

CIMAT

Reporte de proyecto final de AutoML

06 de Diciembre de 2023

Autor: María Guadalupe López Salomón

Maestría en Cómputo Estadístico, CIMAT Sede Monterrey

Eliminando ruido con NAS-DIP

Resumen

Se presenta un método para mejorar la restauración de dos imágenes usando la búsqueda de arquitecturas neuronales y deep image prior. Se emplea el diseño de espacios de búsqueda para optimizar la celda de upsampling y las conexiones residuales cross-level. Utilizando aprendizaje por refuerzo para identificar estructuras de red eficientes sin aprender de datos etiquetados.

Palabras clave: NAS: neural architecture search, DIP: deep image prior, NNI: neural network intelligence, denoising restauración imágenes

Introducción

La eliminación de ruido es una tarea fundamental dentro del procesamiento de imágenes debido a su gran utilidad en diversas aplicaciones. La contaminación por ruido degrada la calidad visual de las imágenes recolectadas y puede afectar el análisis y la implementación de tareas subsecuentes de análisis de imágenes como clasificación, segmentación, entre otras. Recientemente las redes neuronales convolucionales (CNN), han sido exitosamente implementadas en tareas de visión computacional, proporcionando herramientas poderosas para la eliminación de ruido en imágenes o *denoising* [N2N]. Parte del éxito detrás de las (CNNs), reside en su capacidad de aprender *priors* a partir de datasets de gran escala (los *priors* se encuentran en los parámetros y pesos de la red entrenada) [TrueNas]. Tales estructuras de imagen, *priors*, codificadas en la arquitectura de la red son fundamentales para la restauración de imágenes y en particular, para el *denoising*[TrueNas]. Sin embargo, los modelos eliminadores de ruido basados en CNNs dependen enormemente de un gran número de pares de imágenes para entrenarse, además de que es difícil y costoso obtener grandes cantidades de pares de datos etiquetados, limpios y con ruido, para el entrenamiento. Aunado a esto, el rendimiento de los modelos

entrenados con ruido sintético disminuye notablemente al aplicarlos a ruido real debido a diferencias significativas entre ambos tipos de ruido[N2N].

No obstante, se ha demostrado que la inicialización aleatoria de una red neuronal convolucional (CNN) puede servir como *prior* y como técnica efectiva de regularización para la eliminación de ruido [DIP-NAS-EV]. Dicha técnica se conoce como *DIP: deep image prior* o *Prior de imagen profunda* y permite explotar tanto la similitud de texturas como la equivariancia de traslación de las CNNs para producir resultados competitivos de tareas de restauración de imágenes como el *denoising* [DIP-NAS-EV].

Sin embargo, es importante resaltar que la eficacia de DIP depende de la arquitectura de la CNN, es decir, DIP es eficaz cuando una persona está involucrada en la optimización de la CNN para obtener buenos resultados [DIP-NAS-EV].

En este contexto, artículos recientes han mostrado que la estructura de las redes neuronales profundas puede ser utilizada como una estructura de la imagen a priori para realizar diversas tareas de restauración de imágenes[TrueNas]. En particular, los algoritmos basados en *NAS: neural architecture search* o *búsqueda de arquitectura neuronal* han mostrado ser efectivos en el descubrimiento de redes con buen rendimiento en grandes espacios de búsqueda, además de que han sido implementados eficazmente en diversas tareas de eliminación de ruido [TrueNas][DIP-NAS-EV]. En el diseño de redes para eliminar el ruido de imágenes, existen dos aspectos fundamentales: el diseño del *encoder* y el diseño del *decoder* (muestreo ascendente de celdas o upsampling cell) que permita preservar los detalles de la imagen original en el proceso de reconstrucción.

