elektron ámbar

Original Article

Revista elektron, Vol. 9, No. 2, pp. 47-55 (2025)
https://doi.org/10.37537/rev.elektron.9.2.223.2025

Aplicación de redes neuronales en tomografía computarizada por ultrasonido

Application of Neural Networks in Ultrasound Computed Tomography

Malena Díaz Falvo*[1], Martín G. González*† y Leonardo Rey Vega*†

*Facultad de Ingeniería, Universidad de Buenos Aires
Paseo Colon 850, C1063ACV, Buenos Aires, Argentina
†Consejo Nacional de Investigaciones Científicas y Técnicas, (CONICET)
Godoy Cruz 2290, C1425FQB, Buenos Aires, Argentina
[1]mcdiaz@fi.uba.ar

Recibido: 03/11/25; Aceptado: 12/12/25

***Resumen*— En este trabajo se desarrolló un sistema de reconstrucción de imágenes en el marco de la tomografía computarizada por ultrasonido, utilizando técnicas de aprendizaje profundo para la estimación de mapas de velocidad, asociados a la propagación de ondas acústicas. Se abordó el diseño y entrenamiento de diferentes arquitecturas de redes neuronales y se evaluó su desempeño. Para esto, se generó un conjunto de datos sintético mediante simulaciones y se realizó la adquisición de sinogramas reales mediante un sistema experimental que utiliza un transductor de inmersión.**

Palabras clave: tomografía; ultrasonido; DCN; U-Net.

***Abstract*— This work developed an image reconstruction system within the framework of Ultrasound Computed Tomography, utilizing deep learning techniques for the estimation of velocity maps associated with acoustic wave propagation. The design and training of different neural network architectures were addressed, and their performance was evaluated. To this end, a synthetic dataset was generated through simulations, and the acquisition of real sinograms was performed using an experimental system that employs an immersion transducer.**

Keywords: tomography; ultrasound; DCN; U-Net.

I. Introducción

Las técnicas de obtención de imágenes médicas permiten visualizar estructuras internas del cuerpo de forma no invasiva. Entre ellas, la tomografía se destaca por generar imágenes transversales del cuerpo a partir de mediciones sobre diferentes ángulos [1], [2]. En particular, la tomografía computarizada por ultrasonido (TCUS) surge como una alternativa segura frente a la radiación ionizante, con gran potencial para la detección temprana del cáncer de mama [3], [4]. A diferencia de la radiografía, el ultrasonido se ve afectado en gran medida por fenómenos ondulatorios como reflexión, refracción y difracción [5], [6], lo que convierte la reconstrucción en un problema inverso no lineal de alta complejidad [7], [8].

La TCUS se basa en aplicar un campo acústico conocido sobre un objeto y analizar el campo transmitido o reflejado para estimar propiedades del medio, como la velocidad del sonido o la atenuación acústica [5], [9]. Estas propiedades revelan información sobre la estructura interna del tejido y pueden obtenerse mediante distintas configuraciones de transductores.

Una configuración típica para la adquisición de imágenes tomográficas consiste en rodear el objeto con una serie de transductores, o rotar un transductor alrededor del mismo para sondear el objeto con ondas de ultrasonido y medir la interacción resultante. La opción de girar mecánicamente un transductor alrededor del objeto tiene la ventaja de ser una configuración simple y poco costosa. Por otro lado, la utilización de un arreglo de transductores es generalmente más costosa de implementar, pero acelera enormemente el proceso de adquisición de datos.

Existen varios algoritmos de reconstrucción de imágenes para obtener el mapa de velocidades del objeto de interés. En el presente trabajo, se propone un enfoque basado en redes neuronales para la estimación de los mapas de velocidad del sonido. Para tal fin, se ha generado una base de datos sintética empleando los algoritmos de simulación y reconstrucción utilizados en un estudio previo [10]. Este volumen de datos sintéticos se creó con el objetivo de entrenar las arquitecturas de redes neuronales propuestas y, posteriormente, lograr la reconstrucción del mapa de velocidades asociado a cada medición experimental.

II. Tomógrafo acústico 2-D basado en un único transductor

En el desarrollo de sistemas de TCUS, la precisión de las mediciones depende fuertemente del diseño de la configuración experimental. Una arquitectura robusta y simple no sólo mejora la calidad de los datos, sino que también minimiza la influencia de factores externos, como ruido, interferencias electromagnéticas, variaciones térmicas y errores sistemáticos. Como sistema experimental se utilizó el montaje desarrollado previamente en el laboratorio del Grupo de Láser, Óptica de Materiales y Aplicaciones Electromagnéticas (GLOmAEe), para la adquisición de sinogramas [10].

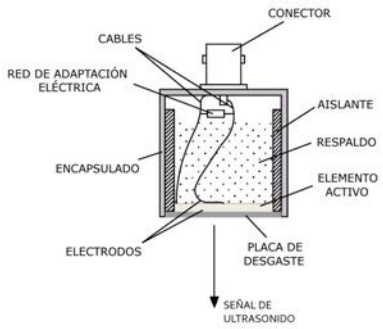

Figura 1. Esquema de un transductor de inmersión.

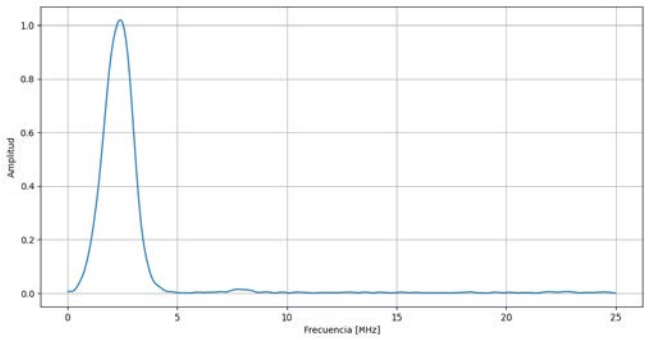

Figura 2. Transformada de Fourier de la señal emitida por el transductor.

II-A. Transductor de inmersión

La Fig. 1 muestra un esquema típico de un transductor de inmersión. Un transductor convierte señales eléctricas en ondas acústicas y viceversa; su función principal es transmitir energía ultrasónica al medio y recibir los ecos reflejados. Entre sus componentes, el más relevante es el elemento activo piezoeléctrico, que efectúa esa conversión electrotromecánica. El piezoeléctrico está polarizado y conectado mediante electrodos al conector eléctrico exterior; a su vez, un respaldo absorbente amortigua vibraciones residuales y la placa de desgaste protege el elemento activo y ayuda a adaptar la impedancia acústica entre el transductor y el medio de acoplamiento [11]. Se utilizó un transductor Olympus V306-SU [12], con frecuencia central de $2{,}25\,\mathrm{MHz}$, patrón de campo no enfocado y diámetro efectivo de $13\,\mathrm{mm}$. El espectro de la señal emitida se muestra en la Fig. 2, donde se observa el pico principal coincidente con la frecuencia central reportada por el fabricante.

