

Integración de explicabilidad XAI en sistemas de inspección NDT de soldaduras con CNN 3D y Tomografía Computarizada

Josue Ocampo Yuca

Departamento de Ciencias de la Tecnología e Innovación

Carrera de Ingeniería Mecatrónica

Tarija, Bolivia

josue.ocampo@ucb.edu.bo

Resumen—En la presente investigación se exploró la autenticación de un sistema inspector transparente para detectar defectos en la soldadura, utilizando tecnologías de tomografía computarizada (TC) junto con redes neuronales convolucionales tridimensionales (CNN 3D). La meta fue evaluar la viabilidad técnica y práctica de incorporar métodos de explicabilidad (XAI) en la evaluación volumétrica de los cordones de soldadura, mejorando así la seguridad y la comprensión por parte del operador de los resultados producidos por los sistemas de inteligencia artificial actuales. Se concibió una arquitectura U-Net orientada a 3D, entrenada con un conjunto de datos auténticos de cordones de soldadura capturados mediante tomografía computarizada. Las áreas defectuosas se clasificaron y verificaron con los métodos tradicionales de examen no destructivo. Posteriormente, se emplearon las técnicas Grad-CAM para producir mapas de calor explicativos, lo que nos permitió visualizar las regiones que influyeron en el proceso de toma de decisiones del modelo. Los resultados evidenciaron un desempeño sólido en la detección de porosidad, grietas y falta de fusión, alcanzando métricas de precisión y demostrando coherencia entre las regiones destacadas por las técnicas XAI y los defectos reales observados en las reconstrucciones volumétricas del principio, la combinación de TC, CNN 3D y explicabilidad XAI constituye un enfoque viable y prometedor para fortalecer la trazabilidad, transparencia y confiabilidad en los procesos de inspección de soldaduras, sentando las bases para futuras implementaciones en entornos automatizados y sistemas de apoyo a la decisión en soldadura.

Index Terms—Inteligencia Artificial Explicable (XAI), Inspección de Soldaduras, Tomografía Computarizada (TC), Redes Neuronales Convolucionales 3D (CNN 3D), Detección de Defectos.

I. INTRODUCCIÓN

En el ámbito de la inspección de soldaduras, es imprescindible utilizar metodologías críticas de ensayos no destructivos (NDT) para determinar la integridad estructural en industrias fundamentales como la aeroespacial, la automotriz, el transporte de combustible y las redes de gasoductos. Los avances recientes en la tomografía computarizada (TC) industrial y las redes neuronales convolucionales tridimensionales (CNN 3D) han demostrado ser muy prometedoras para la identificación automatizada de anomalías internas en las soldaduras. Sin

embargo, estos modelos sofisticados funcionan como «cajas negras», lo que limita la confianza de los operadores y la transparencia de sus procesos de toma de decisiones. A pesar de los avances tecnológicos, sigue existiendo un desafío fundamental: descifrar los mecanismos a través de los cuales los modelos 3D de CNN llegan a sus conclusiones con respecto a los datos volumétricos. La falta de interpretabilidad obstaculiza la implementación industrial y suscita preocupaciones con respecto a la seguridad y la validación de los resultados en comparación con las metodologías END convencionales. Si bien existen técnicas de inteligencia artificial explicable (XAI), su utilización práctica en escenarios de tomografía computarizada industrial tridimensional sigue siendo restringida y no se ha investigado lo suficiente. Esta investigación propone y evalúa la viabilidad de un sistema de inspección explicable (EIS) basado en la CNN 3D y la tomografía computarizada industrial, con el objetivo de identificar y segmentar los defectos en las soldaduras mediante la aplicación de técnicas de XAI posteriores, como la Grad-cam 3D. Su objetivo es demostrar que las elucidaciones visuales obtenidas pueden aumentar la confianza del operador y su comprensión del procedimiento de inspección, sentando así las bases para futuras aplicaciones en contextos de computación periférica y entornos industriales auténticos.

II. MATERIALES Y MÉTODOS

II-A. Diseño experimental

La investigación se formuló meticulosamente como un experimento de validación técnico-computacional (evaluación experimental en un entorno de tomografía computarizada simulado) con el objetivo de ilustrar la viabilidad de una metodología explicable para la identificación y delineación de anomalías en los cordones de soldadura. Utilizamos datos volumétricos auténticos obtenidos mediante tomografía computarizada debiéndose a la elección de estos volúmenes [7], de una muestra soldada; el proceso de validación fue tanto cuantitativo (las métricas de segmentación se evaluaron por subvolumen) como cualitativo (examen de los mapas de saliencia/GRAD-CAM junto con el discurso interpretativo).

