



Implementación de estrategias de reconstrucción de imágenes esparsas mediante sensado compresivo: algoritmos de optimización y redes neuronales

Implementation of sparse image reconstruction strategies using compressive sensing: optimization algorithms and neural networks

Irianys Murgas¹, Alberto Figueroa¹, Maytee Zambrano^{1,2}, Fernando Arias^{1,2}, Edson Galagarza¹, Diego Bouche¹

¹ Grupo de Investigación en Tecnologías Avanzadas en Telecomunicaciones y Procesamiento de Señales (GITTS), Universidad Tecnológica de Panamá, Panamá; ² Centro de Estudios Multidisciplinarios en Ciencias, Ingeniería y Tecnología AIP (CEMCIT-AIP), Panamá

*Autor de correspondencia: maytee.zambrano@utp.ac.pa

RESUMEN. La teoría de compressive sensing (CS) permite recuperar señales a una tasa de muestreo inferior a la requerida por el teorema de Nyquist-Shannon. Gracias a esta teoría se conciben los sistemas de single pixel imaging (SPI), los cuales permiten la adquisición de imágenes 2D con sensores unidimensionales utilizando un conjunto de muestras de luz a través de aperturas codificadas y el algoritmo de optimización CS apropiado. Este trabajo describe el procedimiento a seguir para la reconstrucción de imágenes de un sistema CS-SPI, utilizando algoritmos de optimización convencionales y enfoques basados en redes neuronales, mostrando sus desempeños para diferentes matrices de muestreo, usando el lenguaje de programación Python y la imagen de prueba, cameraman, la cual se ajustó a las entradas requeridas por cada método. Se presentan cinco algoritmos convencionales: OMP, CoSaMP, IRLS, Lasso y BP, así como cinco modelos basados en redes neuronales: DR2, HSCNN-R, AMP-Net, Recon-Net e IstaxRecon. Se muestra el desempeño de estos algoritmos con el fin de orientar la selección del método según las necesidades de cada aplicación en CS-SPI y las cualidades que caracterizan a cada técnica. Los resultados mostraron que para algoritmos convencionales LASSO presentó un buen rendimiento por tiempo de ejecución y en el caso de los modelos basados en redes neuronales la arquitectura HSCNN-R e IstaxRecon mostraron alto desempeño en diferentes tasas de muestreo.

Palabras clave. Algoritmos convencionales, imagenología de un solo píxel, sensado compresivo, redes neuronales.

ABSTRACT. Compressive sensing (CS) theory allows for the recovery of signals at a sampling rate lower than that required by the Nyquist-Shannon theorem. This theory has led to the development of single-pixel imaging (SPI) systems, which enable the acquisition of 2D images with one-dimensional sensors using a set of light samples through coded apertures and the appropriate CS optimization algorithm. This work describes the procedure for reconstructing images from a CS-SPI system using conventional optimization algorithms and neural network-based approaches. Their performance is demonstrated for different sampling matrices using the Python programming language and the test image, cameraman, which was adjusted to the inputs required by each method. Five conventional algorithms are presented: OMP, CoSaMP, IRLS, Lasso, and BP, as well as five neural network-based models: DR2, HSCNN-R, AMP-Net, Recon-Net, and IstaxRecon. The performance of these algorithms is presented to guide the selection of the method according to the needs of each CS-SPI application and the characteristics of each technique. The results showed that for conventional algorithms, LASSO presented good performance in terms of runtime, and in the case of models based on neural networks, the HSCNN-R and IstaxRecon architectures showed high performance at different sampling rates.

Keywords. Conventional algorithms, single pixel imaging, compressive sensing, neural networks.

1. Introducción

Citación: I. Murgas, A. Figueroa, M. Zambrano, F. Arias, E. Galagarza y D. Bouche, "Implementación de estrategias de reconstrucción de imágenes esparsas mediante sensado compresivo: algoritmos de optimización y redes neuronales", *Revista de I+D Tecnológico*, vol. 21, no. 1, pp. (0), 2026.

Tipo de artículo: Original. **Recibido:** 17 de enero de 2026. **Recibido con correcciones:** 16 de marzo de 2026. **Aceptado:** 16 de marzo de 2026.

DOI.

Copyright: 2026 I. Murgas, A. Figueroa, M. Zambrano, F. Arias, E. Galagarza y D. Bouche. This is an open access article under the CC BY-NC-SA 4.0 license (<https://creativecommons.org/licenses/by-nc-sa/4.0/>).

El concepto de imagenología de un solo píxel (SPI), surgió después del desarrollo de la teoría de CS. Esta teoría fue propuesta por [1] y [2], la cual sostiene que es posible recuperar una señal a partir de la señal submuestreada cuando dicha señal original objetivo es esparsa o está esparcida en el dominio de alguna transformación lineal [3]. Esto rompe con la limitación del teorema de Nyquist-Shannon en la adquisición de datos, al reducir la frecuencia de muestreo más allá del límite de este teorema [3].

La técnica de SPI fue descrita por [4] y se basa en el principio de modulación espacial de luz, en el cual se carga una secuencia de patrones estructurados en algún dispositivo de modulación de luz que se utiliza para redireccionar la luz emitida por un objeto a fotografiar iluminado externamente. Esta luz modulada se detecta mediante un sistema de lentes recolectores en un único fotodetector que genera una señal de voltaje de salida equivalente a la cantidad de luz recolectada [5]. Este sistema permite adquirir imágenes al muestrear una escena a una tasa menor a la requerida por el teorema de Nyquist-Shannon, al aplicar algún algoritmo de optimización sobre el conjunto de mediciones junto a los patrones de modulación luz seleccionados. Dichos algoritmos de reconstrucción se agrupan en algoritmos convencionales y en modelos de redes neuronales.

Algunas de las ventajas de un sistema SPI incluyen su capacidad para operar en entornos con poca luz y la adquisición de imágenes con un amplio rango de longitud de onda [6].

Las primeras investigaciones que combinaron CS y SPI utilizaron distintos tipos de algoritmos para reconstruir la imagen, los cuales, se pueden dividir en: algoritmos de optimización convexa, algoritmos tipo greedy, algoritmos de optimización no convexa, algoritmos de minimización de la distancia de Bregman y algoritmos de minimización de variación total, como se muestran y comparan en la publicación de [3]. Se ha demostrado la utilidad de estos algoritmos en diversos estudios, sin embargo, existen problemas relacionados con su uso, como el tiempo de reconstrucción y la calidad de la imagen resultante dependiente de una alta tasa de muestreo [7]. A pesar de los avances en SPI, estos sistemas presentan dificultades en aplicaciones reales [8]. Debido a ello, los trabajos más recientes se enfocan en técnicas de reconstrucción basadas en neural network (NN), especialmente, convolutional neural network

(CNN). Las CNN aprovechan el desarrollo de la velocidad de los cálculos realizados por las graphics processing units (GPU) para permitir velocidades de cálculo más altas que las disponibles en los procesadores convencionales [9].

