

Restauración de Imágenes con Descenso de Gradiente

Fernando Leon Franco

07-05-2025

1. Carga y Preprocesado de la Imagen

En este proyecto trabajamos sobre una imagen en escala de grises que se carga mediante la clase `Imagen`, la cual encapsula tanto la lectura como el redimensionamiento y el guardado. Al inicializarla, se verifica automáticamente que la ruta exista y se almacenan sus dimensiones en `alto` y `ancho`. A continuación, reducimos el tamaño original dividiendo ambas dimensiones por un factor de **10**, con el fin de acelerar los cálculos sin sacrificar demasiada calidad visual.

2. Generación de Ruido y Parámetros

Para evaluar la robustez de los métodos de optimización aplicamos ruido gaussiano con desviaciones estándar $\sigma = 0, 10, 20, 30, 40, 50$. Cada nivel de ruido se genera sumando una matriz con distribución normal a la imagen original y se procesa de forma independiente para comparar cómo responden los algoritmos ante degradaciones crecientes.

3. Formulación de la Energía L_2

La función objetivo que minimizamos es:

$$J(u) = \frac{1}{2}\|u - f\|^2 + \frac{\lambda}{2}\|\nabla u\|^2, \quad (1)$$

donde f representa la imagen con ruido y $\lambda = 0,2$ regula la suavidad. El término $\|\nabla u\|^2$ utiliza el laplaciano discreto de cinco puntos para capturar la curvatura local.

4. Métodos de Descenso de Gradiente

Exploramos tres variantes clásicas: descenso **simple**, **momentum** y **Nesterov**. Cada método corre hasta **1500 iteraciones** o hasta que la norma del gradiente sea menor que $\epsilon = 10^{-6}$. La tasa de aprendizaje α se prueba en **0.001, 0.02, 0.5, 1.0 y 2.0**, y para momentum y Nesterov usamos $\beta = 0,8$.

5. Visualización y Métricas de Calidad

Al finalizar cada experimento generamos figuras con tres filas y tres columnas: la primera fila muestra la imagen original, la segunda la imagen con ruido

y la tercera las restauraciones. Calculamos **PSNR** (acelerado con Numba) y **SSIM** para medir calidad estructural.

6. Resultados de Ruido Bajo ($\sigma = 10$)

Con $\alpha = 0,02$ y $\beta = 0,8$:

Método	Iteración mínima	Norma mínima	P
Simple	623	$2,22 \times 10^{-2}$	
Momentum	182	$2,54 \times 10^{-3}$	
Nesterov	151	$2,38 \times 10^{-3}$	

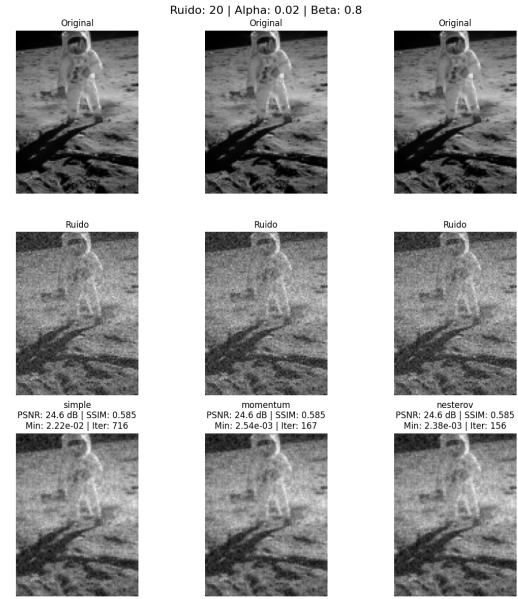


Figura 2: Desempeño con ruido incrementado ($\sigma = 20$, $\alpha = 0,02$, $\beta = 0,8$)

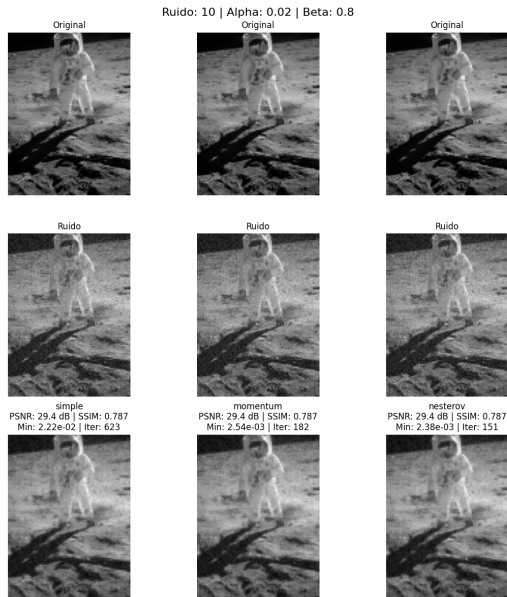
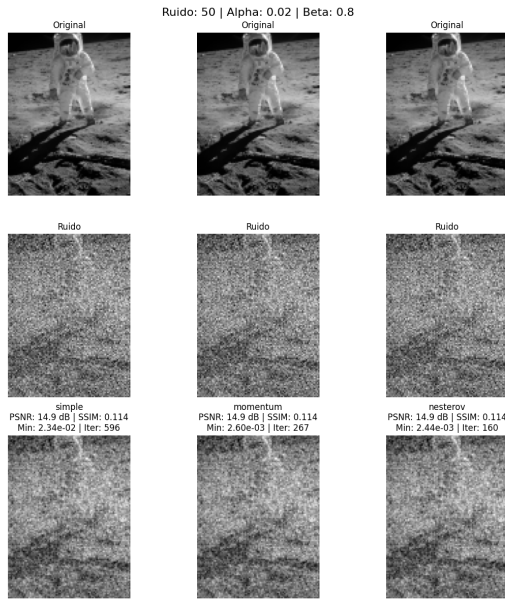


Figura 1: Comparación visual del desempeño con $\sigma = 10$, $\alpha = 0,02$, $\beta = 0,8$



[2] ChatGPT, *Discusión técnica sobre optimización y restauración de imágenes*. [En línea]. Disponible: <https://chatgpt.com/share/681bc65d-fd68-8002-9bb2-cbd5f8291eb2>

Figura 3: Comparación visual con alto nivel de ruido ($\sigma = 50$, $\alpha = 0,02$, $\beta = 0,8$)

7. Conclusiones

La combinación de una energía L_2 bien balanceada con descenso acelerado (Nesterov) entrega la restauración más eficiente frente a niveles moderados de ruido. Mientras que valores de α **muy pequeños** ralentizan la convergencia, valores **demasiado grandes** generan inestabilidad extrema. El valor óptimo identificado es $\alpha = 0,02$.

Referencias

- [1] Machine Learning Mastery, *Gradient Descent with Nesterov Momentum*. [En línea]. Disponible: <https://machinelearningmastery.com/gradient-descent-with-nesterov-momentum-from-scratch/>