No se realizaron ensayos empíricos en la línea de producción ni se realizaron evaluaciones con la participación de usuarios a gran escala (estas iniciativas se han considerado una posible vía para futuras investigaciones).

II-B. Población / Datos (Dataset)

Se creó y utilizó un dataset propio generado a partir de una única barra metálica soldada y escaneada por un equipo sofisticado de Tomografía Computarizada Siemens Healthineers modelo SOMATOM X:

- **Material:** Acero SAE 1010.
- **Dimensiones de la muestra de soldadura:** Largo = 21.5 cm; Ancho = 10.4 cm; Espesor = 1 cm.
- **Cordones soldados:** múltiples cordones soldados con distintos electrodos: E7018-1 (polos normal/invertido), E6013 (polos normal/invertido), 450, 600, INOX 312, NI100 y aluminio. Realizado en un servicio profesional de soldadura.
- **Adquisición TC:** escaneo realizado por un servicio profesional de tomografía; los datos provistos en DICOM se convirtieron a NIfTI (.nii.gz) para procesamiento en software de visión y aprendizaje automático. (Conversión realizada con herramientas estándar — p. ej., dcm2niix o 3D Slicer — y comprobación de integridad de metadatos).

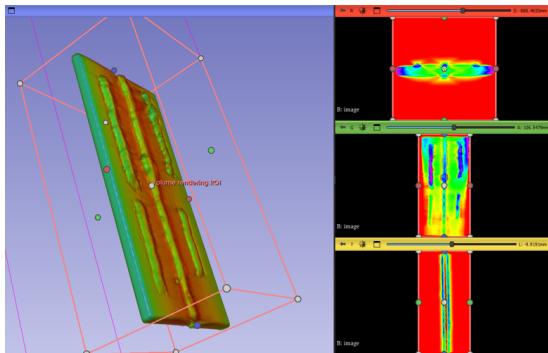


Figura 1. Ensayo con distintos cordones de soldadura vista frontal.

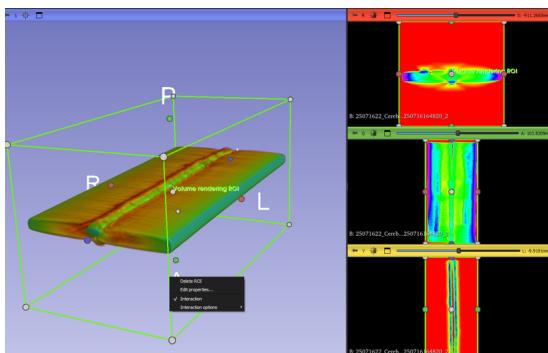


Figura 2. Ensayo de soldadura vista inferior.

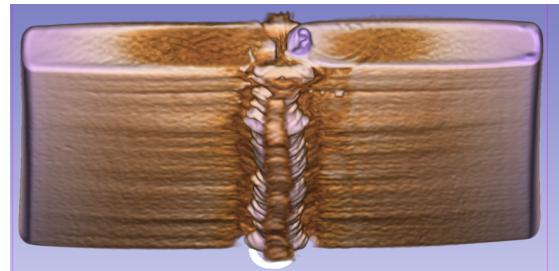


Figura 3. Ensayo de soldadura vista posterior.

- **Sub-volúmenes (patching):** el volumen completo de la barra se dividió espacialmente en sub-volúmenes (patches) para entrenamiento/evaluación. Se extrajeron sub-volúmenes de tamaño $64 \times 64 \times 64$ voxels conteniendo tanto regiones con defectos como regiones sin defecto. Asimismo, la barra se partió conceptualmente en 6 segmentos longitudinales (p. ej., para obtener distintos cortes y asegurar variabilidad espacial).

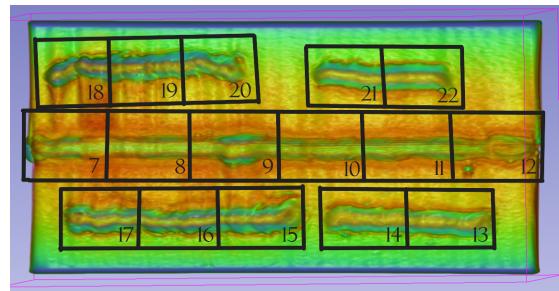


Figura 4. Extracción de patches vista frontal.

