ĐẠI HỌC QUỐC GIA TP.HCM

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**



**MÔN: XỬ LÝ ẢNH SỐ & THỊ GIÁC MÁY TÍNH**

**ĐỒ ÁN MÔN HỌC**

**Giảng viên: TS. NGÔ ĐỨC THÀNH**

Học viên: Lê Nguyễn Sơn Nguyên – CH1702039

TP. Hồ Chí Minh, tháng 12 năm 2019

**MỤC LỤC**

[I. TỔNG QUAN 2](#_Toc31006319)

[II. KỸ THUẬT NHẬN DIỆN NHIỄU VÀ PHƯƠNG ÁN XỬ LÝ 2](#_Toc31006320)

[III. CÁC CÔNG CỤ 5](#_Toc31006321)

[IV. MÔI TRƯỜNG TRIỂN KHAI 5](#_Toc31006322)

[V. KẾT QUẢ 5](#_Toc31006323)

[VI. KẾT QUẢ & NHẬN XÉT 26](#_Toc31006324)

[TÀI LIỆU THAM KHẢO 27](#_Toc31006325)

1. TỔNG QUAN

Bài làm với mục đích áp dụng các kỹ thuật trong phần xử lý ảnh số đã học vào việc “cố gắng” phục hồi lại những hình ảnh đã bị nhiễu, giảm chất lượng như làm mờ, nhiễu muối tiêu, nhiễu gaussian, hoặc kết hợp nhiều loại tác động với nhau. Bài làm xử dụng ngôn ngữ lập trình Python và các thư viện phổ biến kèm theo.

1. KỸ THUẬT NHẬN DIỆN NHIỄU VÀ PHƯƠNG ÁN XỬ LÝ

Nguồn dữ liệu (tập ảnh nhiễu):

<https://drive.google.com/drive/folders/1XjHhQOOuQN-TRbMIldOYMtHBYokH6ntz?usp=sharing>

* ***Mô tả nguồn dữ liệu:***

Dữ liệu chứa 5 ảnh nhiễu với các loại nhiễu khác nhau hoặc kết hợp.

Các bước:

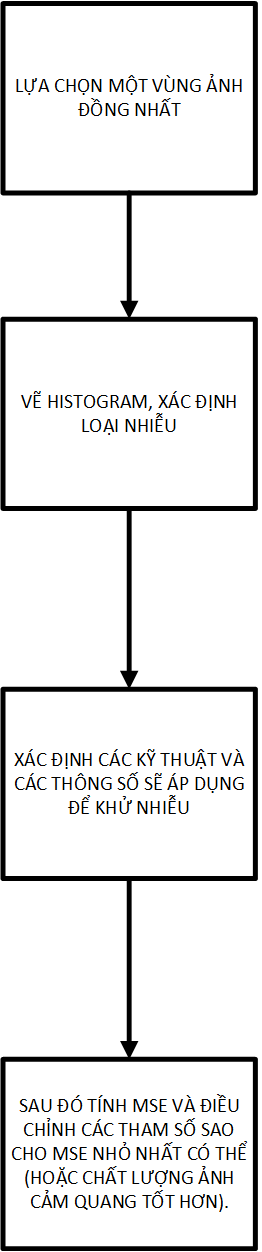


Figure 1 ĐỐI VỚI MIỀN KHÔNG GIAN



Figure 2 ĐỐI VỚI MIỀN TẦN SỐ

1. CÁC CÔNG CỤ

Ngôn ngữ lập trình: Python.

Thư viện sử dụng:

1. OpenCV: xử lý ảnh, load, save, histogram,….
2. Scikit-image: sử dụng model restore và các công cụ hỗ trợ khác,…
3. Numpy, Scipy: dùng để tính toán trên vector: cosine, sort,…
4. Jupyter Notebook: xây dựng các file notebook chạy python tương tác trực tiếp trên nền web.
5. MÔI TRƯỜNG TRIỂN KHAI

* Local: Windows 8.1
* Cấu trúc Notebook:

<https://github.com/lenguyensonnguyen/DIP2019_LNSNGUYEN_CH1702039>

File Jupyter notebook (chi tiết từng bước quá trình làm, code + giải thích) có thể xem trực tiếp tại GibHub.

1. KẾT QUẢ

Đối với ảnh số 1:



Figure 3 ẢNH NHIỄU SỐ 1



Figure 4 ẢNH GỐC SỐ 1

Lấy một phần đồng nhất của ảnh nhiễu

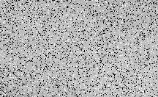


Figure 5 MỘT PHẦN ĐỒNG NHẤT ẢNH NHIỄU SỐ 1

Vẽ Histogram và xác định được nhiễu muối tiêu + nhiễu gaussian

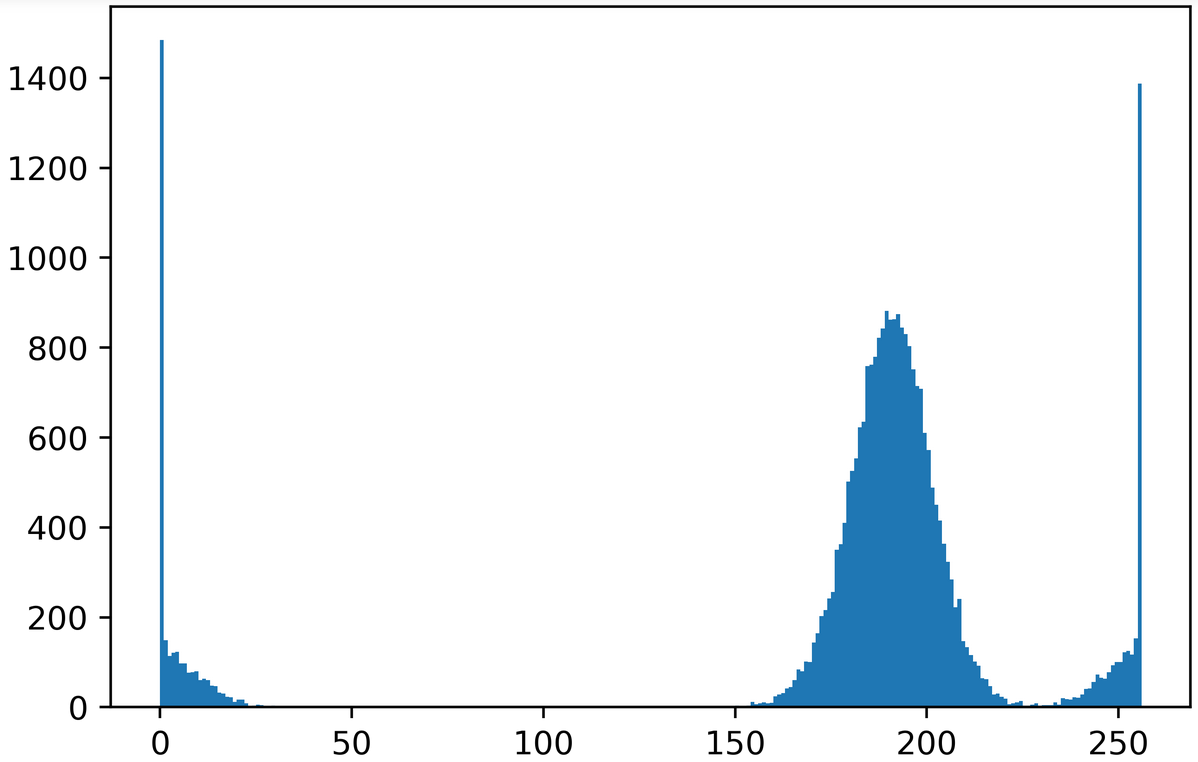


Figure 6 HISTOGRAM CHO PHẦN ĐỒNG NHẤT ẢNH NHIỄU SỐ 1

Áp dụng kỹ thuật **Median** và **Mean** để khử nhiễu Gaussian và muối tiêu.



