

# Reconstrucción de imágenes con técnicas de Deep Learning a partir de hologramas de Fresnel simulados computacionalmente.

Gutierrez, C. David<sup>1</sup> and Ruiz, J. Nicolay<sup>1</sup>  
 Instituto de Física de la Universidad de Antioquia

(\*Electronic mail: joseph.ruiz@udea.edu.co)

(\*Electronic mail: cdavid.gutierrez@udea.edu.co)

(Dated: 24 November 2022)

En este artículo se presenta los resultados del desempeño de un modelo de Deep Learning aplicado a la óptica. La red neuronal generativa (GAN) fue puesta en la tarea de resolver la propagación inversa de un campo óptico simulado. Se encontró que con información parcial del campo óptico la GAN produjo resultados óptimos en el proceso de reconstrucción. Esto representó una alternativa eficiente frente a los algoritmos clásicos de reconstrucción con la misma cantidad de información.

[...] en la última década, los modelos de Deep learning Neural Networks (DNN's) han exhibido un amplio campo de aplicación en el procesamiento de imágenes, que en consecuencia, permite acelerar parte o la totalidad del proceso de reconstrucción de hologramas<sup>1</sup>.

## I. INTRODUCCIÓN

Con el propósito de mejorar la resolución de imagen en microscopía electrónica, Gabor desarrolló la holografía, trabajo por el cual se le otorgó el premio nobel en 1971. Básicamente, es una técnica de dos pasos que consiste en, primero, un registro total, tanto de la amplitud como la fase del frente de onda de la luz, denominado holograma, y segundo, un proceso de reconstrucción de dicho registro<sup>2</sup>. En contraste con una fotografía, un holograma captura una perspectiva más completa de la imagen; debido al registro de fase no todos sus puntos residen en un mismo plano, por lo cual es posible mirar alrededor de objetos en primer plano y notar qué hay detrás<sup>3</sup>. Con el acceso a fuentes de luz coherente durante los 60's, la holografía ha notado numerosas aplicaciones a nivel industrial, comercial e investigación que hacen de la técnica un foco de interés en la actualidad.

Métodos numéricos basados en transformadas de Fresnel y Fourier son usados para el proceso de generación y reconstrucción de hologramas digitales simulados. La reconstrucción de hologramas requiere una reducción dimensional del campo que conlleva a la supresión de información asociada con la fase. Para recuperar esta pérdida de información, algoritmos robustos como demodulación, desenvolvimiento y calibración de fase suelen ser utilizados<sup>1,4</sup>. Estos métodos requieren mucho tiempo y demandan alta complejidad computacional para cálculos de hologramas. Adicionalmente, se resalta la limitada calidad de las imágenes reproducidas de hologramas debido al ruido moteado, aberraciones ópticas, etc<sup>1</sup>.

No obstante, en la última década, los modelos de Deep learning Neural Networks (DNN's) han exhibido un amplio campo de aplicación en el procesamiento de imágenes, que en consecuencia, permite acelerar parte o la totalidad del pro-

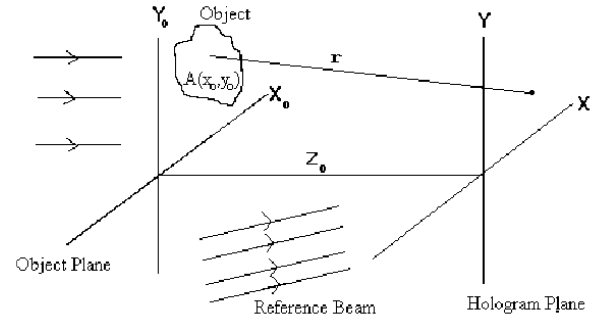


FIG. 1. Geometría en el proceso de **registro de hologramas de Fresnel**<sup>5</sup>. El objeto puede ser tratado como una apertura planar en un plano opaco que difracta una onda monocromática que ilumina desde la izquierda al plano del objeto. Las coordenadas  $(x_0, y_0)$ ,  $(x, y)$  corresponde al plano del objeto y el plano del holograma (observación), respectivamente.  $A(x_0, y_0)$  representa la amplitud del campo de entrada transmitido por el objeto.

ceso de reconstrucción de hologramas, obteniendo con éxito imágenes reproducidas con mejor calidad de imagen que los métodos convencionales<sup>1</sup>. En particular, un modelo ampliamente utilizado se basa en Conditional Generative Adversarial Network (GAN).

