Pursuit.
   (OMP is a greedy algorithm that includes at each step the atom most highly correlated with the current residual)

Original Image

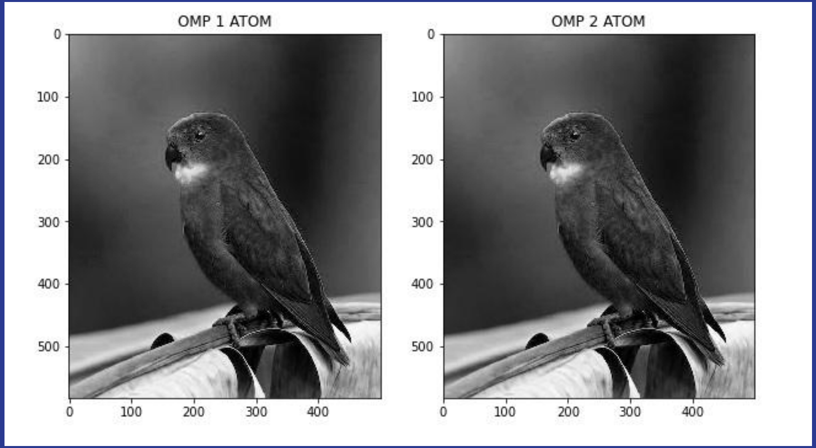


Grayscale/Distorted Image



```
Difference from the normal image:  28.67627521090818
```

Denoising Results

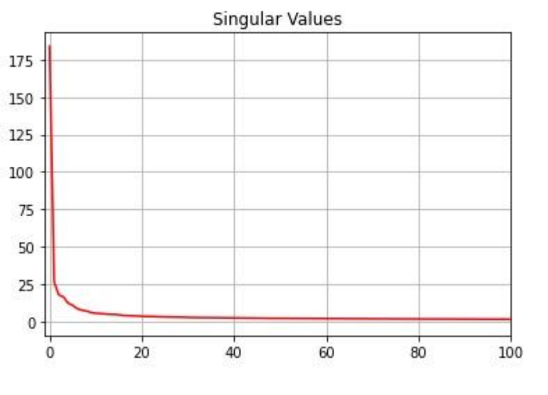


Denoising RGB Results

Steps:

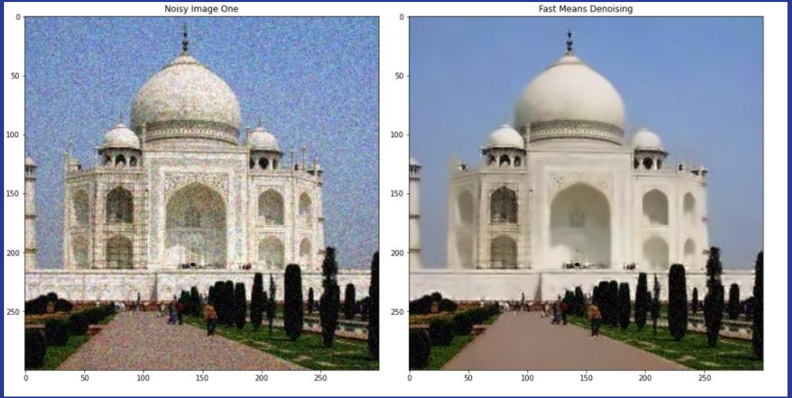
1. Image with distortion was uploaded, converted to 2D array.
2. Prior to denoising, information was derived about the image, such as:
   - *The RGB channels of image were extracted and plotted.*
   - *SVD was calculated to plot Singular Value decay.*
   - *Condition Number was derived: 38760.51847133644*
3. Finally the image was denoised using CV2 function, cv2.fastNlMeansDenoisingColored, which implements the Non-Local Means algorithm.
   Non Local Means: non-local means filtering takes a mean of all pixels in the image, weighted by how similar these pixels are to the target pixel.

Singular Value Decay via SVD



Noisy Image

Denoised Image

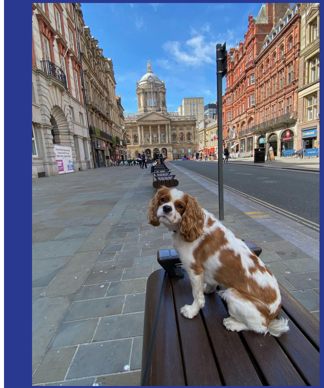


Inpainting Results

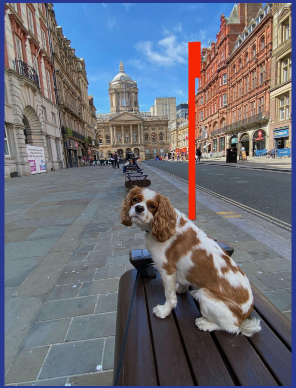
Steps:

1. Load damaged image and mask for the area to be restored into the notebook
2. For the damaged image to retrieve color we load with integer value 1 (for cv2.imread_color), and for the mask we want the grayscale image we set integer to 0 (for xv2.imread_grayscale)
3. Generate “restored” images using OpenCV inpaint function.
   a. (1) uses the fast marching method proposed by Alexandru Telea
   b. (2) uses the Navier-Stokes method