Figure 7 ẢNH NHIỄU SỐ MỘT SAU KHI XỬ LÝ

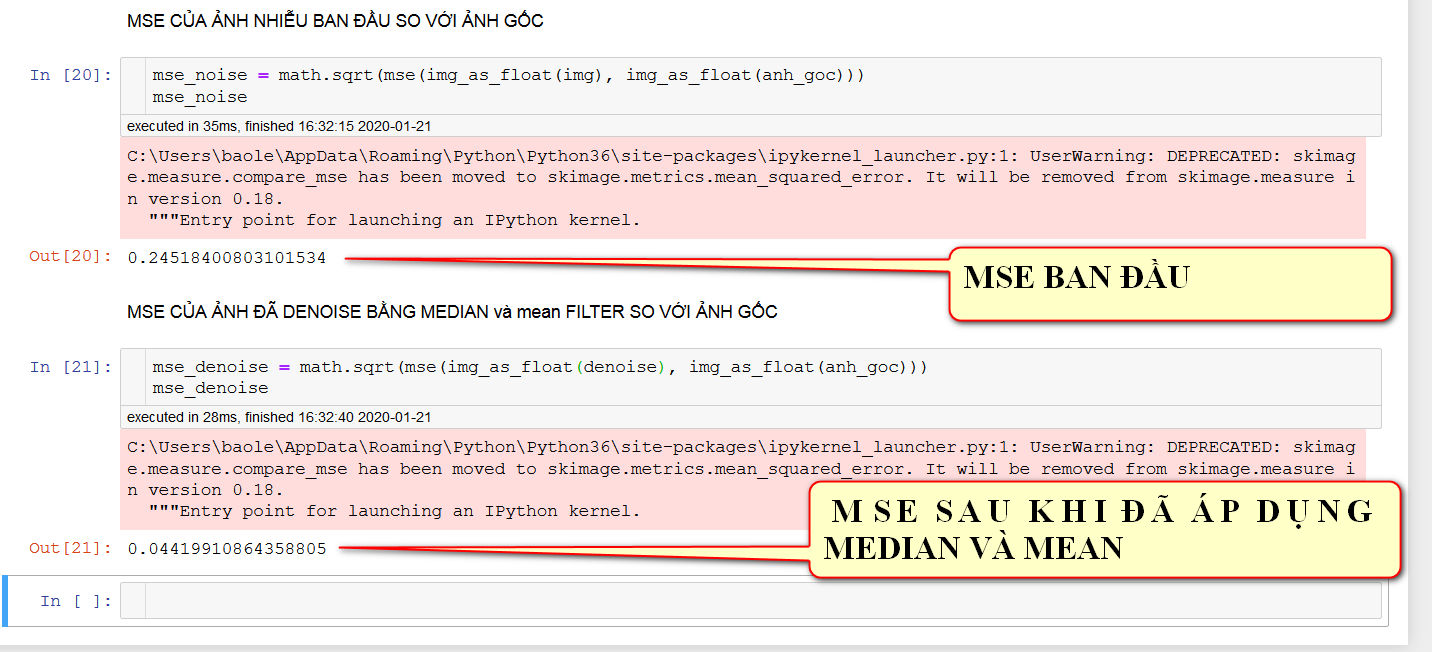


Figure 8 TÍNH MSE CHO PHẦN XỬ LÝ ẢNH NHIỄU SỐ 1

Chi tiết file Jupyter Notebook:

<https://nbviewer.jupyter.org/github/lenguyensonnguyen/DIP2019_LNSNGUYEN_CH1702039/blob/master/restore_image1n.ipynb>

Ảnh nhiểu số 2:



Figure 9 ẢNH NHIỄU SỐ 2



Figure 10 ẢNH GỐC SỐ 2

Ảnh bị Motion Blur + Gaussian Noise nên dùng **Wiener** và **Mean Filter** để phục hồi lại “phần nào” phần bị tác động.