En los sistemas de TCUS, el acoplamiento acústico entre el transductor y el objeto resulta crítico para garantizar mediciones confiables. En nuestro caso, se utilizó agua destilada como medio de transmisión del ultrasonido, ya que su impedancia acústica es similar a la de los tejidos biológicos y permite una transmisión eficiente de la energía.

II-B. Configuración experimental

En la Fig. 3 se muestra la cuba de plástico acrílico en la que se realizaron las mediciones. Ésta cuenta con orificios laterales para los transductores, un eje de rotación conectado a un motor paso a paso (Newport PR50CC con controlador ESP-300), juntas de goma para mantener estanqueidad y una válvula de desagüe. Para determinar la velocidad de sonido se midió la temperatura del agua con un termistor NTC calibrado en nuestro laboratorio.

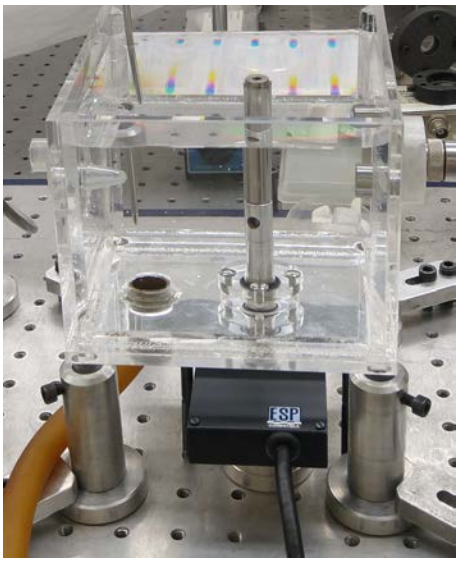

Figura 3. Cuba acrílica utilizada en el sistema de medición.

En la Fig. 4 se presenta el esquemático del sistema de transmisión y recepción de ultrasonido. Durante la transmisión, el generador de pulsos (HP 222A) cortos ($< 25\mathrm{ns}$) de tensión se conecta directamente al transductor, mientras que durante la recepción se acopla al amplificador (Picosecond 5828A) a través del conmutador T/R. Las señales se registraron con un osciloscopio (Tektronix TDS2024B). El control y adquisición de datos se realizaron desde una computadora, mediante un algoritmo implementado en Python basado en la librería PyVISA.

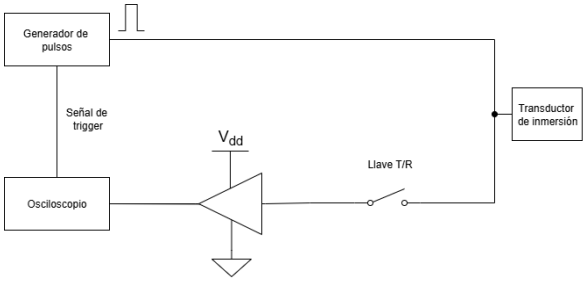

Figura 4. Esquema del sistema experimental de emisión y detección ultrasónica.

II-C. Procedimiento de adquisición

El generador de pulsos excita el transductor, que emite una onda ultrasónica a través del agua. Esta señal interactúa con el objeto bajo estudio, y parte de ella es reflejada hacia el transductor, que ahora actúa como receptor. Una vez finalizada la emisión, la caída de tensión en los terminales del conmutador T/R desciende por debajo del umbral de ± 2 V, lo que provoca que el circuito conmute y permita el paso de la señal acústica recibida. Esta señal atraviesa el interruptor, se dirige al amplificador y, por último, es digitalizada por el osciloscopio. Luego de cada adquisición, el objeto rota un ángulo controlado por el motor paso a paso. La temperatura se mide al inicio y al final de cada sesión

para relevar el cambio de la velocidad del sonido durante la adquisición de un sinograma.

II-D. Ruido experimental

Se realizaron mediciones sin un objeto presente, a fin de relevar y caracterizar el ruido que presenta la configuración. Este ruido puede estar asociado a interferencias eléctricas, provenientes de equipos o dispositivos externos al sistema. La Fig. 5 muestra cinco de las mediciones de ruido obtenidas, digitalizadas por el osciloscopio, junto con el espectro en frecuencia de cada una de esas señales. Para el espectro, se realizó un acercamiento al rango de 0 a 5 MHz, para visualizar claramente las componentes espectrales con mayor aporte.

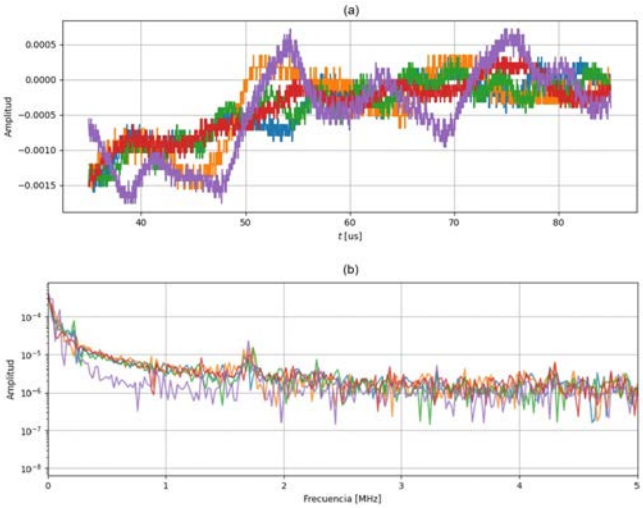

Figura 5. (a) Ejemplo de señales de ruido medidas en ausencia de objeto. (b) Espectro en frecuencia de las señales medidas, con un acercamiento a un ancho de banda de 10MHz.

El ruido adquiere mayor relevancia para objetos con índice de refracción similares al agua o para determinados ángulos de medición donde la relación señal a ruido (SNR) es baja.

III. Redes neuronales profundas

III-A. Redes convolucionales densas

Las redes convolucionales densas (DCN) son redes neuronales convolucionales caracterizadas por tener una conectividad densa, a modo de aumentar el flujo de información entre capas. Esta red introduce conexiones directas desde cada capa a todas las capas posteriores, por lo que cada instancia recibe como entrada la concatenación de las salidas de todas las capas previas. La salida x de la capa ℓ puede expresarse por la siguiente ecuación:

$$x_\ell = H_\ell([x_0, x_1, ..., x_{\ell-1}]) \quad (1)$$

donde $[x_0, x_1, ..., x_{\ell-1}]$ representa la concatenación de las salidas producidas por las capas anteriores, y $H_\ell(\cdot)$ es una transformación no lineal. Esta transformación consiste en una serie de operaciones convolucionales que pueden estar acompañadas por etapas de normalización, funciones de activación no lineales como ReLU y operaciones de agrupamiento (pooling), dependiendo del diseño de la arquitectura.