A través de NAS queremos descubrir una arquitectura de CNN que capture contundentemente los priors de imágenes para la tarea de restauración *learning-free* de eliminación de ruido. Es decir, a través de NAS se busca utilizar el enfoque de Deep Image Prior (DIP)

para automatizar la identificación de la arquitectura más adecuada de la red neuronal convolucional CNN, particularmente la parte del encoder-decoder para la tarea de eliminación de ruido en dos imágenes de la base de datos “Miscellaneous”.

A continuación, se presenta una breve descripción de los métodos y datos utilizados. Posteriormente, se describe la metodología implementada y los resultados obtenidos con la respectiva discusión y análisis de los mismos. Finalmente, se comentan las conclusiones obtenidas a partir de la implementación de este proyecto.

Materiales y Métodos

De acuerdo con [TrueNas], en la búsqueda del diseño de una estructura *encoder/decoder* apropiada para la eliminación de ruido, se deben considerar dos aspectos fundamentales, el primero es el diseño del *decoder* y en particular el diseño de la *celda de upsampling* que han recibido poca atención en comparación con el diseño de *encoders*. Y el segundo, dado que la resolución espacial de las características decrece progresivamente a lo largo del encoder de características, es crucial para la red poder recuperar mapas de características con gran resolución espacial. Una de las propuestas para enfrentar este desafío es diseñar *patrones de conexión residuales o de salto*, los cuales permiten conectar características de diferentes niveles o capas de la red, entre el encoder y el decoder (también conocidas como *cross-level features*). De manera que se propone realizar la búsqueda de *celdas de upsampling* para el *decoder* y *patrones de conexión de salto* entre el *encoder-decoder*, considerando espacios de búsqueda para ambos componentes propuestos en [TrueNas].

Recientemente los algoritmos de **NAS: neural architecture search** por sus siglas en inglés, han sido aplicados exitosamente en diferentes tareas de restauración, clasificación, segmentación de imágenes, entre muchas otras y se pueden agrupar en diversas categorías que dependen del *algoritmo de búsqueda*. Los métodos primarios son evolución, aprendizaje por refuerzo y búsqueda diferenciable [DIP-NAS-EV][TrueNas]. Los algoritmos basados en *aprendizaje por refuerzo (RL)* adoptan estrategias que les permiten entrenar una red recurrente que produce una secuencia de símbolos que describen la arquitectura de una red o una estructura de celda repetible. En particular, en la implementación de este proyecto, el algoritmo de búsqueda basado en aprendizaje por refuerzo (RL) con una red recurrente (RNN) como controlador permite realizar la búsqueda de la *celda de upsampling* y los *patrones de conexión cross-level*.

Por lo tanto, motivados por el algoritmo de búsqueda de arquitecturas neuronales (NAS), el cual ha mostrado excelente rendimiento en espacios de búsqueda de tamaño considerable, retomamos la implementación realizada por (Y., Chen et al.) [TrueNas], donde se aprovecha el *aprendizaje por refuerzo (RL)* con un con-

trolador de *red neuronal recurrente (RNN)* y se usa la proporción *peak signal to noise ratio (PSNR)* como recompensa que guía la búsqueda de la arquitectura. Al buscar simultáneamente la *celda de upsampling* y las *conexiones de características cross-level*, se pretende descubrir una arquitectura de CNN que sea capaz de capturar conocimientos o *priors* sólidos y estructurados de la imagen en la eliminación de ruido.

Se ha demostrado en recientes estudios, que al mapear ruido aleatoriamente a una imagen degenerada, la red CNN no entrenada, es capaz de realizar tareas de eliminación de ruido o *denoising* con rendimiento competitivo [N2N][DIP][TrueNas]. Con el propósito de descubrir arquitecturas que capturen *priors* de imágenes más sólidos, consideraremos la búsqueda de una *celda de upsampling* y un *patrón de conexiones residuales cross-scale* sin aprender de datos emparejados. Para lo cual aplicaremos una búsqueda de aprendizaje por refuerzo (RL) con un controlador de red recurrente (RNN) usando la relación de señal-ruido (PSNR) como recompensa para buscar la mejor estructura de red f_θ^* . Posterior al paso de NAS, se reinician aleatoriamente los pesos de la estructura de red con mejor rendimiento f_θ^* y se optimiza el mapeo desde el ruido aleatorio a la imagen degenerada.