Uno de los ecosistemas de acceso abierto más reciente para procesamiento de datos adquiridos por sistemas CS-SPI mediante redes neuronales es OpenSpyrit [10], que provee las herramientas y bases de datos con mediciones de acceso abierto para completar todo el proceso de reconstrucción de manera reproducible y comparar los resultados de diferentes algoritmos. Otros ejemplos de aplicación de algoritmos de CS basados en redes neuronales se encuentran en [11], [12] y [13].

En comparación con los algoritmos iterativos los algoritmos basados en Deep learning (DL) pueden alcanzar reconstrucciones más rápidas, así como una alta calidad [14], [15]. Sin embargo, cabe destacar que los métodos basados en DL requieren una gran cantidad de datos de pares de entrada y salida para entrenar la red neuronal [16].

Cada aplicación de SPI determina el tipo de algoritmo que se necesite utilizar. Ambos métodos, el convencional y aquel basado en redes neuronales, producen resultados favorables dependiendo de su aplicación, por lo que no se descarta el uso de uno sobre el otro método y por lo tanto se muestran ambos métodos en el trabajo. A pesar de que los algoritmos basados en redes neuronales son más rápidos y precisos que los convencionales, se pueden utilizar ambas técnicas [15], [17].

En este contexto, el presente artículo propone una guía reproducible para la reconstrucción de imágenes en un sistema CS-SP. Además, se muestra una evaluación de los algoritmos dentro de cada familia de métodos, algoritmos convencionales y algoritmos basados en redes neuronales, con el fin de orientar la selección del método más conveniente según las necesidades de la aplicación del lector. En particular, los algoritmos convencionales se evalúan utilizando PSNR, SSIM, tiempo de reconstrucción y error relativo, tal como en [3], mientras que cinco modelos basados en redes neuronales se evalúan mediante PSNR y SSIM.

Este trabajo se estructura de la siguiente forma: primero se presenta la explicación de los fundamentos teóricos de CS, los algoritmos de optimización convencionales utilizados y la estructura de cinco redes neuronales para reconstruir imágenes. Finalmente se

presentan los resultados de la reconstrucción de estos algoritmos utilizando diferentes matrices y tasas de muestreo.

2. Marco Teórico

Se explican los fundamentos de CS, así como de los algoritmos convencionales y enfoques basados en redes neuronales. Adicionalmente, se explica la teoría de las matrices de muestreo utilizadas para generar las muestras experimentales. Para todo el procesamiento de las mediciones, reconstrucciones y comparaciones, se utilizó el lenguaje de programación Python y repositorios de GitHub de código abierto.

2.1 Teoría de sensado compresivo

Sea $x \in \mathbb{R}^N$ la señal original, representada como un vector columna de dimensión $N \times 1$. Considérese $\{\psi_i\}_{i=1}^N$ un conjunto de N vectores de base ortonormal de \mathbb{R}^N , y $\Psi \in \mathbb{R}^{N \times N}$ la matriz cuyas columnas corresponden a los vectores bases ψ_i . De esta forma la señal x puede representarse como una combinación lineal de dichos vectores bases [18]:

$$x = \sum_{i=1}^N s_i \psi_i \text{ o } x = \Psi s, \quad (1)$$

en esta expresión, $s \in \mathbb{R}^N$ es el vector de coeficientes, donde cada componente es el productor interno $s_i = \langle x, \psi_i \rangle$. Los vectores x y s constituyen una representación equivalente de la misma señal, en los dominios de tiempo/espacio y de la base Ψ , respectivamente [18].

La adquisición de las mediciones se realiza mediante una matriz de medición $\Phi \in \mathbb{R}^{M \times N}$, con $M \ll N$, y donde la matriz tiene columnas ϕ_i para cada $1 \leq i \leq M$. El vector de mediciones $y \in \mathbb{R}^M$, que corresponde al productor interno $y_i = \langle \phi_i, x \rangle$, se obtiene como

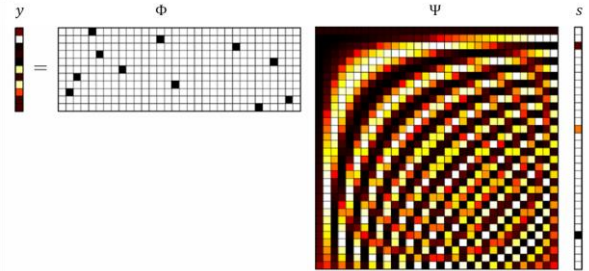
$$y = \Phi x. \quad (2)$$

Al sustituir la representación de x dada en (1) dentro de la ecuación (2), se obtiene la relación entre las mediciones y los coeficientes esparcos s ,

$$y = \Phi x = \Phi \Psi s = \theta s, \quad (3)$$

en esta notación, $\theta = \Phi \Psi \in \mathbb{R}^{M \times N}$ es la matriz de sistema efectiva que combina la matriz de medición Φ con la base de esparcidad Ψ .

El proceso de CS se ejemplifica en la Figura 1 y el objetivo de CS es que a partir de $M \ll N$ mediciones se puede reconstruir la señal original x [18].



Para la reconstrucción de la señal x sea efectiva se necesitan que se cumplan algunas características en las matrices de la ecuación. El vector s debe ser esparso con respecto a la base Ψ , lo que significa que es una combinación lineal de solamente $k \ll N$ vectores bases y que k de sus N coeficientes son significativamente distintos de cero.

La otra característica es diseñar una matriz de medición Φ que sea incoherente con respecto a la base Ψ sobre la cual la señal x es esparsa. Esto significa que las filas de Φ no están correlacionadas con las columnas de Ψ [19]. Típicamente estas matrices consisten en proyecciones aleatorias.

Existen matrices genéricas que son suficientemente incoherentes con respecto a una base Ψ [20], [19]. Entre ellas destacan las matrices de medición de Bernoulli y Gaussiana, las cuales satisfacen la propiedad de isometría restringida (RIP) para bases Ψ genéricas con alta probabilidad. Entre las bases Ψ más utilizadas están la de Fourier y la de Wavelets.

Existen matrices genéricas que son suficientemente incoherentes con respecto a una base Ψ [20], [19]. Entre ellas destacan las matrices de medición de Bernoulli y Gaussiana, las cuales satisfacen la propiedad de isometría restringida (RIP) para bases Ψ genéricas con alta probabilidad. Entre las bases Ψ más utilizadas están la de Fourier y la de Wavelets.