Dado que la concatenación de (1) no es viable cuando el tamaño de los mapas de características cambia, la red es dividida en varios bloques densamente conectados, definidos por la transformación H_ℓ. Entre ellos, se definen capas de transición, que incluyen normalizaciones, convoluciones y agrupamientos para reducir la dimensión de los datos. En la Fig. 6 se muestra un diagrama en bloques de una red de tipo DCN, utilizada para la clasificación de imágenes. Se observa que la misma cuenta con tres bloques densos y dos capas de transición entre los mismos, las cuales están compuestas por una operación de convolución (C) para reducir la cantidad de canales, seguida de una capa de agrupamiento (P) para reducir la resolución espacial de cada canal. Además, a la entrada de la red se encuentra una capa de convolución para extraer las características básicas de la imagen y, a la salida, una capa de agrupamiento seguida de una capa completamente conectada (L) para obtener la clasificación esperada de la imagen de entrada.

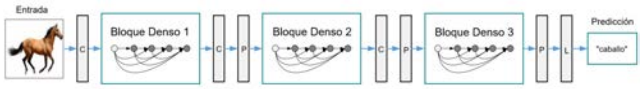

Figura 6. Diagrama de una red DCN.

Por otro lado, en los bloques densamente conectados, cada operación H_ℓ produce k mapas de características, lo que implica que la capa ℓ tiene $k_0 + k \cdot (\ell - 1)$ entradas, donde k_0 es el número de canales de entrada. A este hiperparámetro k se lo denomina tasa de crecimiento, y su función principal es controlar el incremento progresivo de la información extraída en la red a medida que se agregan capas [13].

III-B. U-Net

En la Fig. 7 se muestra la otra red usada en este trabajo, que tiene una arquitectura tipo U-Net [14]. Estas redes reciben su nombre por la forma de su estructura, ya que cuentan con un camino descendente, uno ascendente y uno de conexión entre ambos, resultando en una estructura con forma de U (ver Fig. 7). El primer camino se denomina ruta de contracción (codificador) y está compuesto por distintas capas convolucionales, junto con operaciones de submuestreo, que buscan reducir la resolución de la entrada, aumentando la cantidad de canales. De esta manera se extraen las características más relevantes de la imagen de entrada para cada resolución, codificando los datos. Cada capa de submuestreo reduce la resolución de la imagen y aumenta la profundidad o número de canales. El otro camino se denomina ruta de expansión (decodificador) y está compuesto por capas convolucionales transpuestas, las necesarias para decodificar los datos hasta su resolución original. El punto medio entre estos dos caminos se denomina cuello de botella y es la capa que representa el mayor punto de abstracción respecto a la entrada original, ya que los datos se encuentran en su máxima compresión. Por último, existen las conexiones de atajo, que son conexiones entre las distintas rutas que buscan acelerar el entrenamiento y alivianar el problema del desvanecimiento del gradiente [14]. Este problema se presenta cuando los gradientes se vuelven demasiado pequeños, ya que continúan disminuyendo y

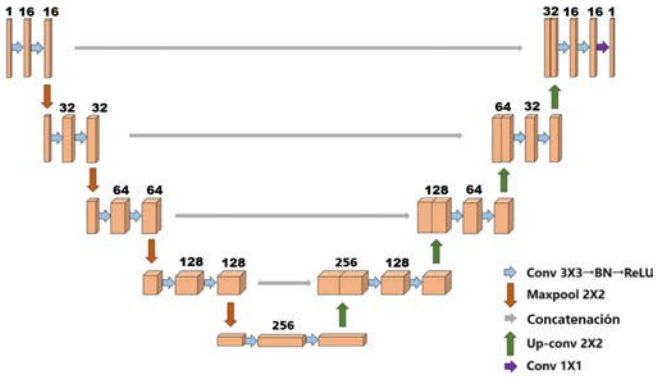

Figura 7. Diagrama de una U-Net.

actualizando pesos de las capas, hasta que se vuelven tan cercanos a 0 que la red prácticamente no se actualiza.

En la Fig. 7, además, se presentan los valores de los parámetros que caracterizan a esta red utilizados en este trabajo. Se observa que la entrada posee un único canal, que es transformado a 16 canales mediante una doble convolución. A partir de allí, el número de canales se duplica en cada etapa del codificador, pasando de 16 a 32, luego a 64, 128 y finalmente 256 en el cuello de la red. Cada una de estas etapas reduce la dimensión espacial mediante operaciones de *max-pooling*. En el camino del decodificador, las dimensiones espaciales se recuperan mediante operaciones de sobremuestreo, mientras que el número de canales se reduce progresivamente a la mitad. En cada nivel del decodificador, las características recuperadas se concatenan con aquellas provenientes del codificador en la misma escala, permitiendo preservar tanto la información local como la global. Finalmente, una última convolución proyecta los 16 canales a la cantidad deseada de canales de salida.

III-C. Autoencoders variacionales

Los *autoencoders* son redes neuronales diseñadas para aprender una representación comprimida de los datos de entrada y, posteriormente, reconstruirlos. Están compuestos por un codificador, que extrae las variables latentes relevantes; un cuello de botella, que contiene la representación comprimida; y un decodificador, que reconstruye la entrada original a partir de dichas variables.

A diferencia de los *autoencoders* determinísticos, los *autoencoders* variacionales (VAE) aprenden una distribución probabilística continua del espacio latente, en lugar de una representación fija. Mediante una reparametrización, la variable latente se define como

$$z = \mu_x + \sigma_x \odot \epsilon, \quad \epsilon \sim \mathcal{N}(0, I), \tag{2}$$

donde $\odot$ denota la multiplicación elemento a elemento, $\mathcal{N}(0, I)$ la distribución normal estándar y μ_x y σ_x la media y desviación estándar de la distribución del espacio latente, respectivamente. Esta formulación permite separar los componentes determinísticos y estocásticos, facilitando el entrenamiento y la generación de nuevas señales [15], [16].

La Fig. 8 ilustra un esquema general de un VAE: el codificador produce los parámetros μ_x y σ_x, a partir de los cuales se obtiene la variable latente z, que luego el decodificador utiliza para reconstruir la salida $\hat{x}$.

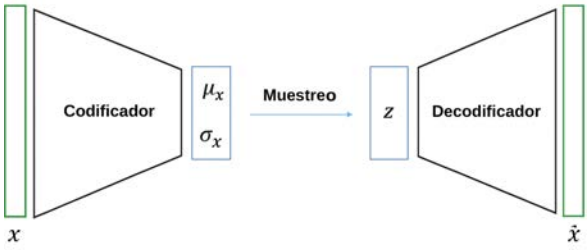

Figura 8. Esquema de un *autoencoder* variacional.