Figure 11 MỘT PHẦN ĐỒNG NHẤT CỦA ẢNH NHIỄU SỐ 2

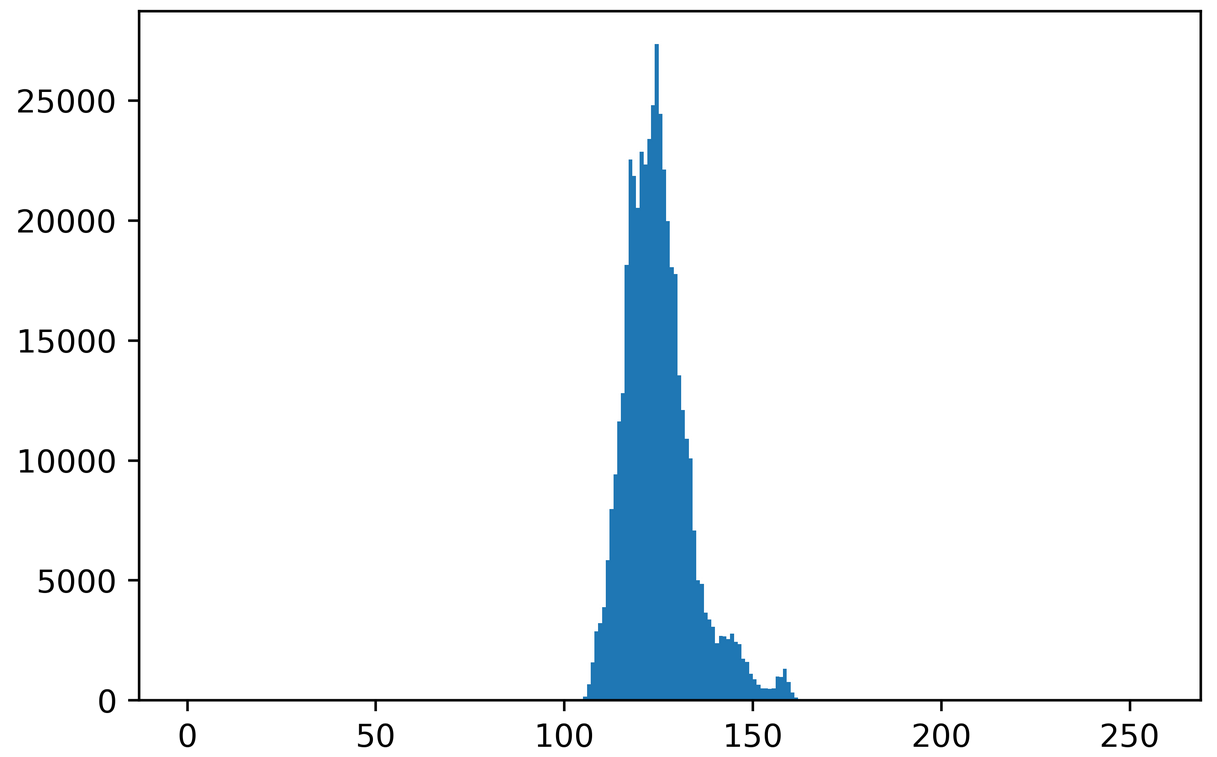


Figure 12 HISTOGRAM PHẨN ĐỒNG NHẤT ẢNH NHIỄU SỐ 2



Figure 13 ẢNH KẾT QUẢ SỐ 2

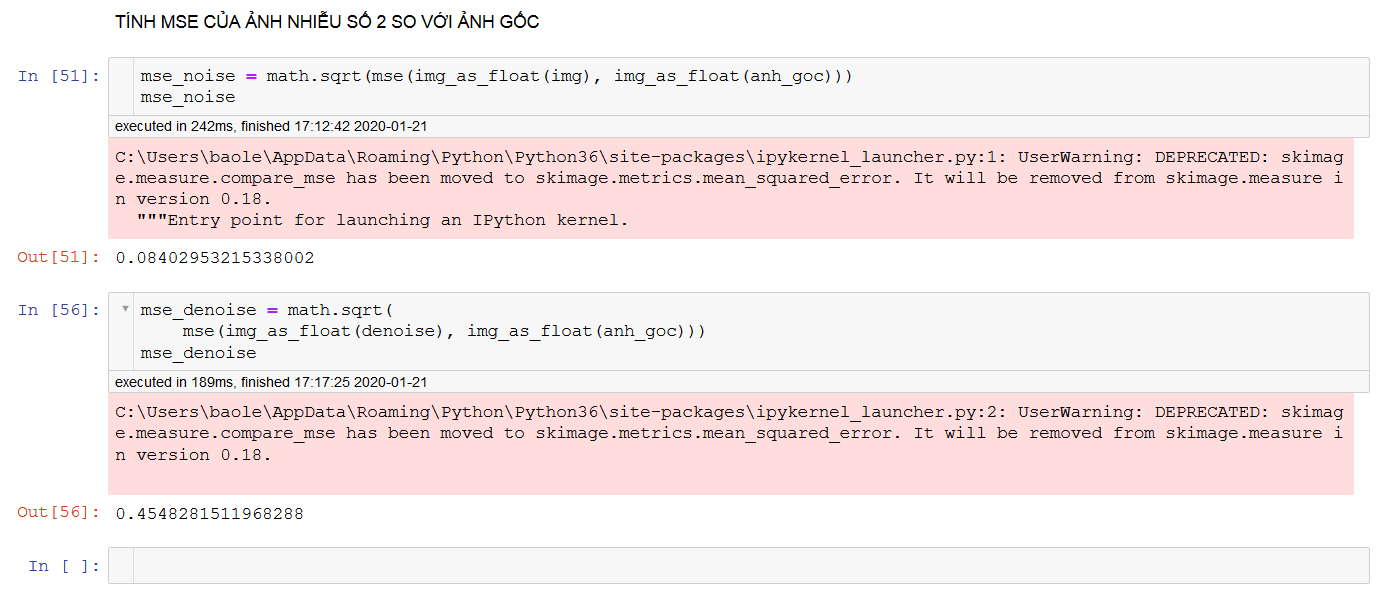


Figure 14 PHẦN TÍNH MSE

Chi tiết file Jupyter Notebook (Ảnh nhiễu số 2):

<https://nbviewer.jupyter.org/github/lenguyensonnguyen/DIP2019_LNSNGUYEN_CH1702039/blob/master/restore_image2n.ipynb>

Ảnh nhiễu số 3:



Figure 15 ẢNH NHIỄU SỐ 3



Figure 16 ẢNH GỐC SỐ 3

Chọn một vùng ảnh đồng nhất.

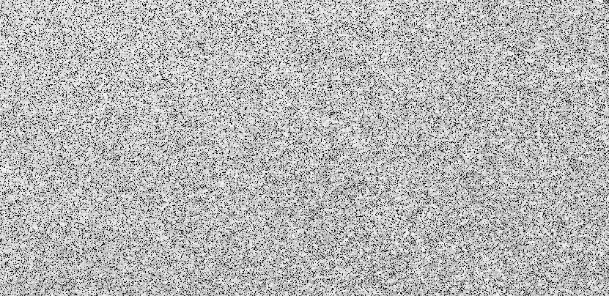


Figure 17 VÙNG ẢNH ĐỒNG NHẤT ẢNH NHIỄU SỐ 3

Vẽ histogram.