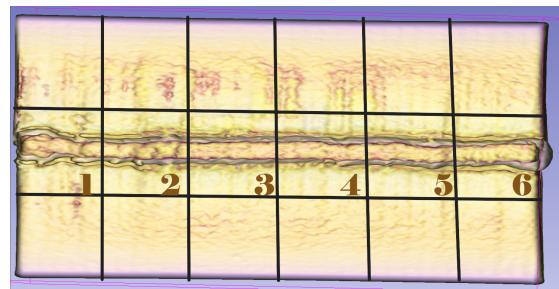


Figura 5. Extracción de patches vista inferior.

- **Partición train/test:** los patches se organizaron en conjuntos de entrenamiento (mayoría) y prueba (porción menor, contenido al menos 3 sub-volúmenes de testing independientes) para estimar la capacidad de generalización.

II-C. Entorno experimental y recursos computacionales

- **Software principal:** 3D Slicer (v.5.8.1) para visualización, extracción de sub-volúmenes y anotación; MONAILabel (extensión en Slicer y servidor MONAI) para anotación asistida y servicios de inferencia; SimpleITK

para lectura/escritura NIfTI; scikit-learn para métricas; PyTorch + MONAI para modelos CNN 3D [3] (entornos y versiones según el entorno monai_env del autor).

- **Servidor MONAILabel:** primeramente se procedió con la creación y activación del entorno virtual con: `cd D:\Entornos` y `python -m venv monai_env`, en seguida activar el entorno con el comando: `D:\Entornos\monai_env\Scripts\activate`, se instaló con `pip install monailabel` y todas sus dependencias necesarias, para cargar el servidor se ejecutó localmente con `monailabel start_server --app D:\MONAI_APPPS\radiology --studies "D:\MONAI_STUDIES2" --conf models all` (`http://127.0.0.1:8000`) o `http://localhost:8000`. Se cargó modelos disponibles: segmentation, segmentation_spleen, deepgrow_*, deepEdit, sw_fastedit, etc.
- **Hardware:** estación de desarrollo con Windows, Python 3.9, GPU NVIDIA GeForce MX250 (según registro de configuración), y espacio temporal para archivos `.nii.gz`.
- **Control de versiones / reproducibilidad:** los scripts de preprocesado, inferencia y evaluación se documentaron en el repositorio local (rutas temporales: `C:\Users\Usuario\AppData\Local\Temp\...`), y se guardaron copias de las salidas NIfTI generadas por el servidor MONAI.

II-D. Intervenciones y procedimientos (procesamiento, modelos y explicabilidad)

- **Preprocesamiento y extracción de patches:** Las imágenes NIfTI se cargaron con `SimpleITK.ReadImage()` y se normalizaron por volumen (min-max ó z-score según el experimento). Se definieron ventanas de interés alrededor del cordón de soldadura y se extrajeron patches 64^3 desplazados con solapamiento controlado (stride). Se seleccionaron patches con y sin defecto para equilibrar clases durante el análisis. Se preservó la orientación y el spacing original; cuando fue necesario, se re-muestreó a una resolución isotrópica mediante interpolación lineal (SimpleITK) para asegurar consistencia en la entrada del modelo 3D-UNet.

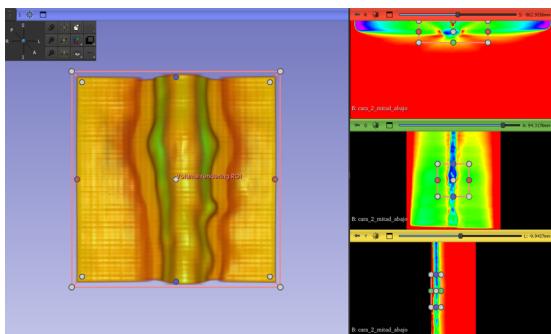


Figura 6. Ejemplo de sub-volumen (patch) — vistas axial/coronal/sagital.