DIP:(*Deep image prior*) por sus siglas en inglés, propone emplear una red neuronal convolucional (CNN) aleatoriamente inicializada que muestre una imagen, usando su estructura como un prior de la imagen; este método no requiere de aprendizaje previo pero produce resultados limpios con bordes más nítidos [DIP].

Las redes profundas se aplican en la generación de imágenes al aprender redes *encoder/decoder* $x = f_\theta(z)$ que mapean un vector codificado z a una imagen x . Este enfoque es bastante útil para muestrear imágenes provenientes de distribuciones condicionadas a una observación degradada x_0 y resolver problemas como la eliminación de ruido. El objetivo de DIP reside en investigar implícitamente el *prior* o conocimiento capturado previamente por la elección particular de una estructura de red generadora, antes de que se aprenda algún parámetro [DIP].

De modo que para descubrir las estructuras de redes que capturen priors de imágenes más contundentes, consideraremos la tarea de buscar la mejor estructura del encoder-decoder, f_θ^* , sin aprender de datos emparejados.

DIP considera la siguiente parametrización de la red neuronal $x = f_\theta(z)$, de una imagen $x \in R^{H \times W \times 3}$ en el conjunto de entrenamiento, donde $z \in R^{H \times W \times C}$ es un tensor/vector codificado y θ representa a los parámetros de la red [DIP]. Generamos una versión *degradada* x_0 al agregar ruido a la imagen limpia x . Posteriormente, muestreamos una imagen con ruido $z \in R^{H \times W \times C}$ y forzamos a la red buscada f_θ a mapear la imagen de ruido z a x . Para llevar a cabo la *eliminación de ruido o denoising* entrenamos la red usando

la siguiente función objetivo [**TrueNas**]:

$$f_\theta^* = \mathcal{L}_{denoising}(\theta) = \|f_\theta(z) - x_0\|_2^2 \quad (1)$$

donde $f_\theta(z) \in R^{H \times W \times 3}$ es la imagen *limpia o sin ruido* x . La función objetivo anterior, nos conduce al siguiente problema de optimización:

$$\min_{\theta} \|f_\theta(z) - x_0\|^2 \quad (2)$$

De manera que, podemos recuperar la imagen limpia $x^* = f_{\theta^*}(z)$ de una observación ruidosa x_0 después de sustituir el valor θ^* que minimice la expresión (2).

De manera general, a partir de las imágenes originales, también denominadas *ground-truth* se puede obtener la relación señal-ruido (PSNR) entre la imagen original y la salida de la red $f_\theta(z)$ y determinar así, la mejor estructura f_{θ^*} .

0.1. Descripción de datos utilizados

Debido a la complejidad y recursos computacionales que conlleva la implementación del algoritmo NAS-DIP, se tomaron en cuenta únicamente dos imágenes pertenecientes a la base de datos “Miscellaneous”, que se muestran en las figuras (1)



Figura 1: Airplane F-16

y (2), cuyo tamaño es de 512×512 píxeles para cada una, ambas se encuentran en escala de color RGB.



Figura 2: Sailboat in lake

Así mismo se definieron dos espacios de búsqueda, retomando los definidos en el artículo de [**TrueNas**], donde el espacio de búsqueda de la *celda de upsampling* consiste de cinco características fundamentales: *submuestreo de características espaciales, transformación de características, tamaño del kernel, tasa de dilatación y capa de activación*. El espacio de búsqueda definido para la *celda de upsampling* se muestra en el

anexo 1. Cada paso del conjunto posee opciones discretas. La trayectoria azul indica el submuestreo bilineal aditivo, el rojo corresponde al stride 2 con convolución transpuesta y el verde representa la convolución de sub-píxeles.