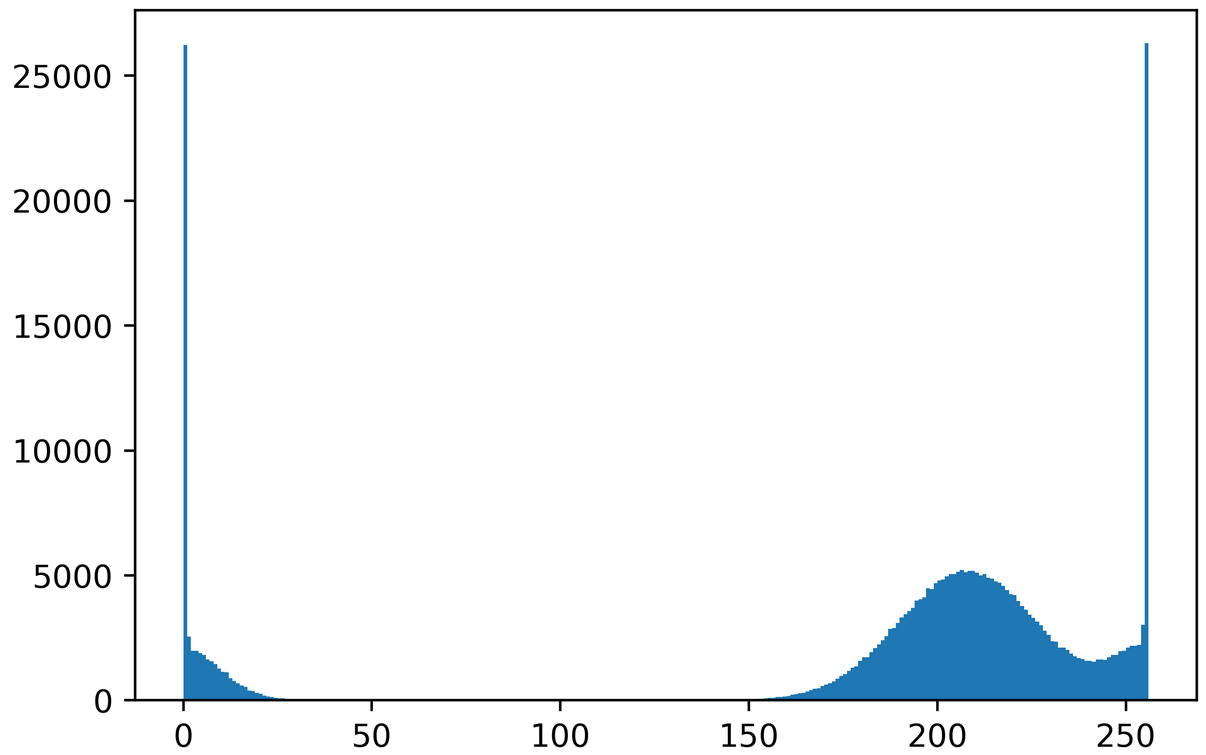


Figure 18 HISTOGRAM PHẦN ĐỒNG NHẤT ẢNH NHIỄU SỐ 3

Xác định loại nhiễu là muối + tiêu.

Áp dụng kỹ thuật **Median** filter.



Figure 19 ẢNH KẾT QUẢ SỐ 3



Figure 20 MSE CHO ẢNH NHIỄU SỐ 3

Chi tiết file Jupyter Notebook (Ảnh nhiễu số 3):

<https://nbviewer.jupyter.org/github/lenguyensonnguyen/DIP2019_LNSNGUYEN_CH1702039/blob/master/restore_image3n.ipynb>

Ảnh nhiễu số 4:



Figure 21 ẢNH NHIỄU SỐ 4

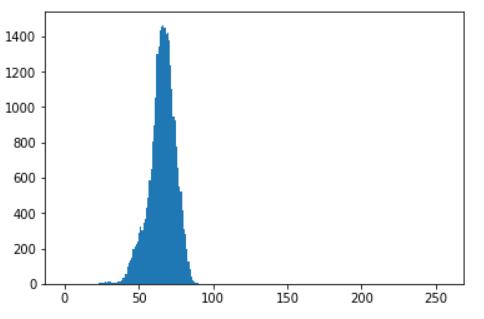


Figure 22 ẢNH GỐC SỐ 4

Chọn một vùng ảnh đồng nhất.



Vẽ histogram.



Xác định là nhiễu Gassian.

Áp dụng **Mean** filter + **Wiener** để phục hồi lại ảnh.