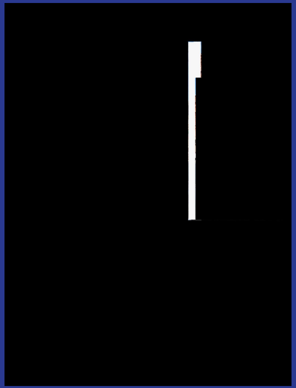
Original Image



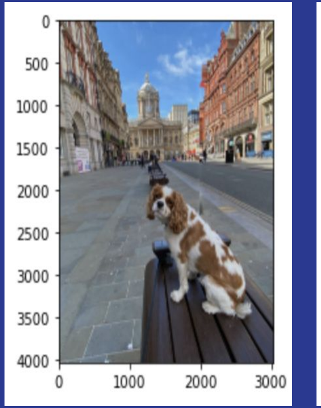
Damaged Image



Mask

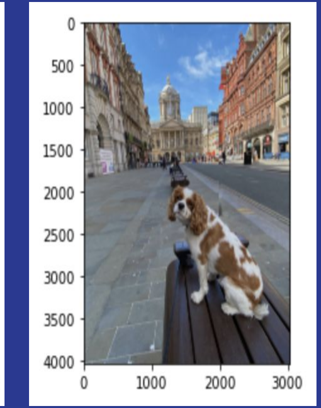


TELEA



(1)

NS

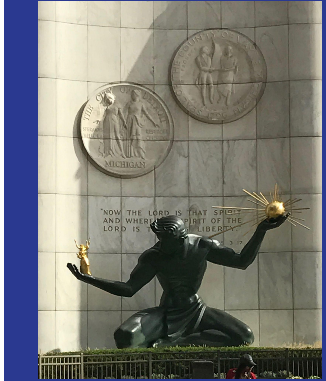


(2)

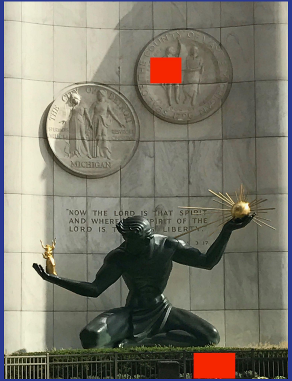
Part C: Continued

- TELEA
  - Fast Marching Method (FMM)
  - To estimate missing pixels, takes a normalized weighted sum of pixels from a neighborhood of the pixels.
  - To estimate the color of the pixels, the gradients of the neighborhood pixels are used.
- Navier-Stokes (NS)
  - Incorporates concepts of fluid mechanics and partial differential equations

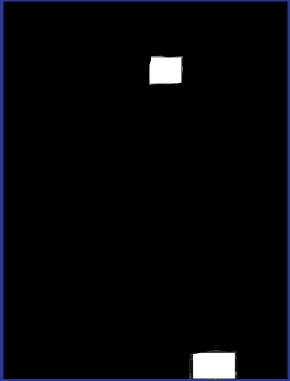
Original Image



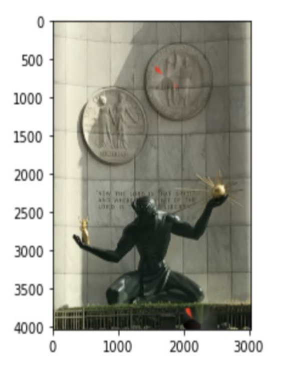
Damaged Image



Mask

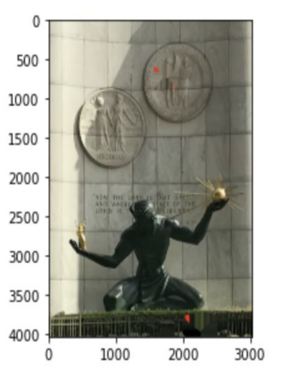


TELEA



(1)

NS



(2)

Summary/Limitations:

Part A

Denoising Grayscale

The Sparse Rep. Journal specified that they used OMP followed by K-SVD to better their Denoising results, we were unable to apply this second step (K-SVD) due to limitations of the scikit dictionary method used.

Part B

Denoising RGB

Initially we intended to use the same image as A, and add random noise, then denoise it, but CV2 colored only works with specified dtype images, so a pre-distorted image was utilized instead. This in turn is why information was derived about the image instead of signal to noise ratios. (Noise information)

Part C

Inpainting

Masking was done manually whereas modern CNN methods can generate the masks. The techniques fail to generalize cases where the surrounding regions might not have suitable information to fill the missing parts. The missing regions also require properties of the would-be-present objects.

Project Conclusion

Insights

- Learned what inpainting was!
- Prior to course, never worked with dictionary learning.
- Acquired understanding of new algorithms for sparse representations

SOURCES:

Research Paper Link: https://ieeexplore.ieee.org/document/4392496

For Part A/B

- https://opencv24-python-tutorials.readthedocs.io/
- https://en.wikipedia.org/wiki/Sparse_dictionary_learning
- https://en.wikipedia.org/wiki/K-SVD
- https://scikit-learn.org/stable/auto_examples/decomposition/plot_image_denoising.html
- https://en.wikipedia.org/wiki/Non-local_means

For Part C/D

- https://en.wikipedia.org/wiki/Inpainting
- https://wandb.ai/site/articles/introduction-to-image-inpainting-with-deep-learning
- https://link.springer.com/content/pdf/10.1007/978-1-4419-7011-4.pdf
- https://www.researchgate.net/publication/238183352_An_Image_Inpainting_Technique_Based_on_the_Fast_Marching_Method