Experimentos y Resultados

La implementación se llevó a cabo en Google Collab, usando una unidad de procesamiento gráfico NVIDIA V100 con 100GB de memoria. Así mismo, el modelo implementado se extrajo del repositorio de GitHub que se exhibe en referencias. El modelo se implementó usando PyTorch y el código asociado se envía como anexo del presente proyecto.

Se consideraron 1200 iteraciones por 15 repeticiones para cada imagen, un valor de la desviación estándar asociado al ruido agregado de $\sigma = 25$, valor considerado como se propone inicialmente en el artículo de DIP [**DIP**]. Se utilizó una tasa de aprendizaje de $l_r = 0.01$

Para las dos imágenes analizadas, consideramos los mismos espacios de búsqueda definidos para la *celda de upsampling* y las *conexiones residuales cross-scale* implementados en el artículo [**TrueNas**].

Es importante mencionar que debido a la complejidad computacional y a la cantidad de parámetros e hiperparámetros asociados a la implementación del algoritmo NAS-DIP, nuestra implementación se basó en gran medida en la implementación del repositorio que se cita en referencias. En el presente proyecto se incluyen los parámetros y medidas más significativos.

Explicación del enfoque NAS-DIP

1. Búsqueda de Celdas Upsampling y Conexiones Residuales Cross-Scale:

- **Celda de upsampling:** Una parte del modelo que se encarga de aumentar la resolución de las imágenes durante el procesamiento en la red.
- **Patrones de conexiones Residuales Cross-Scale:** Se refieren a cómo las diferentes capas de la red están interconectadas, especialmente cómo las características de distintas escalas y resoluciones se combinan.

2. Aprendizaje por Refuerzo (RL) y Controlador de Red Neuronal Recurrente (RNN):

- Se utiliza un controlador RNN para explorar diferentes arquitecturas de red.
- El RL se aplica para “premiar” las arquitecturas que producen mejores resultados, medidos mediante la relación señal-ruido (PSNR).

3. Optimización de la Red

- Después de encontrar la arquitectura óptima f_θ^* en una base de datos de entrena-

miento, esta estructura se usa para optimizar imágenes específicas.

- En este proceso, se parte de pesos aleatorios y se ajustan para reconstruir o mejorar una imagen dada.

Cuando implementamos NAS-DIP para dos imágenes únicamente, no tenemos como tal “conjuntos de entrenamiento” y “conjuntos de prueba”. En este contexto, las imágenes individuales analizadas actúan como entradas (diferentes entre sí) para optimizar la red. Para cada una de las imágenes analizadas NAS-DIP buscó la mejor arquitectura y luego la utilizó para reconstruir o mejorar esa imagen específica. En ambos procesos se ajustaron los pesos de la red (inicializados aleatoriamente) para que la salida de la red se acerque lo más posible a la imagen objetivo (imagen original). Al efectuar el código en dos imágenes por separado, se efectuaron dos búsquedas de arquitecturas y posteriormente se aplicaron dichas arquitecturas óptimas para procesar ambas imágenes.

A través de este enfoque no podemos permitirnos encontrar una generalización de arquitecturas debido a la escasez de datos. La complejidad computacional aunado a la cantidad de recursos de tiempo y procesamiento son un factor limitante dentro de esta implementación. No obstante, la implementación de NAS-DIP a imágenes en solitario permite centrarnos en la optimización de la arquitectura de la red para cada una de las imágenes en específico. El modelo busca aprender la mejor forma de procesar y mejorar cada imagen en particular, más que generalizar a partir de múltiples ejemplos. Consideración muy importante dentro del procesamiento de imágenes, puesto que es muy complicado generalizar una arquitectura que pueda eliminar perfectamente el ruido en imágenes que tienen estructuras completamente diferentes.