Figure 23 ẢNH KẾT QUẢ SỐ 4

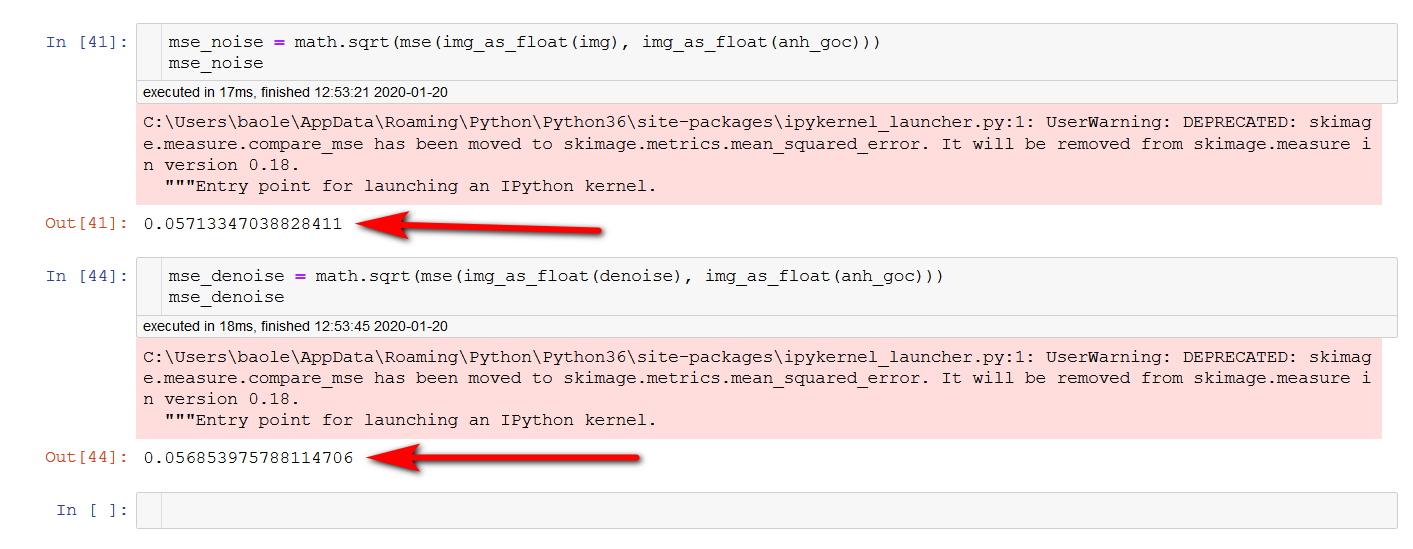


Figure 24 PHẦN TÍNH MSE CHO ẢNH NHIỄU SỐ 4

Chi tiết file Jupyter Notebook (Ảnh nhiễu số 4):

<https://nbviewer.jupyter.org/github/lenguyensonnguyen/DIP2019_LNSNGUYEN_CH1702039/blob/master/restore_image4.ipynb>

Ảnh nhiễu số 5:



Figure 25 ẢNH NHIỄU SỐ 5

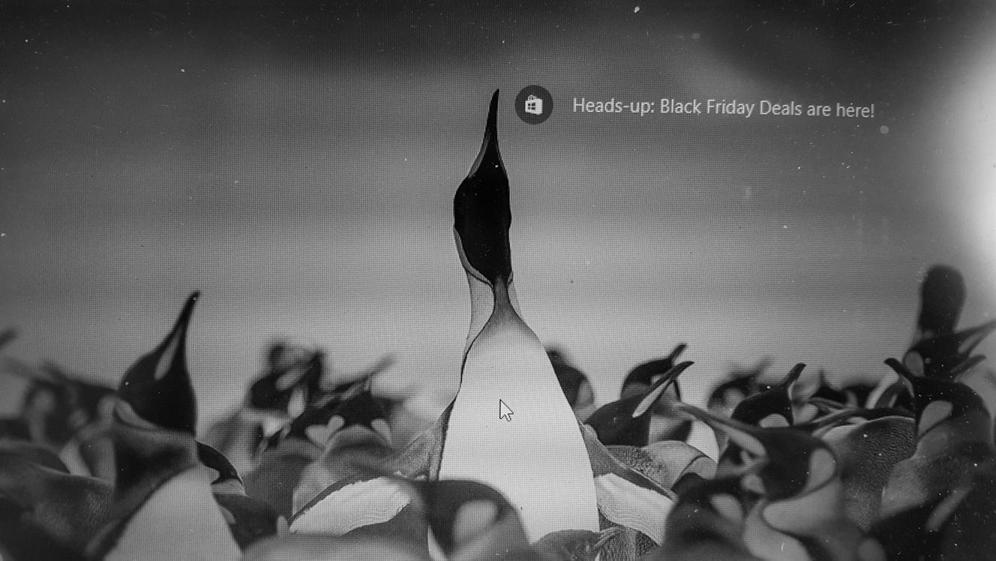


Figure 26 ẢNH GỐC SỐ 5

Vẽ Histogram của ảnh nhiễu và ảnh gốc.