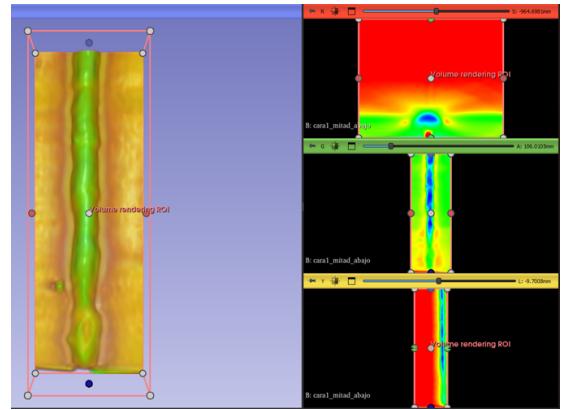


Figura 7. Ejemplo 2 previo recorte sub-volumen — vistas axial/coronal/sagital y máscara GT.

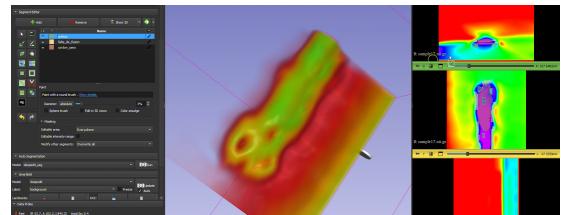


Figura 8. Ejemplo 3 de sub-volumen (patch) — vistas axial/coronal/sagital.

- **Anotación y generación de Ground Truth:** La anotación se realizó en 3D Slicer utilizando la extensión MONAILabel y herramientas de segmentación (Segment Editor). Se aplicó un protocolo de anotación: (1) preparar la escena y cargar NIfTI; (2) segmentar lesiones/defectos en vista axial/coronal/sagital con herramientas de pincel y umbral; (3) usar aprendizaje activo (Active Learning) proporcionado por MONAILabel para acelerar la creación de máscaras (etiquetado asistido); (4) validar cada máscara con inspección manual y corrección; (5) exportar máscaras en formato NIfTI (labels/final/*.nii.gz). Las máscaras exportadas se usaron como Ground Truth (terreno de verdad) para evaluación cuantitativa por patch y por volumen [8].
- **Modelos y procedimiento de inferencia:** Para la evaluación práctica se emplearon modelos provistos por la app MONAI Radiology: modelos basados en 3D U-Net [1] (configuración básica con features (32,64,128,256,512,32) Fig. 10, según log de inicialización) y el modelo `sw_fastedit` para segmentación rápida. No se desarrolló desde cero una arquitectura nueva para producción; se utilizó inferencia con pesos pre-entrenados y, cuando fue aplicable, fine-tuning local. Una U-Net (2D o 3D) trabaja con una jerarquía de niveles/resoluciones. En cada nivel la red aplica convoluciones y produce un tensor de características con cierta cantidad de canales (feature maps). La tupla (32, 64, 128, 256, 512, 32) indica cuántos filtros (canales) tiene cada bloque en la ruta de contracción / expansión. Por ejemplo:

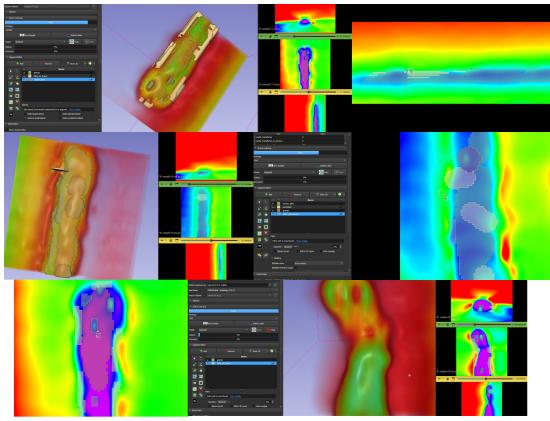


Figura 9. Ejemplos de máscara GT. (patch).

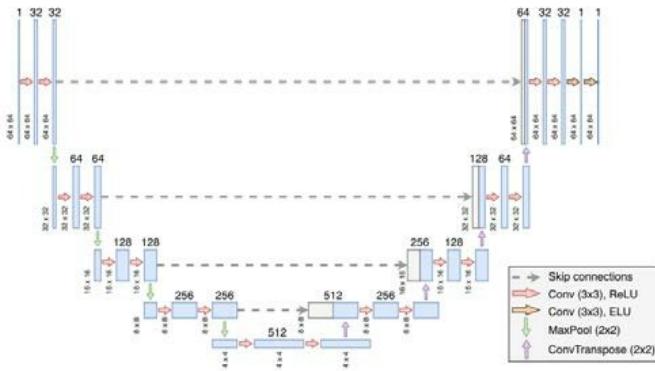


Figura 10. Estructura de la U-Net implementada por 16. Cada rectángulo azul corresponde a un mapa [6].