Para cada imagen analizada se obtuvo la mejor imagen de todas las iteraciones en cada repetición y la última imagen obtenida en la repetición, así como el valor obtenido de la métrica de PSNR para cada una, el cual representa una medida que se utiliza comúnmente para evaluar la calidad de las imágenes reconstruidas o comprimidas en comparación con las originales. Se basa en el error cuadrático medio (MSE) entre la imagen original y la imagen procesada. Primero, se calcula el MSE entre la imagen original y la imagen reconstruida o procesada. El MSE se calcula promediando el cuadrado de la diferencia entre los píxeles correspondientes de las dos imágenes:

$$MSE = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [I(i, j) - K(i, j)]^2$$

Donde I es la imagen original, K , es la imagen reconstruida, y m, n son las dimensiones de las imágenes. Una vez que se ha calculado el MSE, el PSNR se calcula a través de

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

donde PSNR es el peak signal-to-noise ratio, MAX_I es el valor máximo posible para un píxel en la imagen. Y MSE es el error cuadrático medio. Un valor más alto de PSNR generalmente indica una mejor calidad de la imagen reconstruida en comparación con la original. Sin embargo, se ha mostrado que el PSNR no siempre se correlaciona perfectamente con la percepción de calidad de imagen humana. A veces, una imagen con un PSNR más bajo puede parecer de mejor calidad a los ojos humanos que una con un PSNR más alto. En aplicaciones de compresión de imágenes, un PSNR en el rango de 30 a 50 dB suele considerarse aceptable o bueno. Para tareas de restauración o reconstrucción de imágenes, un PSNR más alto es generalmente deseable.

Así mismo, con el propósito de obtener mayor información acerca de la estructura de cada imagen, se obtuvo la transformada de Fourier de cada imagen obtenida. La transformada de Fourier nos proporciona la información en frecuencias de la señal o imagen muestreada por lo que si existe la presencia de ruido o de otro tipo de aberraciones, el espectro de frecuencias de cada una de las imágenes lo exhibirá. Consideraremos la implementación de esta medida como otra forma de corroborar que a medida que aumentaban el número de iteraciones y el número de repeticiones en las imágenes, se observaban espectros de Fourier para cada una de las imágenes más limpios y definidos exhibiendo así la estructura real de la imagen libre de ruido. Se usó como forma de validar que efectivamente nuestra arquitectura NAS-DIP iba optimizando la función objetivo, permitiendo limpiar el ruido presente en la imagen degradada inicial.

Exhibimos las restauraciones de las primeras repeticiones y las últimas para cada imagen, así como su espectro de Fourier asociado en los anexos.

Conclusiones

A partir de los resultados obtenidos, podemos notar que la tarea de eliminación de ruido se cumple satisfactoriamente con la implementación del modelo. Es evidente la mejora significativa en la calidad de las imágenes pues ya no presentan ruido al llegar a las últimas repeticiones, incluso desde las primeras repeticiones se observan mejoras visuales en la presencia de speckle en las imágenes. Sin embargo, es importante remarcar que el costo computacional que tiene la implementación de este modelo es bastante significativo, por lo que no se recomienda para tareas de restauración de imágenes sencillas. Si bien la implementación de NAS-DIP como técnica de eliminación de ruido tuvo un rendimiento bastante remarcable como se puede ver en el valor del PSNR, el cual fue alto considerando el tipo de tarea, hay que tener en cuenta la clase de tareas que se van a realizar, los objetivos que se tengan, así como la base de datos disponible. Si son bases

de datos que corresponden al mismo tipo de imagen, la implementación de NAS-DIP va a ser inmejorable, si por el contrario son imágenes diferentes, el rendimiento que proporcionará NAS-DIP puede ser sobrepasado por otro tipo de modelos menos costosos. Finalmente, es importante destacar que se deben de tomar en cuenta otro tipo de métricas a la hora de medir la calidad de la restauración, ya sean criterios basados en histogramas y frecuencias relativas de los niveles de grises en los píxeles, o bien basados en la distribución espacial de los mismos como el criterio de máxima entropía. Métricas que nos puedan aportar mayor información no solo sobre la calidad de la reconstrucción y el rendimiento del modelo implementado sino también sobre los priors de la imagen para implementar mejoras en los diseños de espacios de búsqueda para tareas de procesamiento futuras.