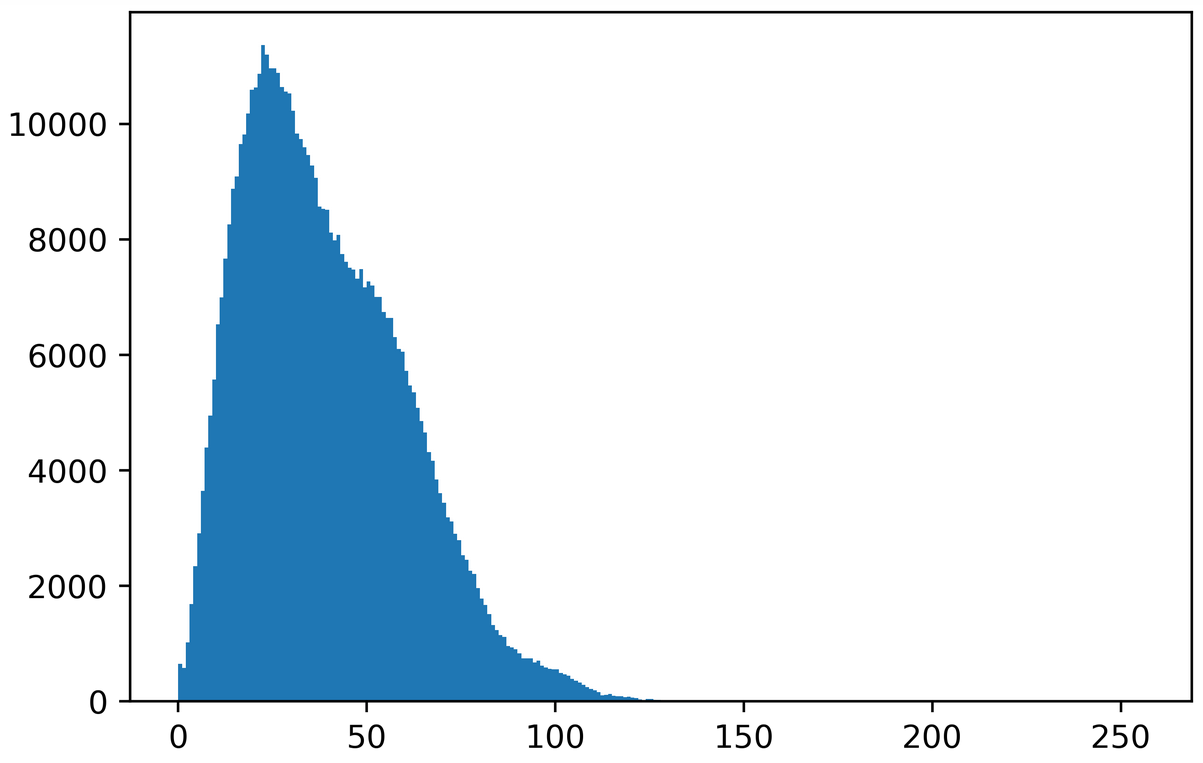


Figure 27 HISTOGRAM ẢNH NHIỄU SỐ 5

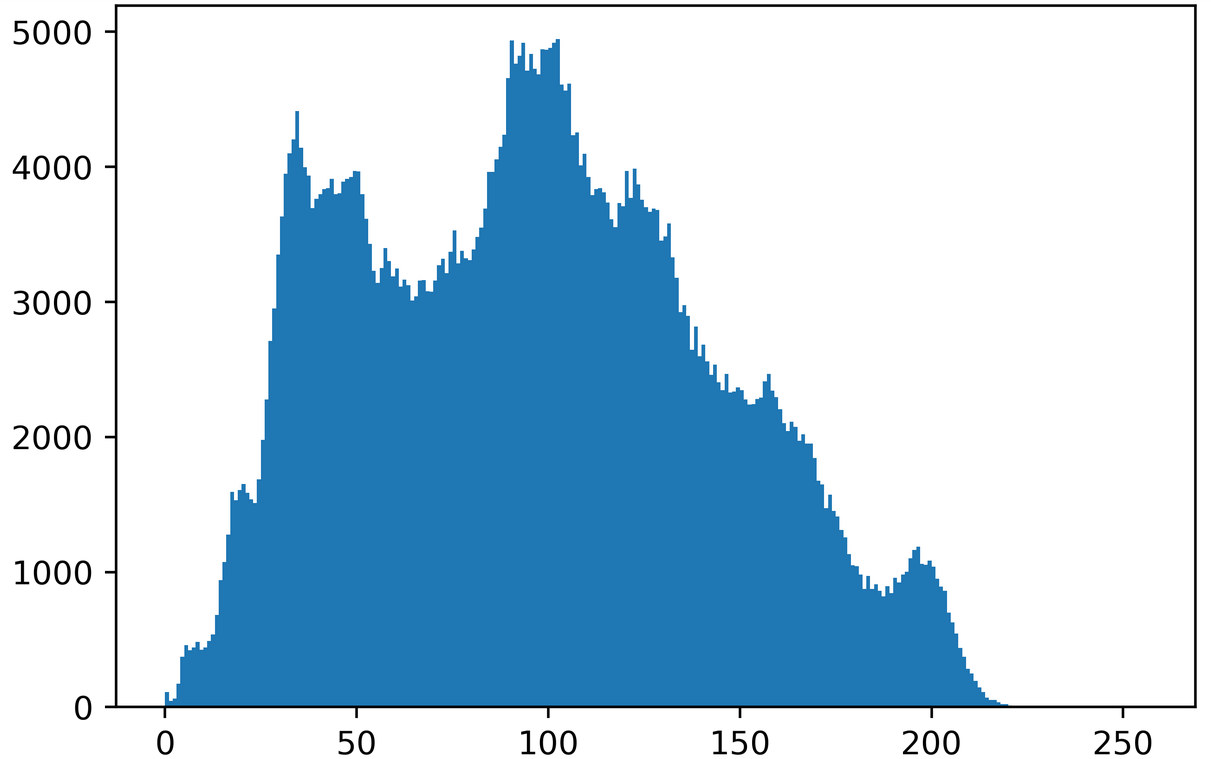


Figure 28 HISTOGRAM ẢNH GỐC SỐ 5

Dùng **HISTOGRAM MATCHING** để xử lý ảnh nhiễu.