2.2 Matrices de muestreo

La matriz de medición tiene dimensiones $M \times N$, donde M corresponde al número de mediciones que se tomarán de la imagen, y N es la cantidad de pixeles de la imagen, tal que $M < N$.

Se utilizaron tres tipos de matrices, las cuales se mencionan por [3], las cuales son: matriz de medición aleatoria Gaussiana, matriz de medición aleatoria de Bernoulli y la matriz de Hadamard.

Cada una de las matrices empleadas se construyeron con un tamaño de $N \times N = 4096 \times 4096$. Después estas matrices se almacenaron para asegurar repetibilidad en cada prueba. En cada prueba se recortaron las matrices dependiendo de la cantidad de M mediciones que se desean de la imagen.

2.2.1 Matriz de medición aleatoria Gaussiana

Constituye una distribución gaussiana donde los elementos son independientes y obedecen una media de cero y una varianza de $1/M$ [3]. Se utilizó una matriz gaussiana con media de cero y varianza de uno.

2.2.2 Matriz de medición aleatoria de Bernoulli

Posee elementos que son independientes y están uniformemente distribuidos, los cuales obedecen una distribución binomial de Bernoulli con una probabilidad de un medio [19]. Sus valores únicamente son cero y uno.

2.2.3 Matriz de Hadamard

La matriz de Hadamard se construye a partir de una potencia múltiplo de dos. Sus valores son -1 y 1, y sus columnas son ortogonales [3]. La matriz utilizada en CS generalmente es un recorte de M -filas de la matriz de Hadamard de tamaño $N \times N$.

2.3 Diccionario o Base de esparcidad

Las señales o imágenes que se van a reconstruir deben tener una representación esparsa en alguna base o diccionario Ψ como la transformada Discreta de Coseno (DCT) o la Transformada de Wavelet (WT).

Durante las pruebas se empleó la base DCT, la cual consiste en una transformada ortogonal que descompone una imagen en su espectro espacial de frecuencia.

2.4 Algoritmos Convencionales

Durante las pruebas se utilizaron cinco algoritmos, los cuales son: Orthogonal Matching Pursuit (OMP), Lasso Regression, Basis Pursuit, Compressive Sampling Matching Pursuit (CoSaMP) e Iterative Rewighted Least Squares (IRLS).

2.4.1 Orthogonal Matching Pursuit (OMP)

Es un algoritmo iterativo de CS, una mejora del algoritmo Matching Pursuit. En cada iteración se elige del diccionario un elemento que mejor de aproxime al residuo.

En lugar de simplemente tomar el producto escalar del residuo y el nuevo elemento del diccionario para obtener el peso del coeficiente, se ajusta la función a todos los elementos del diccionario ya seleccionados mediante mínimos cuadrados, o se proyecta la función ortogonalmente en todos los átomos del diccionario seleccionado, de esta forma no selecciona repetidamente el mismo elemento [18].

OMP encuentra el valor absoluto más alto del producto interno entre cada columna y el residuo r , localiza la

columna de θ con la mayor correlación con el residuo [3], tal como se representa en la ecuación (4).

$$r = y - \theta \hat{s}. \quad (4)$$

El algoritmo está disponible a través de la función `OrthogonalMatchingPursuit` de la librería `sklearn.linear_model` la cual se describe en [21].

2.4.2 Lasso Regression

Least Absolute Shrinkage and Selection Operator (LASSO) es una técnica de minimización de la norma l_1 que equilibra la complejidad del modelo con la capacidad descriptiva [19]. Es un algoritmo de optimización convexa.

Dado un conjunto de mediciones $y \in \mathbb{R}^M$ y una matriz de medición $\Phi \in \mathbb{R}^{M \times N}$, la regresión busca encontrar la relación entre las columnas de Φ que sea más coherente con los resultados de y . La regresión con mínimos cuadrados tenderá a dar como resultado un vector x que tenga coeficientes distintos de cero para todas las entradas, lo que indica que se deben usar todas las columnas de Φ para describir a y . LASSO agrega un término de penalización de 1 para regularizar el problema de regresión de mínimos cuadrados, es decir, para evitar el sobreajuste [19].

Su representación matemática se adjunta en la ecuación (5).

$$x = \operatorname{argmin}_{x'} \|\Phi x' - y\|_2 + \lambda \|x\|_1, \quad (5)$$

donde λ es un parámetro de regularización. La función Lasso de la librería `sklearn.linear_model` permitió aplicar el algoritmo de reconstrucción [21].

2.4.3 Basis Pursuit

Es una técnica de procesamiento que tiene como objetivo descomponer una señal en una superposición de elementos del diccionario que tienen la menor norma l_0 de los coeficientes, sujeta a la restricción de igualdad de la ecuación (6), tal como se describe en [3].

$$\min_{\hat{s} \in \mathbb{R}^N} \|\hat{s}\|_{l_1} \text{ sujeta a } y = \theta s, \quad (6)$$

El algoritmo se aplica con la librería `cvxpy` [22].

2.4.4 Compressive Sampling Matching Pursuit (CoSaMP)

Es un algoritmo iterativo de reconstrucción basado en OMP. CoSaMP aplica un umbral estricto (Hard Thresholding) al seleccionar las M entradas más grandes del vector obtenido, tras aplicar la pseudo inversa a y .

Las columnas de Ψ , seleccionadas para la pseudo inversa, son obtenidas al aplicar umbral estricto de magnitud $2M$ a Ψ^* aplicado al residuo de la iteración previa y añadiendo estos índices al conjunto de la iteración previa, donde Ψ^* es la conjugada compleja de Ψ . En CoSaMP un factor importante es que se utiliza la pseudo inversa en cada iteración. [23].

Se utilizó el repositorio de GitHub encontrado en [24].

2.4.5 Iterative Reweighted Least Squares (IRLS)

Es un método de optimización no convexo. Este tipo de algoritmos recuperan la señal a través de pequeñas mediciones al reemplazar la norma l_1 con la norma l_p donde $p \leq 1$. El equivalente de la ecuación $\min_{\hat{s} \in \mathbb{R}^N} \|\hat{s}\|_{l_1}$ sujeta a $y = \theta s$ es la ecuación (7) mostrada a continuación.

$$\min_{\hat{z} \in \mathbb{R}^N} \|\hat{s}\|_{l_p} \text{ sujeta a } \theta s = y, \quad (7)$$

donde $p < 1$. IRLS consiste en reemplazar la función objeto l_p con una l_2 ponderada.

$$\min_s \sum_{i=1}^N w_i s_i^2 \text{ sujeta a } \theta s = y, \quad (8)$$

donde los pesos son actualizan con la iteración previa $s^{(N-1)}$ conde (2) es una aproximación de primer orden $w_i = |s_i^{(N-1)}|^{p-2}$. La solución de (8) puede obtenerse explícitamente a través de la siguiente iteración de $s^{(N)}$ [3].