Referencias

- [1] K. HO & A. GILBERT AND H. JIN & J. COLLO-MOSSE(2021). *Neural architecture search for deep image prior*. [Elsevier: Computer and Graphics].
- [2] Y. CHEN & C. GAO & H. JIN & E. ROBB & J. HUANG & V. TECH(2020). *NAS-DIP: Learning Deep Image Prior with Neural Architecture Search*. [Computer Vision and Pattern Recognition (cs.CV)].
- [3] D. ULYANOV & A. VEDALDI & V. LEMITSKY(2018). *Deep Image Prior*. [Computer Vision Foundation].
- [4] T. HUANG & S. LI AND X. JIA & H. LU & J. LIU(2021). *Neighbor2Neighbor: Self-Supervised Denoising from Single Noisy Images*. [Computer Vision and Pattern Recognition (cs.CV)]
- [5] ARICAN, METIN & KARA, OZGUR & BREDELL, GUSTAV & KONUKOGLU, ENDER(2022). *ISNAS-DIP: Image-Specific Neural Architecture Search for Deep Image Prior* [Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition]

Anexo 1

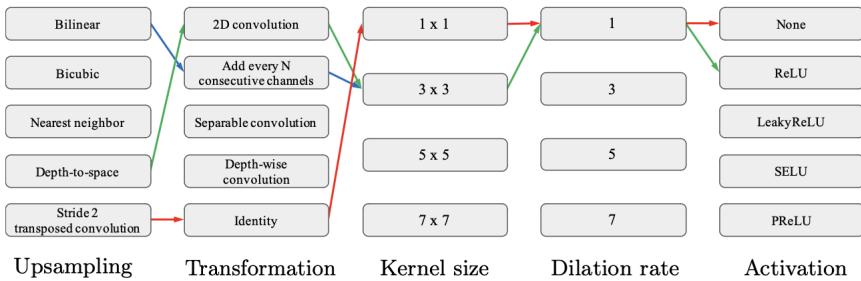


Figura 3: Espacio de búsqueda definido para la celda de upsampling



Figura 4: Imagen 1 degradada con ruido

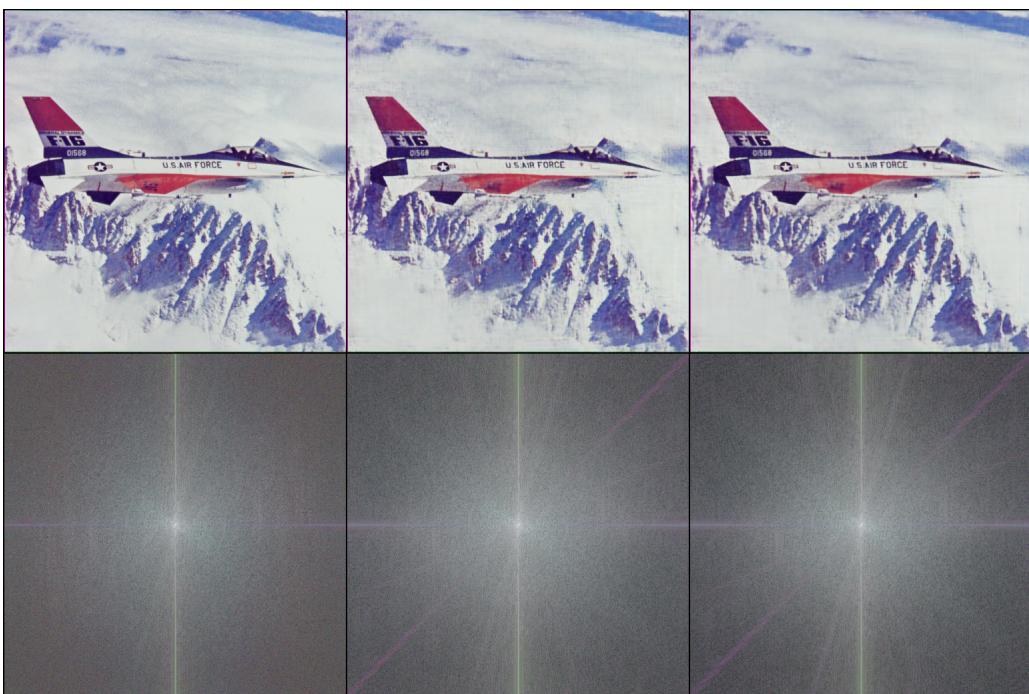


Figura 5: Mejor imagen y última imagen obtenida en la primera repetición. Espectro de frecuencias asociado. La mejor iteración obtenida en esa repetición fue la 1188. El mejor valor asociado del PSNR fue de 31.287

Anexo 2

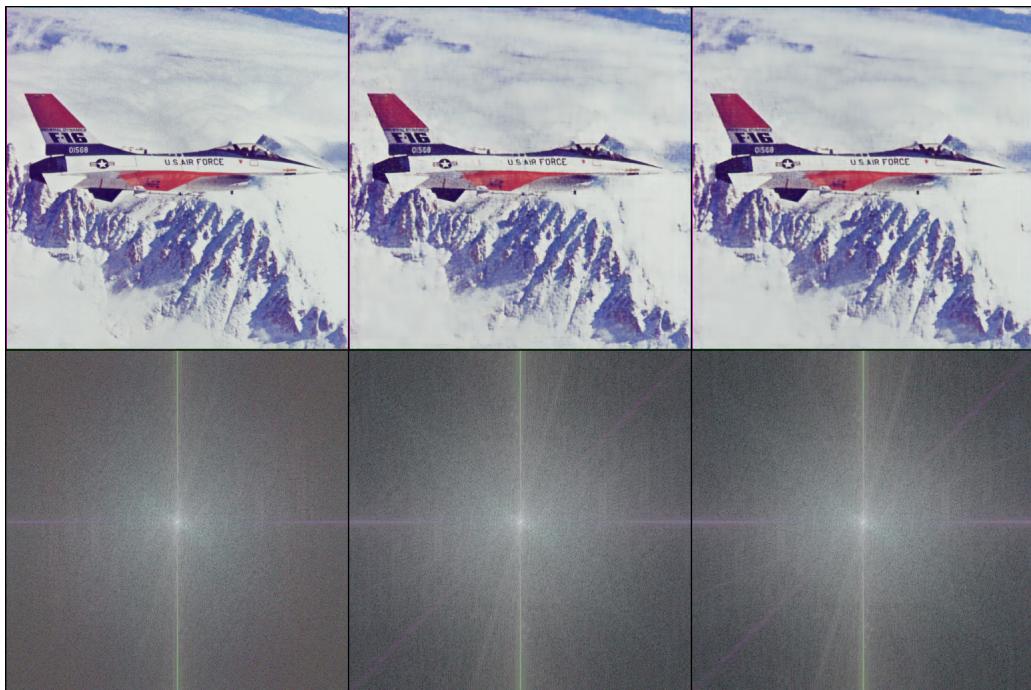


Figura 6: Mejor imagen y última imagen obtenida en la última repetición. Espectro de frecuencias asociado. La mejor iteración obtenida en esa repetición fue la 1184. El valor mejor PSNR asociado fue de 31.393