Este trabajo propone evaluar cuantitativa y cualitativamente el desempeño de un modelo de Deep Learning basado en Redes Neuronales Generativas como alternativa para la reconstrucción de la parte real de hologramas de Fresnel simulados computacionalmente. La generación de hologramas se encuentra restringida a imágenes binarizadas cuadradas de tamaño 256 pixels.

## II. MARCO TEÓRICO

### A. Holografía de Fresnel simulada computacionalmente

La holografía digital puede ser simulada mediante métodos basados en transformada de Fourier (*FFT*) –discreta en el caso computacional– como se mencionó en la sección I. Para la generación de hologramas de Fresnel, de acuerdo a la

teoría de difracción de Fresnel<sup>4</sup>, se debe definir una expresión para la propagación a cierta distancia del campo difractado de un objeto (propagador de Fresnel). Para ello se calcula la convolución del campo de entrada (onda transmitida por el objeto) mediante una función de transferencia en el plano de observación<sup>5</sup>. Sea  $U_0(x, y)$  un campo de entrada, de tal manera que el campo propagado a una distancia  $z$  está dado por<sup>6,7</sup>

$$U_z(x, y) = FFT^{-1}[FFT[U_0(x_0, y_0)]H_z(v, u)], \quad (1)$$

que consiste en la convolución del campo de entrada con  $H_z(v, u)$ , donde  $H_z(v, u)$  es la función de transferencia de Fresnel, definida como<sup>5,7</sup>

$$H_z(v, u) = \left(\frac{-j}{\lambda z}\right) e^{jkz} e^{-j\pi\lambda z(v^2 + u^2)}, \quad (2)$$

siendo  $\lambda$  la longitud de onda,  $k$  el número de onda y  $(v, u) = (\frac{x}{\lambda z}, \frac{y}{\lambda z})$ , coordenadas en el espacio de las frecuencias (coordenadas en el plano del holograma).

Posteriormente, se genera una onda plana de referencia  $R(x, y)$ —cuya longitud de onda  $\lambda$  es la misma usada para el campo transmitido—, que interfiere con el campo propagado en el plano de observación a un ángulo determinado. Tal campo de referencia se puede generar simplemente con

$$R(x, y) = \tilde{A}(x, y) e^{-j\pi(\cos(\alpha)x + \cos(\beta)y)}, \quad (3)$$

donde  $\alpha, \beta$  son los ángulos que forma el vector de propagación de la onda plana respecto a la normal al plano del holograma, y  $\tilde{A}(x, y)$  la amplitud compleja del campo<sup>5</sup>.

El registro total (tanto amplitud como la fase) de la intensidad resultado de adicionar el campo propagado [Ec.1] y la onda de referencia [Ec.3],

$$I(x, y) = |U_z(x, y) + R(x, y)|^2, \quad (4)$$

corresponde al holograma del objeto. En ausencia de la  $R(x, y)$  se asume una propagación libre del campo  $U_z(x, y)$ . En la figura 1 en la página anterior se representa la geometría de este proceso. Para la reconstrucción del holograma se debe implementar la propagación inversa, usando la función de transferencia de Fresnel inversa  $H_{-z}$ , por lo cual es necesario conocer la distancia  $z$  de propagación<sup>4</sup>.

### III. OBJETIVOS Y METODOLOGÍA

Con el objetivo planteado, se propusieron las siguientes etapas de desarrollo para el proyecto: i) preparar el conjunto de datos para el entrenamiento del modelo de red neuronal convolucional (CNN); ii) diseñar el modelo CNN para el proceso de reconstrucción de imágenes; iii) optimizar los parámetros del modelo según el conjunto de entrenamiento; y iv) medir el desempeño del modelo entrenado, respecto a técnicas convencionales, en hologramas simulados. A continuación se hace un desarrollo más completo de las etapas. (Nota: Los códigos y sus implementaciones se encuentran en el repositorio LabAv2 project CNN Holography).