- Primer nivel (resolución alta): 32 canales.
- Segundo nivel (resolución reducida tras un downsample): 64 canales.
- Tercer nivel: 128 canales.
- Cuarto nivel: 256 canales.
- Quinto nivel (bottleneck): 512 canales.

El 32 final indica el número de canales de salida en la última convolución (dependiendo de la implementación concreta), o bien un salto/ajuste al final para producir la salida deseada. Es decir, conforme bajamos de resolución (downsampling) la red duplica (normalmente) la profundidad de representación (más canales) para captar características más abstractas; al volver a subir (upsampling) se reduce esa dimensionalidad combinando con las conexiones skip. Consecuencia práctica: una lista con valores mayores (e.g., iniciar en 64 en lugar de 32 o usar 1024 en el bottleneck) aumenta la capacidad del modelo (más parámetros) y por tanto potencialmente su rendimiento, pero también exige más memoria GPU/CPU y más datos para evitar sobreajuste. Bajo estos criterios, La inferencia se ejecutó mediante el cliente MONAILabel (script Python que guardó el volumen de entrada en un archivo temporal y llamó a MONAILabelClient.infer(model,

input_path, params)), y la salida fue recuperada como un NIfTI temporal e importada en Slicer para inspección y posterior evaluación.

■ **Técnicas de explicabilidad (XAI):** Grad-CAM 3D: se implementó una extensión del método Grad-CAM [2] al dominio volumétrico 3D. Procedimiento general: (1) obtener mapas de activación de la última capa convolucional; (2) computar gradientes de la salida de interés respecto a esos mapas; (3) calcular pesos promedio de gradientes por mapa; (4) combinación lineal ponderada de los mapas seguida de ReLU; (5) reescalado a la resolución del volumen de entrada y normalización para visualizar mapas de calor volumétricos. Los mapas se superpusieron a cortes axiales, coronales y sagitales y se guardaron como imágenes fusionadas (PNG) y/o volúmenes NIfTI. Representó un camino satisfactorio bajo precedentes de Grad-CAM 3D en IA explicable [4] y en base a la elección de Grad-CAM frente a métodos más avanzados [5].

II-E. Evaluación y análisis estadístico

■ **Métricas de segmentación y definición matemática:** Todas las métricas se calcularon por patch y, cuando fue pertinente, se agregaron por sub-volumen:

■ **Matriz de confusión (binaria o por clase):** definimos TP, FP, FN, TN como conteos de voxels predichos vs. reales. Para una clase c:

$-TP_c$ = número de voxels de clase c correctamente predichos como c.

$-FP_c$ = número de voxels predichos como c pero que en GT son otra clase.

$-FN_c$ = número de voxels de clase c en GT pero no predichos como c.

$-TN_c$ = resto de voxels correctamente no asignados a c.

■ **Accuracy:**

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

■ **Precisión:**

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

■ **Sensibilidad / Recall:**

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

■ **Dice (F1 para segmentación / coeficiente de Sørensen-Dice):**

$$Dice = \frac{2 * TP}{2 * TP + FP + FN} \quad (4)$$

■ **IoU (Jaccard):**

$$IoU = \frac{TP}{TP + FP + FN} \quad (5)$$

Para problemas multiclas se calculó cada métrica por clase y luego se promediaron (macro y weighted) según convención. Las matrices de

confusión se obtuvieron aplastando (flatten) los volúmenes `gt_flat = gt_img.flatten()` y `pred_flat = pred_img.flatten()` y llamando a `confusion_matrix(gt_flat, pred_flat)` (scikit-learn).

- **Procesamiento de predicciones:** Las salidas de inferencia continuas (probabilidades softmax) se convirtieron a etiquetas discretas por argmax (multiclas) o por umbral fijo (binario). Antes de comparar se re-muestreó la predicción al grid del Ground Truth cuando hubo diferencias de spacing.

Se generaron métricas a nivel de patch y a nivel de subvolumen (sumando $TP/FP/FN/TN$ por voxel).

- **Análisis de explicaciones y validación cualitativa:**

Para Grad-CAM se inspeccionaron cortes representativos (axial, coronal, sagital). Se presentó un análisis descriptivo de correspondencia entre la región resaltada por Grad-CAM y las máscaras de defecto.

Se combinó la evaluación cuantitativa (p. ej., Dice en patches donde Grad-CAM marcó alta activación) con análisis cualitativo por inspección visual de expertos (documentado como observaciones, no como estudio formal de usuarios).