Se utilizó el repositorio de GitHub encontrado en [25].

2.5 Algoritmos basados en redes neuronales

Durante las pruebas se utilizaron cinco algoritmos basados en redes neuronales: Recon-Net, DR2, HSCNN-R, AMP-Net e ISTAxRECON.

2.5.1 Recon-Net

Esta arquitectura, descrita en [11], cuenta con una capa densa al inicio y una serie de capas convolucionales en serie. La primera y cuarta capa convolucional utiliza “kernels” de tamaño 11 y genera 64 mapas de características. La segunda y quinta capa convolucional utiliza “kernels” de tamaño 1 y genera 32 mapas de características. La tercera y sexta capa convolucional utiliza “kernels” de tamaño 7 y genera 1 mapa de características.

2.5.2 DR2

Esta arquitectura, descrita en [26], está diseñada a manera de bloques. Cada bloque cuenta con tres capas convolucionales. La primera capa convolucional utiliza “kernels” de tamaño 11 y genera 64 mapas de características. La segunda convolucional utiliza “kernels” de tamaño 1 y genera 32 mapas de características. La tercera convolucional utiliza “kernels” de tamaño 7 y genera 1 mapa de características. Entre cada capa convolucional de un bloque se añade una capa de “Batch Normalization”. Esta arquitectura cuenta con cuatro bloques.

2.5.3 HSCNN-R

Esta arquitectura, descrita en [27], está diseñada a manera de bloques. Todos los bloques son exactamente iguales, pero cuenta con una capa convolucional previa al primer bloque y una serie de capas convolucionales luego del último bloque. A la capa convolucional a la entrada se le llamará bloque de entrada y a la serie de capas convolucionales en la salida, se le llamará bloque de salida. La capa convolucional en el bloque de entrada utiliza “kernels” de tamaño 3 y genera 64 mapas de características sin ReLu. Los bloques respectivos cuentan con 2 capas convolucionales, una de adición y una capa de ReLu a la salida del bloque. El bloque de salida cuenta con una capa de adición, un ReLu y dos capas convolucionales. La capa de adición sumará la salida del bloque de entrada con la salida del último bloque repetitivo.

2.5.4 AMP-Net

Esta arquitectura, descrita en [28], está diseñada a manera de bloques. Cada bloque cuenta con 6 capas convolucionales, una capa de suma, tres capas de Batch Normalization y 3 capas de ReLu. Todas las capas convolucionales a excepción de la última de cada bloque utilizan “kernels” de tamaño 3, genera 32 mapas de características. La última capa convolucional utiliza “kernels” de tamaño 3, genera 1 mapa de característica. El Batch Normalization y ReLu se encuentran entre, la segunda y la tercera, la tercera y la cuarta, la cuarta y la quinta capa convolucional. Al final de cada bloque se encuentra la capa de adición, en ella se suman la salida del bloque previo con la salida de la última capa convolucional del bloque. Esta arquitectura cuenta con 9 bloques.

2.5.5 ISTAxRECON

Corresponde a un modelo basado en dos Sub-Redes para el proceso de reconstrucción. La primera

corresponde a la Sub-Red de Mapeo Lineal, que tiene el propósito de generar una reconstrucción inicial de la imagen a través de las mediciones obtenidas al aplicar CS.

La siguiente es la Sub-Red de Aprendizaje Residual, que tiene el propósito de aproximarse a una óptima reconstrucción final, partiendo del resultado de la Sub-Red de Mapeo Lineal.

El modelo está compuesto por 7 bloques, donde cada bloque consiste en 6 capas convolucionales, un soft thresholding en el medio de estas y una capa de adición al final de cada bloque.

Las capas convolucionales pueden extraer varias características de diferentes regiones locales de los datos de entrada, se componen de una colección de núcleos de convolución, donde cada neurona actúa como un núcleo [29].

2.6 Métricas de evaluación

Para mostrar el desempeño de los algoritmos de reconstrucción se emplean métricas que cuantifican la fidelidad de la imagen reconstruida como el costo computacional del método. En particular, se utilizan el Peak signal-to-noise ratio (PSNR) y el Structural similarity index measure (SSIM) como medidas de calidad. Adicionalmente, se utiliza el tiempo de reconstrucción como indicador de viabilidad práctica, y el error relativo como medida global de discrepancia entre la reconstrucción y la imagen objetivo.

2.6.1 Peak signal-to-noise-ratio PSNRs

Se utiliza comúnmente para medir la calidad de imágenes restauradas. Compara la señal original con la restaurada al considerar el error cuadrático entre los valores de sus píxeles [30].

Sea x la imagen original y \hat{x} la imagen reconstruida, ambas con N píxeles, el mean square error (MSE) se define en la siguiente ecuación:

$$MSE = \frac{1}{N} \sum_{i=1}^N (x_i - \hat{x}_i)^2, \quad (9)$$

donde x_i y \hat{x}_i representan la intensidad del píxel i en la imagen original y reconstruida, respectivamente. A partir de MSE se calcula el PSNR como:

$$PSNR = 10 * \log_{10} \left(\frac{MAX^2}{MSE} \right), \quad (10)$$

donde MAX denota el valor máximo posible de intensidad en la imagen. Valores altos de PSNR indican mejor calidad de reconstrucción.

2.6.2 Structural similarity index measure SSIM

SSIM evalúa la similitud entre las imágenes originales x y restauradas \hat{x} en términos de luminancia, contraste y estructura. Valores de SSIM más altos indican una mejor calidad de restauración [30]. La métrica se expresa en la ecuación (10)

$$SSIM_{(x,\hat{x})} = (2 * \mu_x * \mu_{\hat{x}} + c_1) * (2 * \sigma_{x\hat{x}} + c_2) / (\mu_x^2 * \mu_{\hat{x}}^2 + c_1) * (\sigma_x^2 + \sigma_{\hat{x}}^2 + c_2), \quad (11)$$

donde μ_x y $\mu_{\hat{x}}$ son los valores medios de las imágenes originales y restauradas, σ_x^2 y $\sigma_{\hat{x}}^2$ son la variancia de la imagen original y restaurada. $\sigma_{x\hat{x}}$ es la covariancia entre la imagen original y la restaurada y c_1 y c_2 son constantes que evitan la división por cero.

2.6.3 Tiempo de reconstrucción

Se refiere al tiempo de computación que tarda la reconstrucción de la imagen, a través de las muestras y matrices de medición. Se mide a través de la librería time.