#### A. Construcción de algoritmo de simulación de Fresnel

Para calcular la función de transferencia de Fresnel [Ec.2], se requiere establecer los parámetros de tamaño de píxel  $dx$ , el tamaño en píxeles  $M$  del campo de entrada, longitud de onda  $\lambda$ , acorde a la siguiente condición:

$$z \leq \frac{Mdx^2}{\lambda}. \quad (5)$$

Una descripción básica del algoritmo se resume en lo siguiente: 1) Cargar una imagen del objeto cuyo holograma se quiere generar. 2) Redimensionar a la resolución del holograma que se quiere generar. 3) Definir un  $z$  que cumpla la condición [Ec.5]. Por defecto, sea  $\lambda = 532nm$ , y  $dx = 8\mu m$ . 4) Calcular la función de transferencia. 5) propagar el campo de entrada. 6) Generar el campo de referencia e interferir con el campo de propagado (omitir en caso de propagación libre) 7) Calcular la intensidad y almacenar la imagen resultante.

#### B. Generación del conjunto de datos

En primer lugar se construyó un data set a partir del data set MNIST, conformado por imágenes de  $28 \times 28$  píxeles de dígitos escritos a mano<sup>8</sup>. Para el entrenamiento del modelo se tomó una muestra aleatoria balanceada de los dígitos. Las imágenes seleccionadas pasaron por las siguientes transformaciones. Se escalaron a tamaño  $100 \times 100$  píxeles. Luego fueron embebidas en matrices de ceros de tamaño  $256 \times 256$  por requerimientos de la arquitectura. A estas imágenes, que identificamos como el campo óptico  $U_0(x_0, y_0)$  se le aplicó la transformada de Fresnel dada por la ecuación 1 con el parámetro  $z$  fijado en 1.5 cm. El resultado de este proceso es una matriz compleja  $H(x, y) = U_z(x, y) + R(x, y)$  que sirve como imagen de entrada para el modelo. La imagen objetivo en este caso es la imagen original del dígito  $U_0(x_0, y_0)$ . El conjunto de 1000 pares holograma-dígito fue separado en dos. Una muestra de 800 para el data set de entrenamiento y las otras 200 para el data set de prueba.

Para hacer pruebas de robustez en el modelo se construyeron además otros dos data sets. El primero de ellos replicando el proceso del data set de entrenamiento pero con los valores de  $z$  igual a 1.5, 3.0, 4.5, 6.0 cm. Esto permitió evaluar el modelo en distancias de propagación diferentes a la distancia con la que fue entrenado. Por otro lado, el segundo data set imitó el procedimiento antes descrito, cambiando únicamente las imágenes base. Para este caso fueron recortes de  $100 \times 100$  píxeles del patrón de la USAF-1951 (derechos reservados a Wikimedia Commons). Ambos data sets dan cuenta de la capacidad de generalización del modelo.

Un último data set se conformó con imágenes reconstruidas por un algoritmo de propagación inversa común. La diferencia sustancial fue que como entrada se le dio sólo la parte real del holograma que se había generado para el data set de entrenamiento.

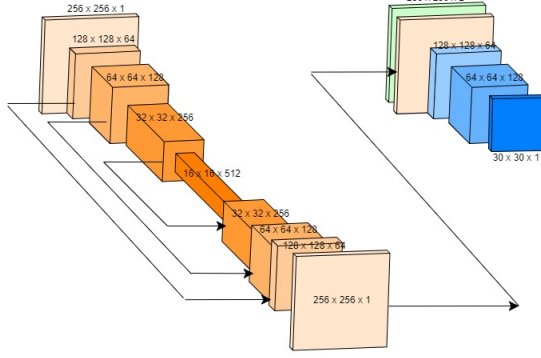


FIG. 2. Estructura por capas convolucionales del modelo. El generador de imágenes (naranja) y el discriminador (azul).