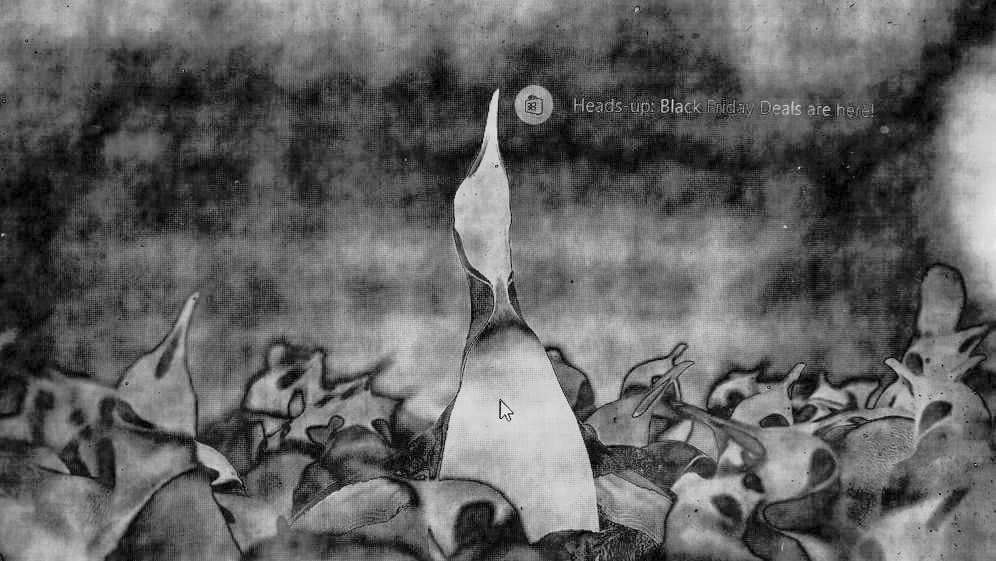


Figure 29 ẢNH KẾT QUẢ SỐ 5

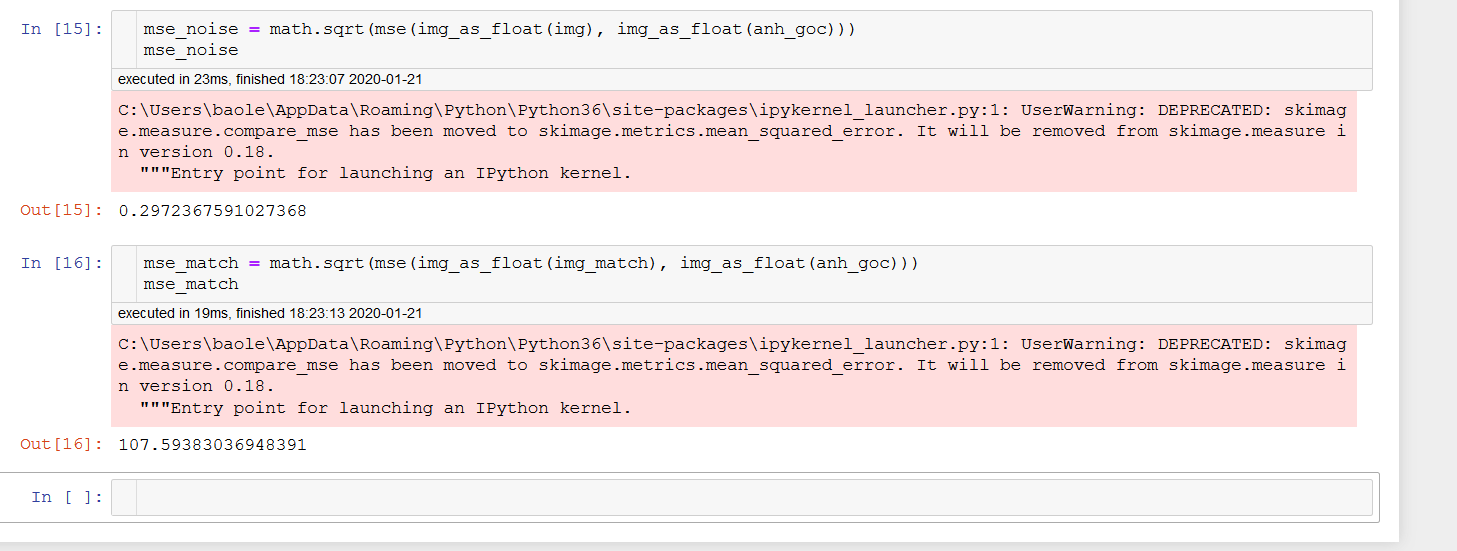


Figure 30 PHẦN TÍNH MSE CHO ẢNH SỐ 5

* Đường link ảnh nhiễu số 5 (dạng Jupyter Notebook trên GitHub):

<https://nbviewer.jupyter.org/github/lenguyensonnguyen/DIP2019_LNSNGUYEN_CH1702039/blob/master/restore_image5n.ipynb>

1. KẾT QUẢ & NHẬN XÉT

* Đã thực hiện việc phân tích các loại nhiễu và áp dụng một số kỹ thuật đã học để phục hồi (denoise) và cho kết quả tương đối khả quan.
* Đa phần các ảnh sau phục hồi có MSE thấp hơn so với ảnh nhiễu ban đầu.
* Tuy nhiên, có một số ảnh do “độ biến dạng (noise)” quá cao nên một số cách phục hồi bình thường không cho kết quả tốt.
* Cần nghiên cứu áp dụng để cải thiện kết quả hơn nữa.

TÀI LIỆU THAM KHẢO

1. Ts. Ngô Đức Thành, *Bài giảng môn Xử lý ảnh số và thị giác máy tính*.