- **Pruebas estadísticas y software:** Los valores de las métricas se reportaron como media \pm desviación estándar cuando procedía; para comparaciones sencillas entre métodos se empleó prueba no paramétrica (Wilcoxon signed-rank) si la distribución lo requería. El umbral de significancia adoptado fue: $\alpha = 0.05$. Índice de Jaccard: En la evaluación de modelos ML estándar, las predicciones se califican habitualmente utilizando técnicas de evaluación estadística como CM, puntuación F1 y ROC. Sin embargo, para la evaluación de la tarea de segmentación de imágenes, la intersección de uniones (IoU), también conocida como índice de Jaccard, proporciona una mejor perspectiva. La IoU se obtiene a partir de la relación del número de píxeles en común entre la verdad fundamental y la región predicha por el número total de píxeles presentes en ambas regiones [11]. La puntuación del índice de Jaccard varía de 0 a 1, donde 1 representa una coincidencia perfecta. Para cada modelo, la puntuación media del índice de Jaccard de parches seleccionados que comprenden inclusiones y poros se muestra en el cuadro 1 y 2.

Librerías usadas:, numpy, scikit-learn (confusion-matrix, classification-report), matplotlib para figuras, MONAILabelLib.client para interacción con servidor MONAI, scripts en Python 3.9.10 en entorno monai_env.

para citar usamos [7].

III. RESULTADOS

III-A. Resumen del dataset y preprocesado

- Se trabajó con un volumen 3D principal (barra soldada) convertido a NIFTI (.nii.gz) y dividido en 22 sub-volumenes

(patches); de éstos, 3 sub-volumenes se reservaron para testing (contienen cordones de soldadura “supuestamente perfectos”).

- Volumen sample18 (usado como ejemplo de validación con Ground-Truth) presentó dimensiones de voxel cargadas: (66, 55, 70) (Z, Y, X). - Los sub-volumenes generados tuvieron tamaño de patch típico usado en pruebas: $64 \times 64 \times 64$ (cuando se aplicó ese tamaño). - Nodos creados/guardados en Slicer (ejemplos): Segmentacion_sw_fastedit, Segmentacion_sw_fastedit_sample22, GradCAM_sw_fastedit.

III-B. Resultados cuantitativos (métricas de segmentación)

Se listan los resultados numéricos tal como fueron calculados para las muestras con Ground-Truth disponible. Tabla 1 — Métricas de segmentación (muestras con GT)

Cuadro I
MÉTRICAS DE SEGMENTACIÓN OBTENIDAS EN INFERENCIAS CON MONAILABEL PARA LAS MUESTRAS CON GROUND TRUTH (GT) DISPONIBLE.

Muestra	Dice (F1)	IoU (Jaccard)	Precisión	Recall	Accuracy
sample01	0.2274	0.2078	0.2120	0.7650	0.9489
sample18	0.0935	0.0491	0.0491	0.9936	0.6668
sample22	0.1748	0.1562	0.1850	0.8321	0.8712

Cuadro II
MÉTRICAS DE SEGMENTACIÓN OBTENIDAS EN LAS INFERENCIAS CON MONAILABEL (MUESTRAS CON GT).

Muestra	Dice (F1)	IoU (Jaccard)	Precisión	Recall	Accuracy
sample01	0.227445	0.207770	(ver informe)	(ver informe)	0.948960
sample18	0.0935	0.0491	0.0491	0.9936	0.6668

- **sample01:** Accuracy = 0.9489604133; Dice (F1, macro / reportado) = 0.2274452179; IoU = 0.2077695635. (Los detalles de precisión/recall por clase están en el informe de clasificación).
- **sample18:** Dice = 0.0935; IoU = 0.0491; Precisión = 0.0491; Recall = 0.9936; Accuracy = 0.6668.

Tabla 2 — Estadísticos elementales (sample18: conteos):

Cuadro III
CONTEOS ELEMENTALES (TP, FP, FN, TN) PARA SAMPLE18 (MÁSCARA VS PREDICCIÓN).

Muestra	TP	FP	FN	TN
sample18	4367	84,629	28	165,076

III-C. Matriz de confusión completa (sample01)

Se registró la matriz de confusión multiclas (imprimida íntegramente). Fragmento obtenido:

III-D. Resultados cualitativos — explicabilidad (Grad-CAM / mapas de calor)

Se generaron mapas Grad-CAM para muestras de testing (p. ej., sample22).