### C. Diseño de la arquitectura de la red

La arquitectura de la red neuronal convolucional fue tomada del artículo Isola<sup>9</sup>, inspirado en el trabajo de Ian Goodfellow<sup>10</sup>. La implementación fue realizada en Colab por [ ]. Sólo se adaptaron las funciones de carga de los datos y la dimensión de las imágenes de entrada a  $256 \times 256 \times 1$ .

Este modelo está conformado por dos redes neuronales convoluciones (CNN) encargadas de realizar dos tareas. El generador recibe imágenes de entrada y a partir de ellas retorna como salida una nueva imagen. Para nuestro caso, el generador recibe la parte real del holograma y tiene como salida el dígito reconstruido. La figura 2 muestra la arquitectura de capas convolucionales del generador. Este consta de un sector donde la imagen es comprimida y se extrae de ella información espacial y de contenido por medio de filtros. Luego pasa el cuello de botella en la parte más baja y pasa por la descompresión en el otro extremo de la red. Hay que notar que los filtros obtenidos en la compresión son pasados al lado simétrico en la descompresión, de manera que no hay pérdidas grandes de información como en un proceso de compresión normal.

La segunda red es el discriminador. Consta solamente del bloque de compresión. Este recibe la imagen de salida del generador y la verdadera imagen o 'ground truth'. Su salida es una imagen que le asigna a parches de la imagen del generador un valor dependiente de si las considera 'realistas' o no.

### D. Funciones de evaluación

Un gran reto en el campo de la generación de imágenes es la evaluación de la calidad de las imágenes de salida<sup>11</sup>. Una propuesta en esta dirección fue dada por Hausel y Ramsauer en 2018 con la implementación de Fréchet Inception Distance (FID) para GANs<sup>11</sup>. El algoritmo FID asocia al conjunto de imágenes reales de dígitos una distribución normal multivariada con media  $\mu_R$  y matriz de covarianza  $\Sigma_R$ . También lo hace para el conjunto de imágenes de salida del generador, con me-

Evaluación del modelo en función del tiempo de entrenamiento.

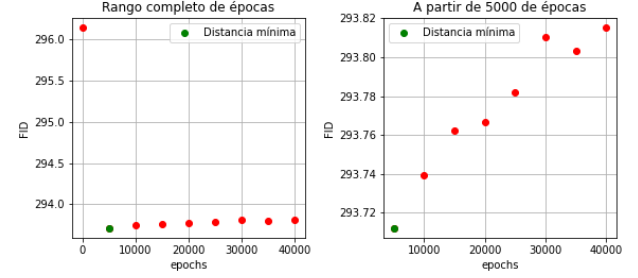


FIG. 3. Fréchet Inception Distance (FID) como función de pérdida en la evaluación del modelo por épocas.

dia  $\mu_G$  y covarianza  $\Sigma_G$ . Finalmente, halla la distancia entre estas dos distribuciones normales mediante la fórmula

$$FID = \|\mu_R - \mu_G\|^2 + \text{Tr}\left\{\Sigma_R + \Sigma_G - 2\sqrt{\Sigma_R \Sigma_G}\right\}. \quad (6)$$

Como función de apoyo, se usó además el coeficiente de similitud estructural (SSIM)

## IV. RESULTADOS Y ANÁLISIS

### A. Pruebas

Se llevó a cabo un proceso de entrenamiento de 40000 épocas. A lo largo de este se registró la métrica FID para la evaluación cada 5000 épocas. La figura 3 muestra los resultados del entrenamiento. Con ello se obtuvo que el tiempo suficiente de entrenamiento está entre 5000 y 10000 épocas. Por este motivo se estableció como versión final de prueba al generador de un modelo entrenado 5000 épocas.

El análisis cualitativo de las imágenes de salida del generador en la figura 4 para el conjunto de prueba son aceptables. El modelo es capaz de reconstruir la forma general del dígito, aunque las reconstrucciones presentan defectos en los bordes. Esta imprecisión puede ser atribuida a la falta de información de los bordes que está contenida en la parte imaginaria del holograma.