IV. Generación del conjunto de datos

IV-A. Imágenes verdaderas

La generación de este conjunto de datos sintético se abordó inicialmente creando un conjunto de imágenes en blanco y negro, con una resolución de 200×200 píxeles.

Para la representación de los objetos bajo estudio, se utilizaron figuras geométricas básicas, específicamente círculos y polígonos regulares con una cantidad de lados entre 3 y 6 (triángulos, cuadrados, pentágonos y hexágonos). Para cada figura se utilizaron diferentes escalas y ángulos de rotación aleatorios. Existen tres configuraciones principales en la generación de estas figuras:

1. **Figuras sólidas:** figuras geométricas sin modificaciones internas y posicionadas en el centro de la imagen, véase Fig. 9(a).
2. **Figuras sólidas con sustracción interna:** Se parte de una figura geométrica maciza y luego se sustraen entre 1 y 3 figuras más pequeñas de su interior; estas pueden ser de cualquiera de los tipos de figuras mencionadas. La sustracción se realiza en distintas posiciones dentro de la figura principal, generando patrones huecos o perforados, véase Fig. 9(b).
3. **Figuras espejadas:** Para cada imagen, se generan dos figuras del mismo tipo, con diferentes tamaños, ambas ubicadas alrededor del centro de la imagen y en cuadrantes opuestos, véase Fig. 9(c).

En total, se generaron 1000 imágenes destinadas a conformar el conjunto de datos de entrenamiento y otras 100 imágenes para el de testeo.

IV-B. Simulación de sinogramas

Las imágenes generadas se emplearon para simular sinogramas representativos del banco experimental. El algoritmo desarrollado rota cada imagen y emite un pulso ultrasónico, registrando las señales reflejadas. Para la simulación acústica se utilizó el programa j-Wave, un simulador numérico basado en JAX, que permite diferenciación automática, paralelización en GPU y resolución eficiente de ecuaciones de onda mediante discretizaciones espectrales o de diferencias finitas [17], [18].

El dominio de simulación fue discretizado en una grilla finita, aplicando condiciones de contorno de *Perfectly Matched Layer* (PML) para evitar reflexiones causadas por los límites del recinto de simulación. El transductor se modeló como una línea de fuentes puntuales sincronizadas, cuya

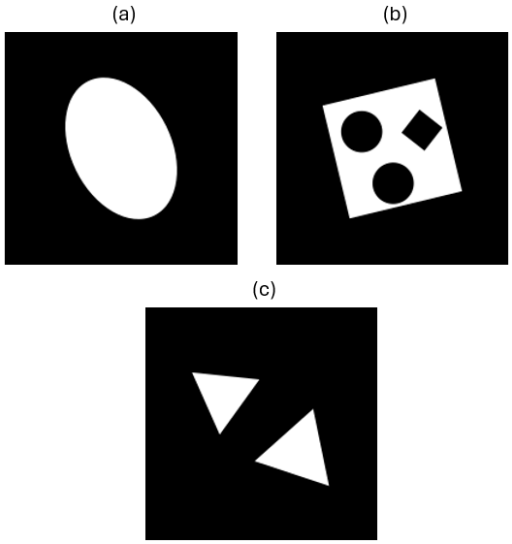

Figura 9. (a) Figura maciza generada (caso 1). (b) Figura generada con sustracción interna (caso 2). (c) Figuras espejadas generadas (caso 3)

aproximación es válida siempre que las dimensiones físicas del sensor en la configuración experimental sean pequeñas en comparación con la longitud de onda de las señales acústicas. En la Fig. 10 se expone una comparación del pulso emitido por el transductor y el simulado, utilizado en la generación de sinogramas. Esta última fue generada aplicando una rampa decreciente junto con una ventana gaussiana a la amplitud de una señal senoidal de 2.25 MHz, a modo de aproximar la señal utilizada al pulso emitido por el transductor.

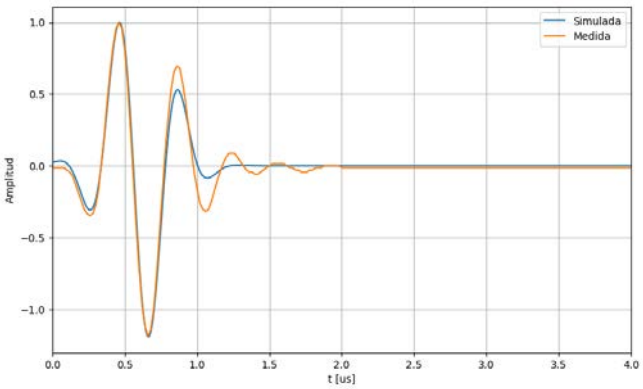

Figura 10. Comparación entre la señal medida emitida por el transductor y la simulada.

La velocidad del sonido del medio (v_s) y del objeto (v_r) se asignó a partir de distribuciones uniformes y teniendo en cuenta las condiciones que se tienen en la configuración experimental:

$$v_s \sim \mathcal{U}(1480, 1500) \text{ m/s} \quad \text{(agua)}$$

$$v_r \sim \mathcal{U}(1600, 1620) \text{ m/s} \quad \text{(goma)}$$

resultando en índices de refracción $n_r \in [1.067, 1.095]$. Para cada imagen, se generaron tres sinogramas con distintas combinaciones de velocidades, obteniendo un total de 3000 imágenes para el conjunto de entrenamiento y 300 para el de evaluación.

Cada simulación consistió en $N_a = 90$ ángulos de rotación y señales temporales de $N_t = 2500$ muestras. El pulso ultrasónico utilizado replica la señal medida experimentalmente. El mismo fue generado a partir de aplicar una rampa decreciente junto con una ventana gaussiana a la amplitud de una señal senoidal. Las señales reflejadas obtenidas para cada ángulo conforman los sinogramas.

IV-C. Generación de ruido coloreado

Para reproducir las condiciones experimentales, se generó ruido coloreado mediante un VAE entrenado con 270 señales de ruido medidas sin objeto. El modelo aprendió la distribución estadística del ruido real y permitió sintetizar nuevas instancias a partir de ruido blanco gaussiano. En la Fig. 11 se muestran cinco señales de ruido generadas por la red, tanto en el dominio temporal como en el de la frecuencia. Estas señales fueron adicionadas a los sinogramas simulados, obteniendo un conjunto de datos más representativo del sistema experimental.