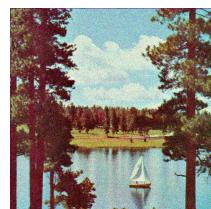


Figura 7: Imagen 2 degradada con ruido

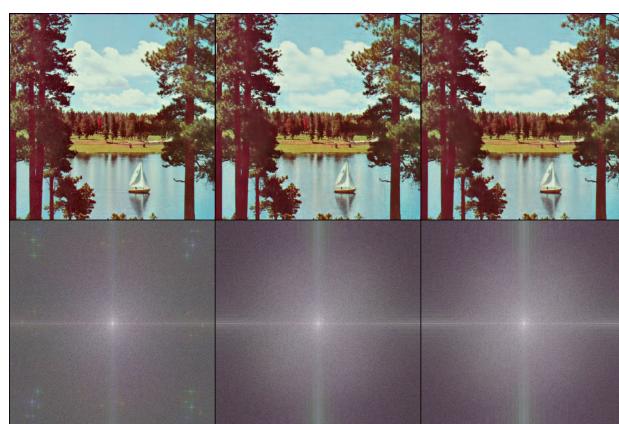


Figura 8: Mejor imagen y última imagen obtenida en la primera repetición. Espectro de frecuencias asociado. La mejor iteración obtenida en esa repetición fue la 1182. El valor mejor PSNR asociado fue de 27.407

Anexo 3

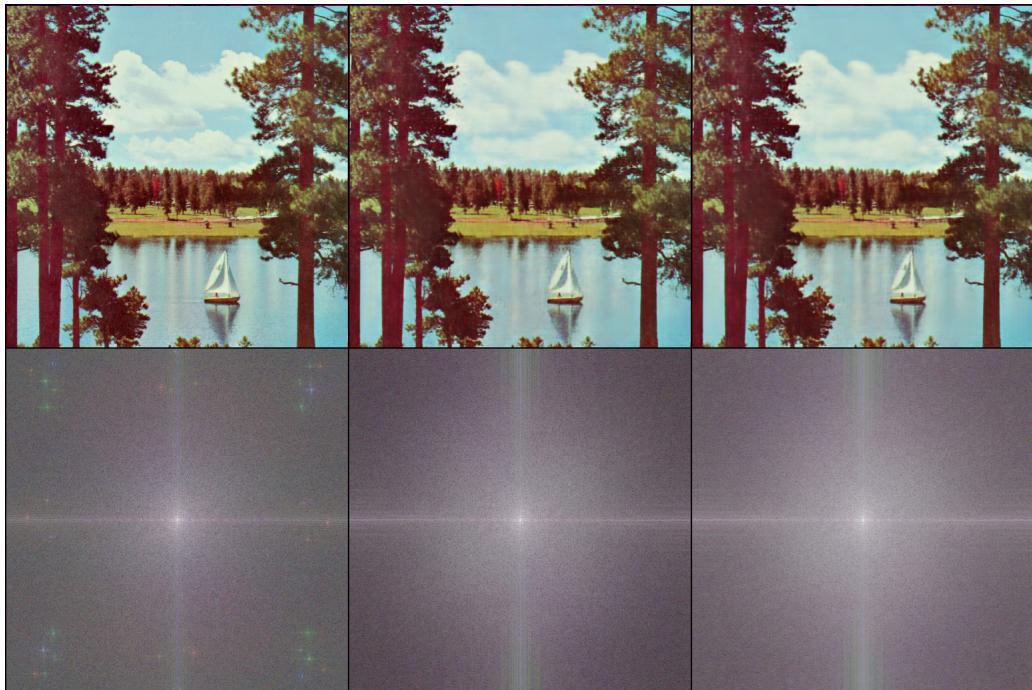


Figura 9: Mejor imagen y última imagen obtenida en la última repetición. Espectro de frecuencias asociado. La mejor iteración obtenida en esa repetición fue la 1195. El valor mejor PSNR asociado fue de 27.496