2.6.4 Error Relativo

Se refiere a la relación entre el error absoluto y el valor exacto, se expresa en la ecuación (12). Se utiliza la imagen real y la reconstrucción.

$$Error\ Relativo = \left| \frac{x - \hat{x}}{x} \right| * 100\% \quad (12)$$

2.7 Procedimiento seguido en los algoritmos convencionales.

La imagen de prueba seleccionada es ‘‘Cameraman’’ la cual es una imagen con dimensiones originales de 100x100 píxeles, común entre las evaluaciones de CS y se muestra en la Figura 2.



Figura 2. Imagen de prueba

La imagen se redimensiona a 64x64 píxeles de manera que se pueda agilizar el proceso de reconstrucción.

Una vez creadas las matrices de muestreo, redimensionada la imagen y definidos los algoritmos de reconstrucción, se puede proceder con la generación de las muestras experimentales.

La imagen de prueba se muestrea con las matrices de medición de Hadamard, Gaussiana o Bernoulli, de manera que se genera y . Es decir, se multiplica la imagen de prueba, que se encuentra en formato de numpy, con la respectiva matriz de muestreo, de manera que se obtienen las mediciones comprimidas.

Con los parámetros, y , Φ y Ψ se puede recuperar la imagen original, aplicando alguno de los cinco algoritmos de reconstrucción propuestos.

Como métricas de evaluación se utilizó el PNSR, SSIM, tiempo de reconstrucción y error relativo de la imagen reconstruida.

2.8 Procedimiento seguido en los algoritmos basados en redes neuronales.

A cada arquitectura utilizada se le anexó un reshape para mantener la consistencia entre las pruebas con cada algoritmo. Se utilizó misma imagen de los algoritmos convencionales. Las métricas de evaluación utilizadas fueron el PSNR y SSIM.

3. Resultados y discusión

Se presentan los resultados del desempeño de los algoritmos convencionales y algoritmos basados en redes neuronales, con respecto a la matriz de medición utilizada y distintas tasas de muestreo.

3.1 Desempeño de los algoritmos convencionales para diferentes matrices de muestreo

Se presentan algunas gráficas comparativas con las cualidades de los algoritmos convencionales.

En el caso de la matriz de Bernoulli los mejores resultados, como se presenta en la Tabla 1, respecto a PSNR, SSIM y error relativo lo presenta el algoritmo BP, y para el tiempo de reconstrucción, el algoritmo OMP. Los algoritmos IRLS y BP, presentaron los tiempos de reconstrucción más elevados, tal cual muestra la Figura 3, tardando alrededor de diez minutos para reconstrucciones con una tasa de muestreo de cuatro mediciones por cada cinco píxeles, en comparación al resto que toman segundos. Lasso es el algoritmo óptimo en resultados de PSNR como se aprecia en la Figura 4, tomando en cuanto el tiempo que tarda el algoritmo. OMP y CoSaMP presentan tiempos cortos para la reconstrucción, sin embargo, sus resultados de calidad son los menos efectivos.

Respecto a la matriz Gaussiana, como se observa en la Tabla 1, los mejores resultados de PSNR, SSIM y tiempo de reconstrucción lo presentan el algoritmo IRLS, el error

relativo óptimo lo presenta CoSaMP. El tiempo de reconstrucción de BP e IRLS sigue siendo grande respecto al resto de algoritmos, como se aprecia en la Figura 5. En este caso, IRLS aumenta su tiempo de reconstrucción respecto a la matriz de Bernoulli. En OMP se aprecia un error relativo excesivamente alto a una tasa de una medición por cada diez píxeles con la matriz Gaussiana, estabilizándose a tasa de una medición por cada cinco píxeles tal cual muestra la Figura 6.

Para la matriz de Hadamard, Lasso presenta valores óptimos respecto a los demás algoritmos en todos los parámetros como se aprecia en la Tabla 1. Además, IRLS y BP siguen demorando más tiempo para reconstrucciones de tasas superiores, no obstante, entre las tres matrices de medición, es con Hadamard donde tardan menor tiempo como se observa en la Figura 7. Cabe destacar que esta matriz de Hadamard desestabiliza el error relativo de todos los algoritmos, debido a que este valor se elevó a tasas de muestreo de una medición por cada diez píxeles hasta una tasa de tres mediciones por cada diez píxeles, por lo que es importante notar en qué rangos la matriz de Hadamard mejora la reconstrucción.

Los algoritmos IRLS, Lasso y BP presentan las imágenes con mayor parecido a la imagen original para la matriz Bernoulli y Hadamard, como se presenta en la Figura 8. Es importante destacar qué aplicación se desea emplear en los diferentes algoritmos de reconstrucción para decidir cuál es el óptimo. Si desean resultados rápidos sin mucha precisión visual, OMP y CoSaMP son aptos. Si se desea calidad en el resultado, sin procurar la rapidez del algoritmo, IRLS y BP son una buena opción. No obstante, preservando ambas características, Lasso es la herramienta óptima para realizar reconstrucciones de imágenes. Este comportamiento se debe a que la penalización l_1 favorece soluciones esparsas frente al ruido.

Los resultados presentados son similares a los mostrados en [3]. Podemos destacar que la matriz de Bernoulli y Hadamard presentaron mejores resultados que la matriz Gaussiana respecto a los parámetros evaluados en la Tabla 1, y se añaden los resultados visuales de esas matrices en la Figura 8.

Tabla 1. Mejores escenarios de los algoritmos convencionales

Mejores Escenarios						
Parámetros	Bernoulli		Gaussiana		Hadamard	
	Valor	Algoritmo	Valor	Algoritmo	Valor	Algoritmo
Error Relativo	6.7943	BP	6.8513	IRLS	6.6665	LASSO

Tiempo de reconstrucción (s)	0.2247	OMP	0.2655	CoSaMP	0.3655	LASSO
PSNR (dB)	29.0886	BP	29.0161	IRLS	29.2535	LASSO
SSIM	0.7877	BP	0.7845	IRLS	0.8007	LASSO

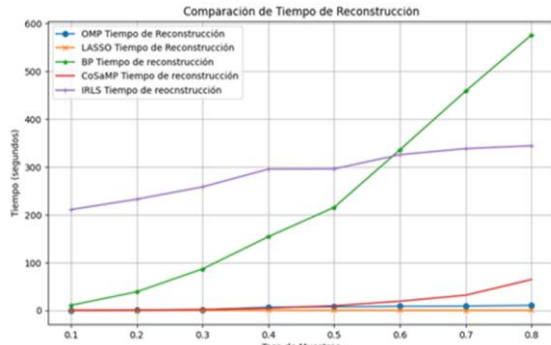


Figura 3. Tiempo de reconstrucción de los algoritmos convencionales con la matriz Bernoulli