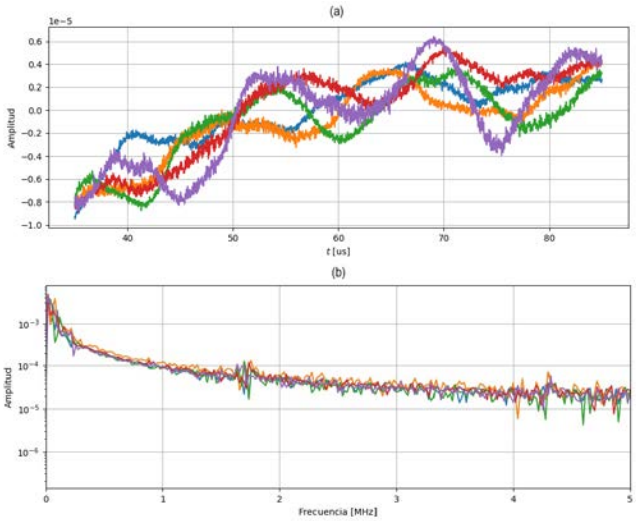

Figura 11. (a) Ejemplos de señales de ruido generadas por la red VAE implementada en este trabajo. (b) Espectro en frecuencia de las señales generadas.

IV-D. Algoritmo de reconstrucción clásico

Se analizó la respuesta de las redes implementadas ante entradas que contienen información espacial, comparándolas con reconstrucciones obtenidas mediante un enfoque clásico. Para ello, los sinogramas se reconstruyeron utilizando el algoritmo descrito en [10], el cual modela la propagación y detección de ondas acústicas en un medio homogéneo, adaptado a la geometría circular del sistema. El algoritmo requiere definir parámetros experimentales, entre ellos el desvío estándar del ruido temporal S_{noise}. La estimación espectral de la señal reflejada se obtiene según:

$$S_w(f) = \frac{P_r(f) P_t^*(f)}{|P_t(f)|^2 + S_{\text{noise}}^2}, \tag{3}$$

donde $P_r(f)$ y $P_t(f)$ representan las transformadas de Fourier de las señales reflejada y emitida, respectivamente. El término S_{noise}^2 estabiliza el filtro ante frecuencias con muy baja amplitud, maximizando la respuesta a ecos coherentes

con el pulso transmitido, debido al numerador que calcula la correlación cruzada entre la señal medida y la transmitida.

Cada sinograma es proyectado sobre una grilla bidimensional centrada en el eje de rotación. Para cada ángulo θ_j, el tiempo de ida y vuelta al píxel (x_i, y_i) se calcula como:

$$t_a(i,\theta_j) = \frac{2}{v_s}\Big[(R_s\cos\theta_j - x_i)\cos\theta_j + (R_s\sin\theta_j - y_i)\sin\theta_j\Big], \tag{4}$$

siendo θ el ángulo de rotación y R_s el radio del transductor. La contribución de cada ángulo es acumulada según:

$$F_i = \sum_{j=1}^{N_a} \tilde{\Psi}_\theta(t_a(i,\theta_j))\,\Delta\theta, \tag{5}$$

obteniéndose la imagen reconstruida F_i. A partir de ahora, este algoritmo se denominará USRT.

Se evaluaron distintos tamaños de imágenes, registrando la PSNR y el tiempo de cómputo. Se encontró que imágenes mayores a 256×256 la PSNR no mejora significativamente, mientras que el tiempo de reconstrucción aumenta de forma considerable. Por este motivo, se adoptó 256×256 para generar el conjunto de datos sintéticos de entrenamiento.

La tomografía por reflexión presenta como limitación la pérdida de información en bajas frecuencias [10]. Por lo tanto, para emular este efecto, se aplicó un filtro pasa-altos sobre las imágenes originales, enfatizando los bordes y eliminando componentes de baja frecuencia. El filtro se implementó mediante la convolución con el kernel de 5×5:

$$\begin{bmatrix} -1 & -1 & -1 & -1 & -1 \\ -1 & 1 & 2 & 1 & -1 \\ -1 & 2 & 4 & 2 & -1 \\ -1 & 1 & 2 & 1 & -1 \\ -1 & -1 & -1 & -1 & -1 \end{bmatrix}.$$

Las imágenes filtradas conforman las imágenes objetivo del conjunto de datos.

V. Redes implementadas

Se utilizaron dos topologías para abordar el problema de reconstrucción de imágenes tomográficas. La primera consta de una arquitectura híbrida, donde se utiliza una red DCN, cuya salida se encuentra conectada a una U-Net. Esta red toma como entrada los sinogramas generados y devuelve una imagen reconstruida con la información sobre los contornos internos y externos del objeto.

Otro enfoque utilizado fue el uso de una red U-Net para el filtrado de los artefactos presentes en métodos de reconstrucción clásicos. De esta forma, se buscó que la red funcione como un filtro que logre discernir entre la información de alta frecuencia correspondiente al objeto y la correspondiente a los artefactos que introduce el algoritmo.

Las redes se entrenaron utilizando el conjunto de datos generado y se validaron adicionalmente con mediciones experimentales. Como funciones de error se emplearon la raíz cuadrática media (RMSE) y el índice de similitud estructural para datos de punto flotante (DSSIM) [19] combinadas de la siguiente forma:

$$\Phi = \alpha\,\mathrm{RMSE} + \beta\,\mathrm{DSSIM}, \quad \alpha = \beta = 0.5, \tag{6}$$

Para evaluar el desempeño del modelo en mediciones reales, se utilizaron sinogramas obtenidos a partir de la configuración experimental. Las reconstrucciones de estos sinogramas se exponen en la Fig. 12. Los mismos se corresponden con mediciones de 90 ángulos de una goma rectangular de tamaño $18{,}5\,\mathrm{mm}\times11{,}6\,\mathrm{mm}$ (izquierda) y de un cilindro de aluminio de $12{,}7\,\mathrm{mm}$ de diámetro (derecha).

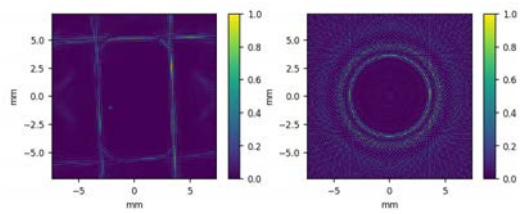

Figura 12. Reconstrucciones de sinogramas obtenidos mediante la configuración experimental.

Todos los entrenamientos se realizaron con una computadora con una CPU Intel i9-10900F, 128 GB de RAM y dos GPU NVIDIA RTX-3090 de 24 GB cada una.

V-A. DCN + U-Net con información USRT

Se evaluó el desempeño de la red según dos variantes. En primer lugar, se probó introducir el mapa de velocidades estimado por el algoritmo USRT a la U-Net, como canal adicional, y luego estimar los mapas de velocidad sin esta información.

A su vez, se evaluaron dos estrategias de entrenamiento:

- **Entrenamiento conjunto en un paso (E1P)**: actualizar todos los parámetros de a red simultáneamente.
- **Entrenamiento en dos pasos (E2P)**: pre-entrenar la DCN, fijar sus pesos y entrenar la U-Net.