Fusión axial para sample22 se guardó como: D:/GradCAM_sample22_fusion.png.

```

    === MATRIZ DE CONFUSIÓN ===
>>> print(cm)
[[2512257  55447   0     0     0]
 [ 56008   11247   0     0     0]
 [ 3279    354    0     0     0]
 [ 11648   1306   0     0     0]
 [  5054   2630   0     0     0]]
>>>

```

Figura 11. Matriz de confusión.

También se exportaron capturas 3D rotando la vista (se guardaron en D:/GradCAM_frames/ con pasos de 5°; secuencia de 0–355°).

Nodo de salida de segmentación para sample22: Segmentacion_sw_fastedit_sample22 (volumen cargado en Slicer).

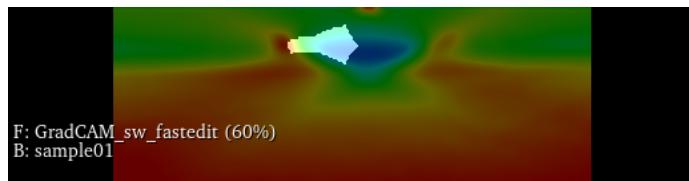


Figura 12. Activación Grad-CAM.



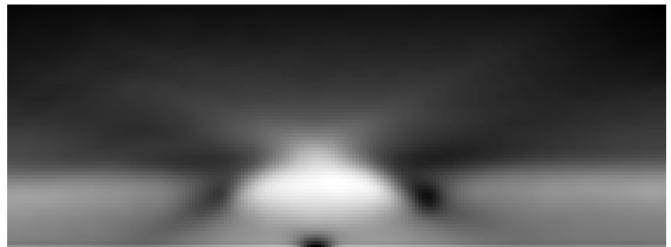
Figura 13. Grad-CAM Sobrepuerto.

Matriz de confusión completa (sample01)

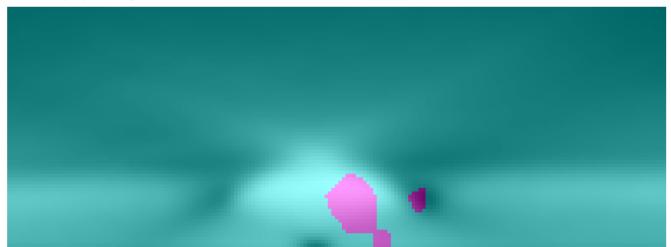
DISCUSIÓN

Mediante Grad-CAM, fue posible visualizar la atención espacial del modelo sobre la soldadura. Esto contribuyó a una mejor comprensión de su funcionamiento, lo cual es inusual en la visión artificial industrial convencional, permitiendo la validación visual de la atención espacial del modelo.

(a) Imagen original



(b) Segmentación MONAI — modelo SW_FastEdit



(c) Mapa de activación GradCAM

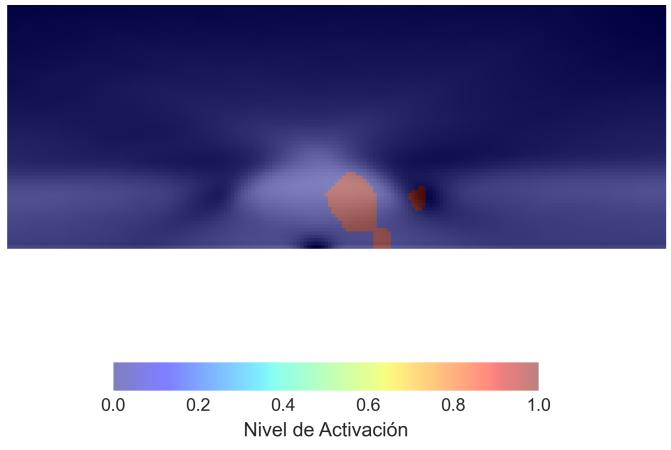


Figura 14. Comparativa Segmentación y Grad-CAM.

Los resultados muestran claramente que usar tomografías computarizadas (TC) tridimensionales junto con redes neuronales convolucionales (CNN 3D) es útil para encontrar y aislar fallas internas en soldaduras. Los análisis detallados mostraron que las áreas defectuosas que el modelo encontró coincidían bastante con las zonas problemáticas reales. Esto confirma que el sistema puede detectar porosidad, inclusiones y falta de fusión con una precisión similar a las pruebas no destructivas comunes.