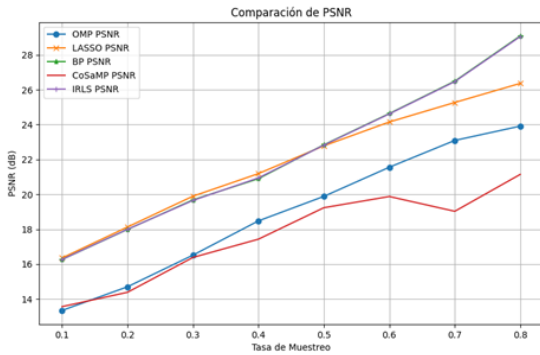


Figura 4. PSNR de los algoritmos convencionales con la matriz Bernoulli

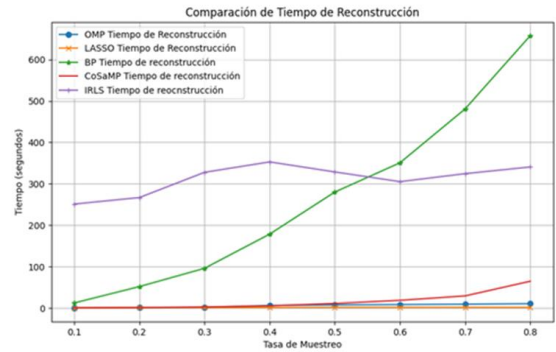


Figura 5. Tiempo de reconstrucción de los algoritmos convencionales con la matriz Gaussiana

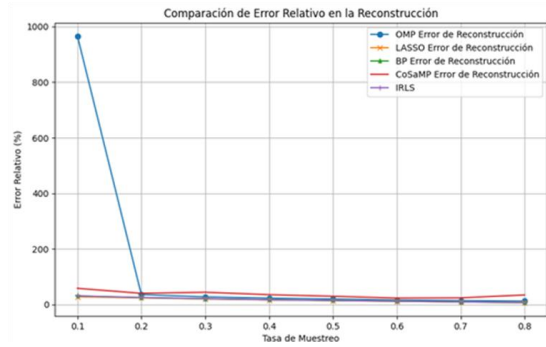


Figura 6. Error relativo de los algoritmos convencionales con la matriz Gaussiana

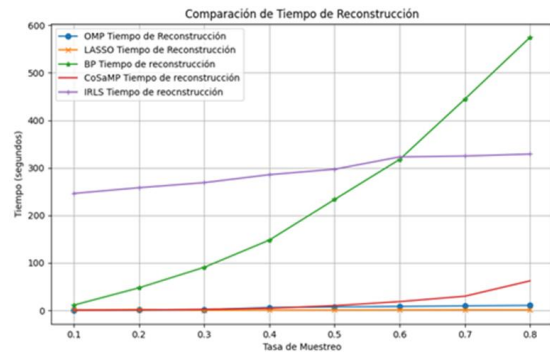


Figura 7. Tiempo de reconstrucción de los algoritmos convencionales con la matriz Hadamard

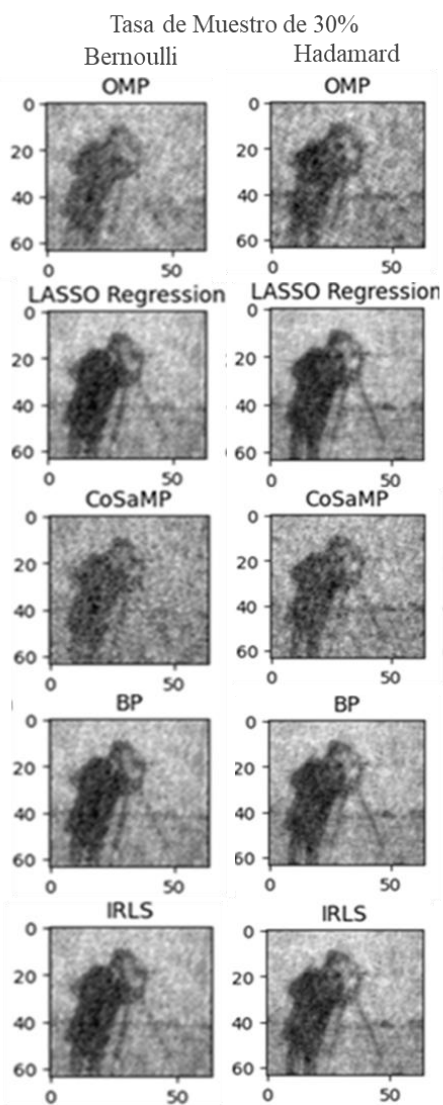


Figura 8. Reconstrucciones de los algoritmos convencionales para la matriz Bernoulli y Hadamard.

3.2 Comparación de los algoritmos basados en redes neuronales

Como se aprecia en la Tabla 2, para una tasa de una muestra por cada dos píxeles, la matriz gaussiana presenta mejores resultados con respecto a la matriz de Bernoulli, considerando los resultados de HSCNN-R, AMP-Net e IStaxRecon, que corresponden a los algoritmos con valores más altos. IStaxRecon obtuvo los segundos mejores resultados con respecto al PSNR y SSIM evaluando ambas matrices. El modelo HSCNN-R obtuvo los mejores resultados en cuanto a PSNR y SSIM con ambas matrices, exceptuando el valor de PSNR del modelo AMP-Net, que fue superior en la matriz Gaussiana.

En el caso de una tasa de una muestra por cada diez píxeles, tal cual se muestra en la Tabla 3, la matriz gaussiana sigue presentando mejores resultados que la matriz de Bernoulli en ambos parámetros de PSNR y SSIM. Los modelos con resultados óptimos siguen siendo HSCNN-R, AMP-Net e IStaxRecon. El modelo HSCNN-R presentó los valores más grandes respecto al SSIM con ambas matrices de medición, por su parte IStaxRecon presentó los resultados más altos en cuanto al PSNR con ambas matrices.

Las pruebas para una tasa de una muestra por cada cien píxeles mostraron resultados diferentes a los casos anteriores como se aprecia en la Tabla 4. La matriz de Bernoulli superó a la matriz Gaussiana para los parámetros de PSNR y SSIM. Los modelos DR2, HSCNN-R, AMP-Net e IStaxRecon destacaron por sus resultados para cada parámetro de evaluación y matriz de medición. El modelo IStaxRecon superó a los demás modelos con un valor más alto de PSNR para la matriz de Bernoulli. Por su parte, el modelo HSCNN-R, con la matriz de Bernoulli, destacó con el parámetro de SSIM respecto a los demás modelos. Con la matriz gaussiana, el mayor valor de PSNR obtenido fue de DR2 y el mayor valor de SSIM fue de HSCNN-R.