El entrenamiento se realizó en 200 épocas para E1P; en E2P se emplearon fases separadas para DCN y U-Net, ambas de 200 épocas cada una. Además, se utilizó el optimizador de Adam para llevar a cabo el aprendizaje de la red.

Dos ejemplos de testeo con datos sintéticos se presentan en la Fig. 13(a) donde se muestra que la red reduce los artefactos presentes en la reconstrucción USRT pero con marcadas irregularidades en los contornos.

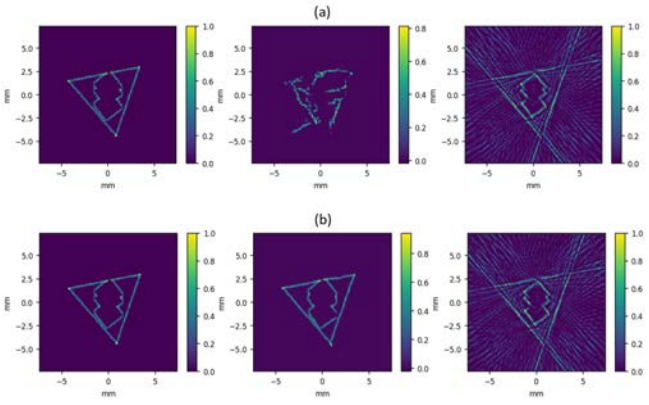

Figura 13. Comparación (objetivo / predicción / USRT) para DCN + U-Net con información USRT, con datos sintéticos. (a) Caso E1P. (b) Caso E2P.

En la Fig. 14(a) se puede ver que sólo el cilindro metálico fue correctamente identificado cuando la red fue aplicada a mediciones reales.

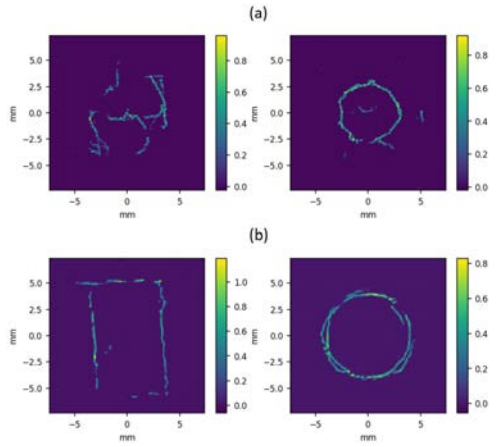

Figura 14. Predicciones (DCN + U-Net con USRT) a partir de sinogramas experimentales. (a) Caso E1P. (b) Caso E2P.

En las Figs. 13(b) y 14(b) se puede apreciar de forma cualitativa que las predicciones mejoran considerablemente al modificar el método de entrenamiento a un enfoque E2P, y que es posible identificar la forma original del objeto en las mediciones reales, aunque los contornos presenten irregularidades con respecto al objeto original.

V-B. DCN + U-Net sin información USRT

Se entrenó nuevamente la red pero esta vez sin incluir la reconstrucción por USRT como segundo canal a la entrada de la U-Net. El objetivo fue evaluar si el modelo podía reconstruir los mapas de velocidad utilizando únicamente la información proveniente de los sinogramas.

Las predicciones con datos sintéticos (ver Fig. 15(a)) muestran que, si bien se preservan las estructuras generales, la red no logra definir contornos nítidos.

Con las mediciones experimentales (ver Fig. 16(a)) las predicciones no presentan mejoras respecto al caso de la subsección V-A, mostrando formas indefinidas y ruidosas.

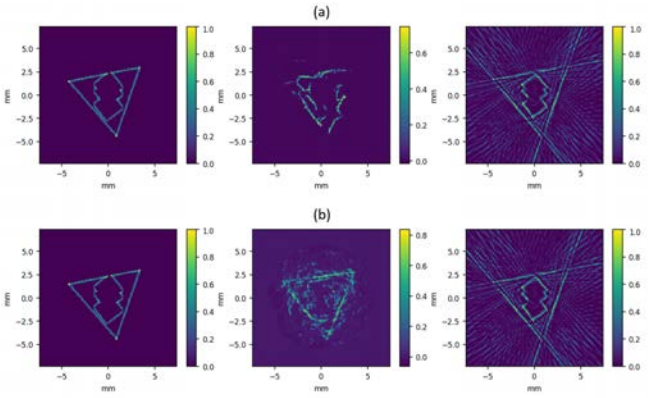

Figura 15. Comparación (objetivo / predicción E2P / USRT) para DCN + U-Net sin información USRT, con datos sintéticos. (a) Caso E1P. (b) Caso E2P.

En el E2P, la función de error mostró grandes saltos en el error de validación (ver Fig. 17), lo que evidencia la dificultad del modelo para generalizar sin información auxiliar de la reconstrucción.

Los resultados obtenidos a partir del conjunto de testeo (Fig. 15(b)) y de las mediciones reales (Fig. 16(b)) confirman que la red no logra reconstruir adecuadamente los contornos, ni es posible distinguir las formas de los objetos reales.

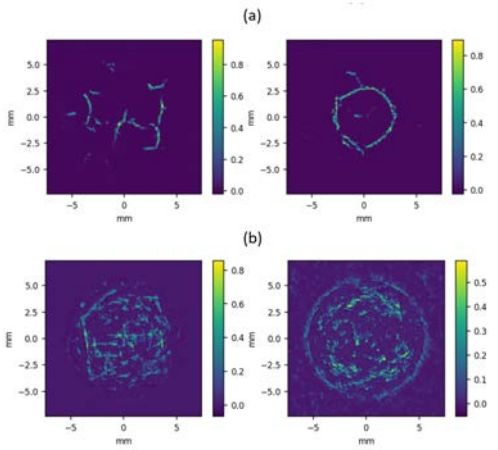

Figura 16. Predicciones (DCN + U-Net sin USRT) a partir de sinogramas experimentales. (a) Caso E1P. (b) Caso E2P.

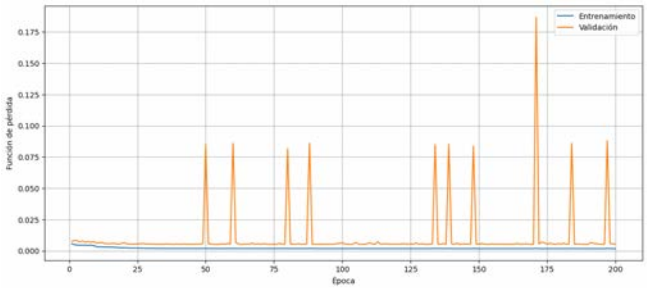

Figura 17. Función de error durante el entrenamiento de la U-Net en el E2P para DCN + U-Net sin información USRT.

En este enfoque, la red recibe como entrada las reconstrucciones obtenidas por el método clásico, y aprende a filtrar los artefactos y transformar las imágenes para que sean más cercanas a las verdaderas.