### B. Evaluación

Para el análisis cuantitativo se estableció como punto de referencia la calidad de la reconstrucción de imágenes por la transformada de Fresnel inversa. En este caso, el coeficiente de similitud estructural promedio sobre el conjunto de prueba fue de 0.0198. Mientras que el del generador fue 0.0071. Esto muestra claramente que la reconstrucción es mejor que el modelo. Sin embargo, se debe tener en cuenta una desventaja de la reconstrucción usual. La transformada de Fresnel inversa necesita como parámetro la distancia de propagación. En tanto que el modelo no requiere de ningún parámetro externo una vez es entrenado.

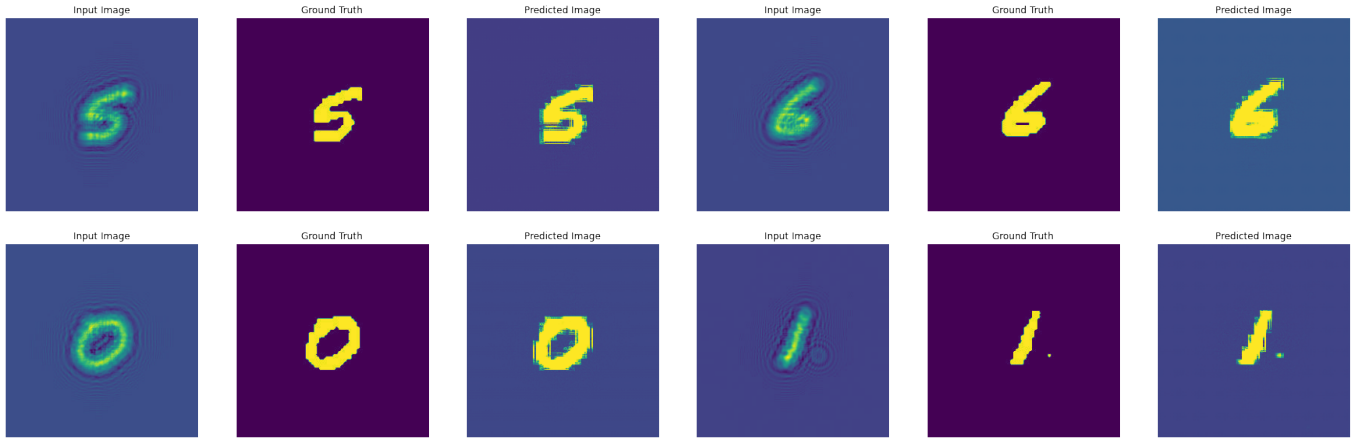


FIG. 4.

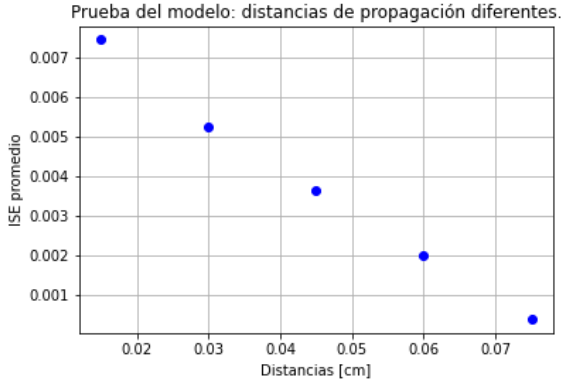


FIG. 5. Índice de similitud estructural entre imágenes del generador y las reales. En función de la distancia creciente de propagación se observó un peor desempeño en la reconstrucción.

Para cada distancia de propagación se tomó el valor promedio del ISE entre las imágenes del generador y las imágenes esperadas. La figura 5 resume estos datos. De allí se notó que el modelo pierde precisión en la reconstrucción para las imágenes con propagaciones diferentes a las del data set de entrenamiento. Esto era de esperar pues estas imágenes están fuera de la clase con la que fue entrenado el modelo.

La prueba con objetos diferentes mostró un desempeño ligeramente mejor. El modelo pudo reconstruir bordes más simples como líneas rectas. Además fue capaz de mantener algunos detalles sobre otros, probablemente debido a su tamaño respecto a la imagen, como lo muestra la figura 6.