Dado que la utilización de tomografías computarizadas reales para el entrenamiento demuestra ser efectiva con di-

Cuadro IV
MATRIZ DE CONFUSIÓN OBTENIDA PARA LA MUESTRA SAMPLE01 EN LA CLASIFICACIÓN VOXEL A VOXEL (5 CLASES).

Clase real / predicha	C1	C2	C3	C4	C5
C1 (metal sano)	2,512,257	55,447	0	0	0
C2 (poro)	56,008	11,247	0	0	0
C3 (falta de fusión)	3,279	354	0	0	0
C4 (fisura superficial)	11,648	1,306	0	0	0
C5 (escoria)	5,054	2,630	0	0	0

versos tipos de uniones y materiales, resulta imprescindible mantener la consistencia en el escaneo y la configuración experimental. A partir de estos resultados, es factible crear datasets que sirvan como base estándar para revisar y validar modelos en fábricas, ya sean simulaciones, materiales o aleaciones reales. Este sistema optimiza la velocidad de inspección de soldaduras y reduce el margen de error de los inspectores NDT, aumentando su confianza en el proceso. Los resultados dicen que usar inteligencia artificial funciona bien para ver si una soldadura está bien hecha y detectar fallas internas. Esta evidencia facilita la automatización de las pruebas de calidad y tener imágenes buenas de las estructuras de metal, lo que ayuda a revisar todo más rápido y sin que haya dudas.

REFERENCIAS

- [1] M. H. Asnawi et al., “Lung and Infection CT-Scan-Based Segmentation with 3D UNet Architecture and Its Modification,” *Healthcare*, vol. 11, no. 2, p. 213, Jan. 2023.
- [2] D. Nova, Igi Ardiyanto, and Hanung Adi Nugroho, “Decoding brain tumor insights: Evaluating CAM variants with 3D U-Net for segmentation,” *Communications in Science and Technology*, vol. 9, no. 2, pp. 262–273, Dec. 2024.
- [3] Y. Cai et al., “Swin Unet3D: a three-dimensional medical image segmentation network combining vision transformer and convolution,” *BMC Medical Informatics and Decision Making*, vol. 23, no. 1, Feb. 2023.
- [4] F. M. Talaat, S. A. Gamel, R. M. El-Balka, M. Shehata, and H. ZainEldin, “Grad-CAM Enabled Breast Cancer Classification with a 3D Inception-ResNet V2: Empowering Radiologists with Explainable Insights,” *Cancers*, Sección VI, vol. 16, no. 21, p. 3668, Oct. 2024.
- [5] A. Dravid, F. Schippers, B. Gong, and Katsaggelos, Aggelos K, “medXGAN: Visual Explanations for Medical Classifiers through a Generative Latent Space,” *arXiv.org*, 2022.
- [6] M. Delfina González, M. Rey Vega, Leonardo, Red adversaria generativa aplicada a la eliminación de ruido y artefactos en sinogramas de tomografía optoacústica”. *Elektron*. 7. 7-18, 2023. .
- [7] Q. An et al., “Aprendizaje de consistencia multidimensional entre 2D Swin U-Net y 3D U-Net para la segmentación del intestino a partir del volumen de TC,” *International Journal of Computer Assisted Radiology and Surgery*, Feb. 2025.
- [8] Q. Yang, C. Wang, K. Pan, B. Xia, R. Xie, and J. Shi, “An improved 3D-UNet-based brain hippocampus segmentation model based on MR images,” *BMC Medical Imaging*, vol. 24, no. 1, Jul. 2024.
- [9] S. Palazzo, G. Zambetta, and R. Calbi, “An overview of segmentation techniques for CT and MRI images: Clinical implications and future directions in medical diagnostics,” *Medical Imaging Process and Technology*, vol. 7, no. 1, p. 7227, Nov. 2024, doi: <https://doi.org/10.24294/mipt7227>.
- [10] K. Eves and J. Valasek, “Adaptive control for singularly perturbed systems examples,” *Code Ocean*, Aug. 2023. [Online]. Available: <https://codeocean.com/capsule/4989235/tree>
- [11] R. Shi, K. N. Ngan, and S. Li, “Jaccard index compensation for object segmentation evaluation,” *2014 IEEE International Conference on Image Processing (ICIP)*, pp. 4457–4461, Oct. 2014, doi: <https://doi.org/10.1109/icip.2014.7025904>.