En la Fig. 18 se presentan los resultados en base al conjunto de datos de testeo. La red logra ajustar sus parámetros de forma tal que la predicción se aproxima notablemente a la imagen verdadera, eliminando los artefactos presentes en la reconstrucción obtenida con USRT.

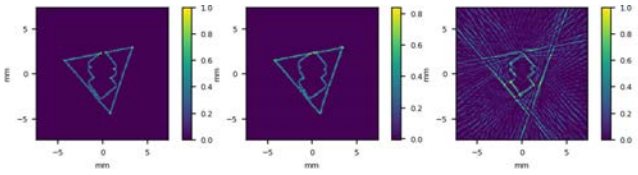

Figura 18. Comparación entre la imagen verdadera, la salida de la U-Net y la reconstrucción USRT.

Posteriormente, se evaluó la red sobre reconstrucciones provenientes de mediciones reales. Como se muestra en la Fig. 19(a), la U-Net logra preservar las formas geométricas de los objetos, aunque no consigue eliminar completamente los artefactos asociados al algoritmo USRT aplicado a mediciones.

Finalmente, se intentó entrenar la red para que la misma conserve las componentes de baja frecuencia en las

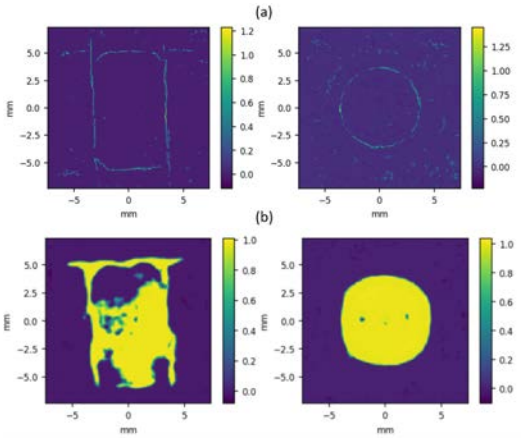

Figura 19. Aplicación de la red U-Net sobre reconstrucciones de mediciones reales. (a) Sin mantener las componentes de baja frecuencia. (b) Manteniendo las componentes de baja frecuencia.

imágenes de salida. Para este caso, la red fue entrenada con imágenes objetivo sin filtrado previo. En las predicciones presentadas en la Fig. 19(b) se observan regiones con textura no uniforme dentro de los objetos macizos, por lo que la red no logra conservar de forma precisa las componentes de baja frecuencia.

VI. Comparación de resultados

En la Fig. 20 se muestra la imagen de referencia utilizada para evaluar las distintas arquitecturas. La Fig. 21 presenta las reconstrucciones obtenidas a partir de un mismo sinograma, incluyendo los resultados de la U-Net, de la red híbrida DCN + U-Net con y sin información USRT y de USRT. A simple vista, las salidas de la U-Net y de la red híbrida con información USRT y E2P son las que más se aproximan a la imagen verdadera.

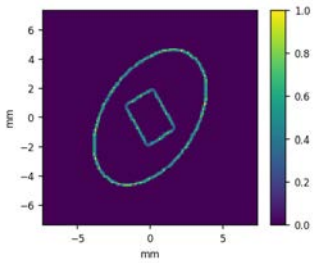

Figura 20. Imagen verdadera de referencia.

La Tabla I resume los valores promedio de las métricas de evaluación, junto con el correspondiente desvío estándar: SSIM, PC, RMSE y PSNR. En todas las métricas, los mejores valores se obtienen para la red híbrida (DCN + U-Net) con información USRT y E2P, seguida de la U-Net, lo que concuerda con los resultados simulados cualitativos presentados en la sección anterior.

Por lo tanto, se tiene que tanto la U-Net como la red híbrida con información USRT y E2P lograron brindar reconstrucciones que superan al método USRT en todas las métricas cuantitativas. Sin embargo, desde una perspectiva puramente visual, cabe mencionar que un observador podría reconocer con mayor facilidad la forma original del objeto en las reconstrucciones obtenidas mediante el método USRT

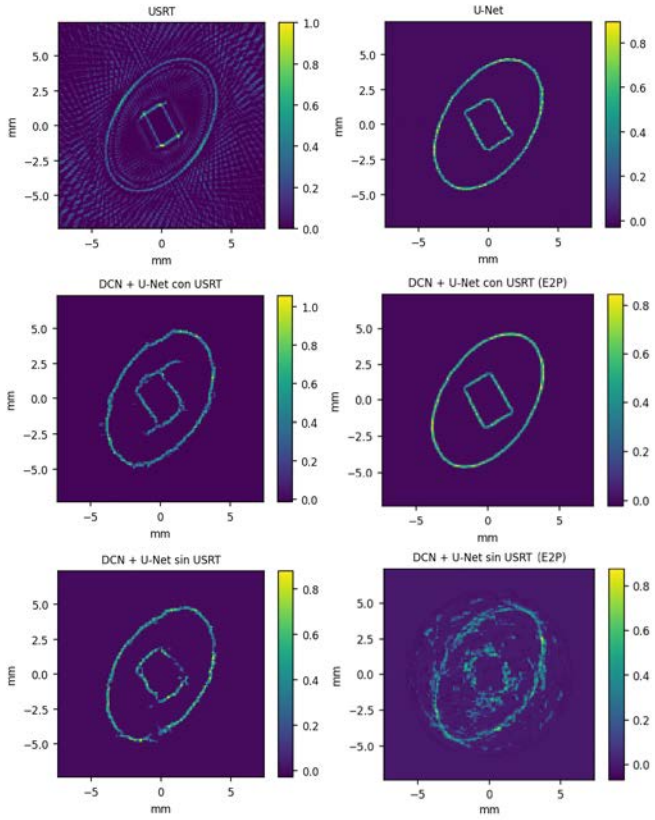

Figura 21. Comparación de las reconstrucciones obtenidas mediante U-Net, DCN + U-Net (con y sin información USRT) y USRT.

Topología	USRT	E2P	SSIM	PC	RMSE	PSNR [dB]
DCN + U-Net	Sí	No	0.82 ±0.06	0.39 ±0.17	0.089 ±0.02	21.3 ±2.1
	Sí	Sí	**0.97** ±0.02	**0.90** ±0.08	**0.039** ±0.01	**28** ±3
	No	No	0.83 ±0.06	0.40 ±0.17	0.088 ±0.02	21.3 ±2.1
	No	Sí	0.51 ±0.09	0.36 ±0.14	0.109 ±0.02	19.4 ±1.6
U-Net	-	-	**0.96** ±0.03	**0.89** ±0.08	**0.041** ±0.01	**28** ±3
USRT	-	-	0.06 ±0.06	0.16 ±0.07	0.124 ±0.02	18.3 ±2

Tabla I
Métricas de evaluación y su desvío estándar para las distintas redes con el conjunto de datos de testeo.

que en las generadas por las redes híbridas. Si bien este algoritmo presenta métricas cuantitativas significativamente inferiores, sus artefactos son sistemáticos y predecibles, lo que facilita su identificación visual. En cambio, las redes pueden introducir distorsiones menos familiares, que dificultan la interpretación de la imagen.