## V. CONCLUSIONES

De acuerdo a los objetivos planteados se logró evaluar con éxito el desempeño del modelo en la tarea de reconstrucción. Tanto cualitativa como cuantitativamente a través de las métricas propuestas. Por sí solo, el modelo presentó buenos resultados en la tarea asignada. Teniendo fallos menores asociados

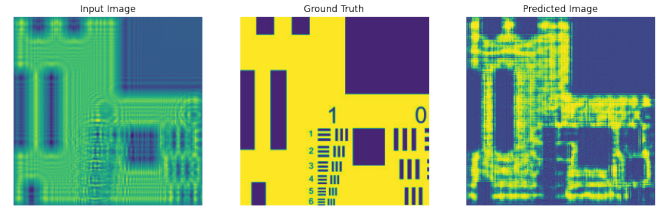


FIG. 6. De izquierda a derecha: parte real del holograma generado, imagen real tomada para el holograma y reconstrucción a través del generador.

a la falta de información deliberada a la que fue sometido.

En contraste con el método de reconstrucción por transformada inversa, el modelo no es suficiente para alcanzar los resultados de dicho método. No obstante, exhibe ventajas como optimización en el tiempo de computo y la no necesidad de conocer la distancia de propagación usada en el registro del holograma para la reconstrucción.

Es posible mejorar los resultados del generador. Por ejemplo implementando una entrada en dos canales que contenga tanto la parte real como la parte imaginaria del holograma, dando cuenta de la fase. O incluso pasando la implementación de la arquitectura a un dominio donde los parámetros son complejos. De modo que el holograma como matriz compleja pueda servir de entrada sin necesidad de separarlo en componentes.

Las métricas usadas, junto con las pruebas de robustez implementadas permitieron determinar que el modelo no puede generalizarse de manera directa. Ni para el caso de otras distancias de propagación, ni para objetos de diferente geometría. Para compensar esta falencia del modelo se propone realizar una nueva serie de entrenamiento con un data set compuesto de hologramas hechos para distancias de propagación más variadas y para diferentes objetos geométricos igual de simples que los dígitos.

## VI. REFERENCIAS

- <sup>1</sup>T. Shimobaba, D. Blinder, T. Birnbaum, I. Hoshi, H. Shiomi, P. Schelkens, and T. Ito, "Deep-learning computational holography: A review," *Frontiers in Photonics* **3** (2022).
- <sup>2</sup>P. Ambs, J.-P. Huignard, and B. Loiseaux, "Holography," in *Encyclopedia of Condensed Matter Physics*, edited by F. Bassani, G. L. Liedl, and P. Wyder (Elsevier, Oxford, 2005) pp. 332–341.
- <sup>3</sup>C. C. Guest, "Holography," in *Encyclopedia of Physical Science and Technology (Third Edition)*, edited by R. A. Meyers (Academic Press, New York, 2003) third edition ed., pp. 381–392.
- <sup>4</sup>U. Schnars and W. Jueptner, *Hologram Recording, Numerical Reconstruction, and Related Techniques* (Springer, Berlin, 2005) Chap. 1-3.
- <sup>5</sup>S. Trester, "Computer-simulated fresnel holography," *European Journal of Physics* **21**, 317 (2000).
- <sup>6</sup>D. G. Voelz, *Computational fourier optics: a MATLAB tutorial*, Vol. 534 (SPIE press Bellingham, Washington, 2011) Chap. 3-4.
- <sup>7</sup>J. Li, Y. Wu, and Y. Li, "Chapter 3 - common diffraction integral calculation based on a fast fourier transform algorithm," (Elsevier, 2010) pp. 257–302.
- <sup>8</sup>L. Deng, "The mnist database of handwritten digit images for machine learning research," *IEEE Signal Processing Magazine* **29**, 141–142 (2012).
- <sup>9</sup>P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition* (2017) pp. 1125–1134.
- <sup>10</sup>I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," (2014).
- <sup>11</sup>M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "Gans trained by a two time-scale update rule converge to a local nash equilibrium," (2017), 10.48550/ARXIV.1706.08500.