Podemos resaltar que los modelos con mayor constancia en óptimos resultados son HSCNN-R e IStaxRecon, tal como se muestra en la Tabla 5, donde se agregan los algoritmos con mejores resultados sin especificar la matriz de muestreo utilizada.

En el caso de HSNCC-R, la estructura residual profunda permite refinar progresivamente la reconstrucción inicial, mientras que en el caso de IStaxRecon el esquema de dos subredes emula el comportamiento de algoritmos iterativos, que mejora la recuperación de detalles estructurales.

Tabla 2. Comparación de los algoritmos basados en redes neuronales con tasa de muestreo de 50%.

Imagen	Método	Tasa Muestreo = 0.5			
		Máscara Bernoulli		Máscara Gaussiana	
		PSNR (dB)	SSIM	PSNR (dB)	SSIM
Cameraman	X0	18.6746	0.5334	26.0536	0.7741
	Recon-Net	23.5603	0.7232	29.4490	0.8758
	DR2	24.4644	0.7639	28.6377	0.8761
	HSCNN-R	27.1030	0.8452	30.3078	0.9125
	AMP-Net	26.6799	0.8246	30.3256	0.9002
	IStaxRecon	<u>26.6889</u>	<u>0.8285</u>	<u>30.3154</u>	<u>0.9025</u>

Tabla 3. Comparación de los algoritmos basados en redes neuronales con tasa de muestreo de 10%.

Imagen	Método	Tasa Muestreo = 0.1			
		Máscara Bernoulli		Máscara Gaussiana	
		PSNR (dB)	SSIM	PSNR (dB)	SSIM
Cameraman	X0	16.1356	0.2268	21.5739	0.6158
	Recon-Net	19.7047	0.5468	22.3007	0.6731
	DR2	20.6219	0.5823	22.2029	0.6667
	HSCNN-R	21.6720	0.6825	22.1921	0.7541
	AMP-Net	<u>21.7096</u>	0.6448	<u>22.5750</u>	0.6716
	IstaxRecon	21.8382	<u>0.6650</u>	22.7111	<u>0.7187</u>

Tabla 4. Comparación de los algoritmos basados en redes neuronales con tasa de muestreo de 1%.

Imagen	Método	Tasa Muestreo = 0.01			
		Máscara Bernoulli		Máscara Gaussiana	
		PSNR (dB)	SSIM	PSNR (dB)	SSIM
Cameraman	X0	13.6396	0.1218	17.3061	0.4206
	Recon-Net	17.6150	0.4379	<u>17.3446</u>	0.4472
	DR2	17.7514	0.4718	17.3812	0.4176
	HSCNN-R	17.3928	0.5004	16.6911	0.4732
	AMP-Net	<u>17.7757</u>	0.4377	17.2623	0.4431
	IstaxRecon	18.0009	<u>0.4967</u>	17.3327	<u>0.4574</u>

Tabla 5. Mejores escenarios de las redes neuronales.

Mejores Escenarios				
Tasa de muestreo	Valor PSNR (dB)	Algoritmo	Valor SSIM	Algoritmo
1%	18.0009	IstaxRecon	0.5004	HSCNN-R
10%	22.7111	IstaxRecon	0.7541	HSCNN-R
50%	30.3256	AMP-Net	0.9125	HSCNN-R

4. Conclusiones

Este trabajo ofrece una guía reproducible sobre cómo realizar los procedimientos de reconstrucción de imágenes aplicando CS, enfocado en un sistema de SPI, así como se presenta el desempeño de los algoritmos de reconstrucción dentro de cada categoría, convencionales y basados en redes neuronales, bajo distintas matrices y tasas de muestreo.

Con las pruebas realizadas, se demuestra la posibilidad de utilización de los algoritmos en herramientas de código abierto, así como se evidencian las fuentes y repositorios útiles para esta tarea.

Los resultados entre los tipos de algoritmos de optimización convencionales permiten destacar el desempeño de cada algoritmo según parámetros como tasa de muestreo y matriz de medición. Destacan la matriz de Bernoulli y la matriz Hadamard como buenas herramientas para resultados más eficientes. Además, el algoritmo LASSO constituye el un óptimo recurso para reconstrucciones rápidas de imágenes con aceptables valores de PSNR y SSIM.

La adición de los algoritmos basados en redes neuronales permite evidenciar los métodos actuales de reconstrucción de imágenes que destacan por su rapidez y calidad de resultados.

Los resultados de las comparaciones de los cinco algoritmos basados en redes neuronales demuestran que HSCNN-R y IstaxRecon reconstruyen las imágenes con mayor calidad y velocidad. La matriz gaussiana destaca para tasas de muestreo superiores mientras que la matriz de Bernoulli lo hace para la tasa de muestreo más pequeña que se utilizó.

En futuros trabajos, es necesario añadir más algoritmos basados en redes neuronales, debido a su potencial en tareas de procesamiento de imágenes en un mayor rango de espectro visible como hiperespectral.

Finalmente, el estudio también constituye una referencia metodológica para investigaciones posteriores, especialmente para aquellos que recién se introducen en el amplio tema de CS-SPI.

AGRADECIMIENTOS

Este trabajo se ha realizado bajo soporte de SENACYT a través de: financiamiento de los proyectos FID18-096, FID21-207 y FID23-158; del Sistema Nacional de investigación SNI del Dr. Fernando Arias y la Dra Zambrano; e Irianys Murgas es becaria del Programa Becas de licenciatura en áreas de ingeniería y ciencias básicas para formar capacidades en semiconductores, 2024.

CONFLICTO DE INTERESES

Los autores declaran no tener conflictos de interés.

CONTRIBUCIÓN Y APROBACIÓN DE LOS AUTORES

I. M.: Conceptualización, Comparaciones de los algoritmos convencionales, análisis comparativos, escritura

M. Z. Financiamiento, Revisión, Corrección y Validación del artículo.

A. F.: Diseño y prueba de la red ISTAxRECON.

E. G.: Conceptualización, Revisión y Edición.

F. A.: Financiamiento, equipamiento, revisión de resultados y revisión.

D. B.: Apoyo experimentación, análisis y revisión

"Todos los autores afirmamos que hemos leído y aprobado la versión final de este artículo."