VII. Conclusiones

En este trabajo, se implementaron y compararon los siguientes enfoques de aprendizaje profundo para la reconstrucción de imágenes: (i) una red DCN seguida por una U-Net y (ii) una U-Net aplicada como post-procesamiento sobre reconstrucciones USRT.

La arquitectura DCN + U-Net que incorporó información USRT y E2P alcanzó las mejores métricas cuantitativas. El E2P permitió que esta arquitectura aprendiera de manera más efectiva las imágenes objetivo, aprovechando la información espacial ya contenida en la reconstrucción USRT. En segundo lugar se ubicó la U-Net. Al recibir como entrada una reconstrucción que ya contiene la mayoría de los datos relevantes, la red fue capaz de preservar los contornos, aunque no logró erradicar en su totalidad los artefactos presentes. Por el contrario, las variantes que partieron directamente del sinograma mostraron pérdidas de detalle y contornos imprecisos, reflejando la dificultad de predecir los mapas de velocidad sin información espacial adicional. Aun así, las redes híbridas superaron al método USRT en las métricas evaluadas, mostrando el potencial del aprendizaje profundo para mejorar la calidad y velocidad de la reconstrucción.

Entre las posibles mejoras se destaca la ampliación y diversificación del conjunto de datos, incorporando simulaciones más realistas para aumentar la robustez del modelo. En particular, las simulaciones generadas en este trabajo consideran el modelado de la velocidad del sonido, con geometrías bien definidas y transiciones abruptas entre materiales, mientras que las mediciones reales, presentan heterogeneidades internas, bordes irregulares, atenuación acústica o fenómenos de dispersión, que no se encuentran modelados en los datos sintéticos generados. Si bien se incorporó ruido experimental en el dominio temporal para reducir parcialmente esta brecha, la ausencia de un modelado explícito de atenuación y de texturas internas constituye una limitación del conjunto sintético utilizado. La incorporación de simulaciones que contemplen estas propiedades físicas más realistas permitiría reducir el desajuste entre datos sintéticos y experimentales, favoreciendo así una mejor generalización del modelo.

También es posible mejorar la sensibilidad del sistema mediante el uso de una etapa amplificadora sobre la señal de excitación del transductor, lo que permitiría obtener señales reflejadas de mayor amplitud y generar reconstrucciones más definidas. Finalmente, futuras líneas de trabajo podrían centrarse en la extensión a modelos de generación de datos sintéticos con mallas tridimensionales y el análisis de su impacto en la estabilidad y convergencia de la red.

AGRADECIMIENTOS

Este trabajo fue financiado por la Universidad de Buenos Aires (UBACYT 20020190100032BA), CONICET (PIP 11220200101826CO) y la Agencia I+D+i (PICT 2020-01336).

REFERENCIAS

[1] A. C. Kak and M. Slaney, *Principles of Computerized Tomographic Imaging*. Philadelphia: Society for Industrial and Applied Mathematics, 2001.

[2] C. Høilund, "The radon transform," Master's thesis, Aalborg University, 2007.

[3] N. Duric, C. Li, O. Roy, and S. Schmidt, "Acoustic tomography: Promise versus reality," in *AIP Conference Proceedings*, vol. 1335, no. 1, 2011, pp. 25–31.

[4] X. Lin, H. Shi, Z. Fu, H. Lin, S. Chen, X. Chen, and M. Chen, "Dynamic speed of sound adaptive transmission/reflection ultrasound computed tomography," *Sensors*, vol. 23, no. 7, p. 3694, 2023.

[5] F. A. Duck, *Physical Properties of Tissues: A Comprehensive Reference Book*. London: Academic Press, 1990.

[6] D. Carroll, L. McKay, C. Hacking *et al.* (2024) Attenuation (ultrasound). Radiopaedia.org. Accessed: 2025-10-31. [Online]. Available: https://radiopaedia.org/articles/attenuation-ultrasound

[7] J. Virieux and S. Operto, "An overview of full-waveform inversion in exploration geophysics," *Geophysics*, vol. 74, no. 6, pp. WCC1–WCC26, 2009.

[8] T. C. Robins, C. Cueto, J. Cudeiro, O. Bates, O. C. Agudo, G. Strong, L. Guasch, M. Warner, and M.-X. Tang, "Dual-probe transcranial full-waveform inversion: A brain phantom feasibility study," *Ultrasound in Medicine and Biology*, vol. 49, no. 1, pp. 283–298, 2023.

[9] W. Han, D. N. Sinha, K. N. Springer, and D. C. Lizon, "Noninvasive measurement of acoustic properties of fluids using an ultrasonic interferometry technique," *The Journal of the Acoustical Society of America*, vol. 104, no. 3, pp. 1404–1411, 1998.

[10] M. Reigada, M. G. González, and L. R. Vega, "Estudio y desarrollo de un sistema para tomografía ultrasónica bidimensional," *Elektron*, vol. 7, no. 2, pp. 40–47, 2023.

[11] Olympus NDT, *Introduction to Ultrasonic Transducers*, Olympus Corporation, 2010, accessed: 2025-10-31. [Online]. Available: https://www.olympus-ims.com/en/ndt-tutorials/transducers/

[12] Evident. Immersion transducers: Thickness and flaw inspection solutions. Accessed: 2025-10-31. [Online]. Available: https://ims.evidentscientific.com/en/probes/single-and-dual-element/immersion-transducers

[13] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 4700–4708.

[14] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, ser. Lecture Notes in Computer Science, vol. 9351. Springer, 2015, pp. 234–241, available at https://arxiv.org/abs/1505.04597.

[15] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013, available at https://arxiv.org/abs/1312.6114.

[16] D. Bergmann and C. Stryker, "What is a variational autoencoder?" IBM Research Blog, 2021, available at https://research.ibm.com/blog/what-is-variational-autoencoder, Accessed: 2025-10-31.

[17] A. Stanziola, S. R. Arridge, B. T. Cox, and B. E. Treeby, "j-wave: An open-source differentiable wave simulator," *arXiv preprint arXiv:2202.04633*, 2022.

[18] R. Frostig, M. J. Johnson, and C. Leary, "Compiling machine learning programs via high-level tracing," in *SysML Conference*, 2018, available at https://github.com/google/jax.

[19] A. Baker, A. Pinard, and D. Hammerling, "DSSIM: a structural similarity index for floating-point data," *arXiv preprint arXiv:2202.02616*, 2022.