REFERENCIAS

- [1] D. L. Donoho, «Compressed sensing,» *IEEE Transactions on Information Theory*, vol. 52, n° 4, pp. 1289 - 1306, Abril 2006.
- [2] E. J. Candes, J. Romberg y T. Tao, «Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information,» *IEEE Transactions on Information Theory*, vol. 52, n° 2, pp. 489 - 509, Feb 2006.
- [3] W. Zhao, L. Gao, A. Zhai y D. Wang, «Comparison of Common Algorithms for Single-Pixel Imaging via Compressed Sensing,» *Sensors 2023*, vol. 23, n° 10, p. Art. no. 4678., May 2023.
- [4] M. F. Duarte, M. A. Davenport, D. Takhar, J. N. Laska, T. Sun, K. F. Kelly y R. G. Baraniuk, «Single-pixel imaging via compressive sampling,» *IEEE Signal Processing Magazine*, vol. 25, n° 2, pp. 83 - 91, Mar. 2008.
- [5] C. A. Osorio Quero, D. Durini y J. Rangel-Magdaleno, «Single-pixel imaging: An overview of different methods to be used for 3D space reconstruction in harsh environments,» *Review of Scientific Instruments*, vol. 92, n° 11, p. Art. no. 111501., Nov. 2021.
- [6] I. Hoshi, T. Shimobaba, T. Kakue y T. Ito, «Single-pixel imaging using a recurrent neural network combined with convolutional layers,» *Optics Express*, vol. 28, n° 23, pp. 34069-34078, Nov. 2020.
- [7] M. A. F. X. & M. C. Zambrano, «Comparative Analysis of Sparse Signal Reconstruction Algorithms for Compressed Sensing,» de *In Twelfth LACCEI Latin American and Caribbean Conference for Engineering and Technology (LACCEI '2014)*, 2014.
- [8] Z. Wei, J. Zhang, W. Du y Z. Wa, «Real-time Single-pixel Imaging Based on Deep Learning,» *International Journal of Emerging Technologies and Advanced Applications*, vol. 1, n° 12, p. 1-5, Ene. 2024.
- [9] G. M. Gibson , S. D. Johnson y M. J. Padgett, «Single-pixel imaging 12 years on: a review,» *Optics Express*, vol. 28, n° 19, pp. 28190-28208, Sep. 2020.
- [10] G. B. Martins, Mahieu-Williame, T. Baudier y N. Ducros, «OpenSpyrit: an ecosystem for open single-pixel hyperspectral imaging,» *Optics Express*, vol. 31, n° 10, pp. 15599-15614, May. 2023.
- [11] K. Kulkarni , S. Lohit, P. Turaga, R. Kerviche y A. Ashok, «ReconNet: Non-Iterative Reconstruction of Images from Compressively Sensed Random Measurements,» de *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, 2016.
- [12] Z. Jian y B. Ghanem, «ISTA-Net: Interpretable Optimization-Inspired Deep Network for Image Compressive Sensing,» de *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Salt Lake City, UT, USA, 2018.
- [13] Z. Zhang, Y. Liu, J. Liu, F. Wen y C. Zhu, «AMP-Net: Denoising-Based Deep Unfolding for Compressive Image Sensing,» *IEEE Transactions on Image Processing*, vol. 30, pp. 1487-1500, 2020.
- [14] K. Song, Y. Bian, K. Wu, H. Liu, S. Han, J. Li, J. Tian, C. Qin, J. Hu y L. Xiao, «Single-pixel imaging based on deep learning,» 25 Oct 2023. [En línea]. Available: <https://arxiv.org/abs/2310.16869>.
- [15] B. Z. X. L. S. Z. Y. & Z. J. Chen, «Self-supervised scalable deep compressed sensing,» *International Journal of Computer Vision*, vol. 133, n° 2, pp. 688-723, 2025.
- [16] H.-Z. L. S.-H. B. R.-B. L. a. X.-H. C. Chen-Hui Wang, «Single-Pixel Hyperspectral Imaging via an Untrained Convolutional Neural Network,» *Photonics 2023*, vol. 10, n° 2, p. Art. 224, Feb. 2023.
- [17] J. & Y. J. Zhou, «Compressive sensing in image/video compression: Sampling, coding, reconstruction, and codec optimization.,» *Information*, vol. 15, n° 2, p. 75, 2025.

- [18] G. Pope, «Compressive sensing a summary of reconstruction algorithms,» ETH Zürich, Eidgenössische Technische Hochschule, Zurich, 2009.
- [19] S. L. Bruton y J. N. Kutz, *Data Driven Science & Engineering Machine Learning, Dynamical Systems, and Control*, Cambridge, U.K: Cambridge University Press, 2017.
- [20] A. S. F. M. M. D. G. & W. P. Bandeira, «The road to deterministic matrices with the restricted isometry property,» *Journal of Fourier Analysis and Applications*, vol. 19, n° 6, pp. 1123-1149, 2013.
- [21] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot y É. Duchesnay, «Scikit-learn: Machine Learning in Python,» 2011. [En línea]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.OthogonalMatchingPursuit.html.
- [22] S. Diamond y S. Boyd, «CVXPY: A Python-Embedded Modeling Language for Convex Optimization,» *Journal of Machine Learning Research*, vol. 17, n° 83, pp. 1-5, Abr. 2016.
- [23] A. Aich y P. Palanisamy, «On application of OMP and CoSaMP algorithms for DOA estimation problem,» 28 Ene. 2018. [En línea]. Available: <https://arxiv.org/abs/1704.01515>.
- [24] cmnemoi, «Repository for the Compressive Sensing course project at CY Tech,» 17 Mar. 2024. [En línea]. Available: https://github.com/cmnemoi/cy_tech_compressive_sensing.
- [25] bernielampe1, «An academic review of compressive sensing recovery algorithms,» 3 Mar. 2024. [En línea]. Available: https://github.com/bernielampe1/cs_recovery/tree/master.
- [26] H. Yao, F. Lai, S. Zhang, Y. Zhang, Q. Tian y C. Xu, «Dr2-net: Deep residual reconstruction network for image compressive sensing,» *Neurocomputing*, vol. 359, pp. 483-493, Sep. 2019.
- [27] Z. Shi, C. Chen, Z. Xiong, D. Liu y F. Wu, «HSCNN+: Advanced CNN-Based Hyperspectral Recovery from RGB Images,» de *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2018, Salt Lake City, UT, USA.
- [28] N. Li y C. C. Zhou, «AMPA-Net: Optimization-Inspired Attention Neural Network for Deep Compressed Sensing,» de *2020 IEEE 20th International Conference on Communication Technology (ICCT)*, Nanning, China, 2020.
- [29] X. Zhao, L. Wang, HanXuming, X. Han, M. Deveci y M. Parmar, «A review of convolutional neural networks in computer vision,» *Artificial Intelligence Review*, vol. 57, p. Art. 99, Mar. 2024.
- [30] R. Arhana y P. S. E. Jeevaraj, «Deep learning models for digital image processing: a review,» *Artificial Intelligence Review*, vol. 57, p. Art. 11, Ene. 2024.