

Identificación y caracterización de materiales 2D usando técnicas de procesamiento de imágenes e inteligencia artificial

Juan Pablo Gómez López
*Departamento de Ingeniería
de Sistemas*
Universidad de Antioquia
Medellín, Colombia
juan.gomez148@udea.edu.co

Jhon Jader Caro Sánchez
*Departamento de
Ingeniería de Sistemas*
Universidad de Antioquia
Medellín, Colombia
jader.caro@udea.edu.co

I. INTRODUCCIÓN

El grafeno, un material bidimensional compuesto por una única capa de átomos de carbono dispuestos en una estructura de panal [1], ha captado la atención de la comunidad científica debido a sus propiedades físicas excepcionales [2] y la promesa de sus diversas aplicaciones prácticas [3], [4], [5].

Sin embargo, la detección y caracterización del grafeno con pocas capas presentan varios desafíos. Los métodos convencionales, que incluyen la microscopía óptica y electrónica, y técnicas espectroscópicas, a menudo son costosos, consumen mucho tiempo y requieren equipo especializado para la detección manual de las regiones del material de interés. Estas limitaciones han generado un interés creciente en buscar alternativas más rápidas y accesibles para identificar y clasificar muestras de grafeno con alta precisión y eficiencia.

En este contexto, las técnicas de Deep Learning (DL) y procesamiento de imágenes (PI) han comenzado a revolucionar el campo de la ciencia de los materiales, ofreciendo soluciones innovadoras para superar las limitaciones de los métodos tradicionales [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], [16]. Estas técnicas, al aprovechar modelos de redes neuronales profundas, permiten analizar grandes volúmenes de datos visuales con rapidez y precisión, lo que facilita la detección y clasificación de materiales como el grafeno en distintas configuraciones y concentraciones.

Este trabajo se enfoca en explorar y aplicar métodos de DL y procesamiento de imágenes para mejorar la predicción y detección del grafeno. En particular, se desea segmentar imágenes en tres regiones de interés: la región del fondo o Background, las regiones que posiblemente se asocien a grafeno monocapa o bicapa, que en este trabajo se denomina Few Layer a dicha clase, y a las regiones que tienen muchas capas de grafeno se le denominan Bulk.

A través de este enfoque, se busca no solo lograr un avance en la caracterización del grafeno, sino también aportar

a la comprensión y aplicación de la inteligencia artificial en la investigación de nuevos materiales, contribuyendo al desarrollo de herramientas analíticas más precisas y eficientes.

II. METODOLOGIA

La base de datos original consta de 200 imágenes obtenidas mediante el microscopio Olympus BX51M, suministradas por el laboratorio de la Universidad Presbiteriana Mackenzie en Brasil. Estas muestras incluyen imágenes capturadas con un microscopio de alta potencia en varias resoluciones de zoom: x100, x50, x20, x10 y x5. Sin embargo, para este trabajo se decidió utilizar únicamente 40 imágenes capturadas con un nivel de zoom de x20, ya que este nivel ofrecía un equilibrio adecuado entre resolución y campo de visión, facilitando tanto la identificación de grafeno como la clasificación de su grosor.

Las muestras se segmentan en tres clases: el fondo de la imagen, Background, las capas delgadas de grafeno, llamadas Few Layer, y las regiones con muchas capas de grafeno, denominadas Bulk, como se mencionó en la sección anterior.

Las imágenes seleccionadas fueron sometidas a un etiquetado manual utilizando la herramienta Roboflow, definiendo regiones específicas de interés para cada clase. Posteriormente, los datos fueron divididos en conjuntos de entrenamiento, validación y prueba.

Durante el procesamiento inicial, se realizó un análisis exploratorio para evaluar la distribución de las clases y su comportamiento en diferentes canales de color, lo que permitió identificar patrones relevantes y posibles desbalances en los datos.

Para la tarea de segmentación semántica, se seleccionó un modelo basado en la arquitectura UNet, un modelo ampliamente utilizado en problemas de segmentación de imágenes biomédicas y científicas. Su diseño, en forma de "U", permite combinar información contextual global con detalles locales, lo que lo hace especialmente útil para tareas de

segmentación semántica como la detección de grafeno. Se exploraron configuraciones con diferentes profundidades del modelo, determinando cuál era la más adecuada en función de las características del conjunto de datos.

Además, se implementaron técnicas de aumento de imágenes y procesamiento de imágenes para aumentar la cantidad de datos disponibles y mejorar su calidad, generando variantes sintéticas de las imágenes originales. Dado que las clases estaban desbalanceadas, se priorizó el F1-score general como métrica principal de evaluación, ya que proporciona un balance entre la precisión y la sensibilidad, reflejando mejor el rendimiento del modelo en este contexto.

La implementación de los modelos y el procesamiento de las imágenes se llevaron a cabo utilizando Python y bibliotecas especializadas como TensorFlow y Sci-kit image. Finalmente, el modelo fue evaluado utilizando el conjunto de datos de prueba mediante las siguientes métricas de validación: Intersection over Union (IoU), el F1-score, accuracy (acc), precision (prec) y recall, validando así su capacidad para segmentar y clasificar correctamente las regiones de interés.

III. DATOS

Para obtener los datos necesarios para entrenar el modelo, solo es necesario ejecutar la celda que hace uso de la API de Roboflow, proporcionando el Api Key **YJeW9pVCy9nQ7BwHkRIO**. Este proceso se realiza dentro del **workspace** denominado **2d-materials-segmentation** y con la versión de descarga **4**. Al ejecutar esta celda, se obtienen las imágenes de 256x256 píxeles a color, junto con sus respectivas etiquetas, que fueron etiquetadas manualmente para indicar las áreas de interés en las imágenes. Los datos se organizarán en tres carpetas: **train**, **test** y **valid**. La carpeta **train** contendrá 32 imágenes para el entrenamiento, **test** incluirá 5 imágenes para la evaluación del modelo, y **valid** tendrá 3 imágenes para su validación. Este proceso asegura que los datos estén correctamente estructurados y listos para su uso en el entrenamiento del modelo de segmentación.

IV. DESCRIPCIÓN DE LA ESTRUCTURA DE NOTEBOOKS

Todos los notebooks del proyecto son completamente reproducibles con solo ejecutarlos y no requieren la subida ni el adjunto de ningún archivo externo.

En el notebook **1-Analisis-exploratorio.ipynb** realiza un análisis detallado del conjunto de datos, evaluando el desbalance de clases y analizando la distribución de colores en diferentes canales por clase, con el objetivo de identificar patrones relevantes que puedan influir en las etapas posteriores del preprocesamiento y entrenamiento del modelo.

Luego el notebook **2-Arquitectura-modelo.ipynb** se centra en el análisis de la profundidad del modelo de segmentación semántica, evaluando configuraciones con 64, 128, 256 y 512 neuronas por capa. En este notebook, se realizan entrenamientos y se analizan las métricas obtenidas tras 100 épocas, con el objetivo de determinar la configuración óptima que balancee la precisión del modelo y su capacidad de generalización.

En el notebook **3-Image Transformations + Augmentation Pipeline.ipynb** se desarrolla un pipeline para automatizar las transformaciones de las imágenes y la generación de datos sintéticos. Este proceso incluye técnicas como rotaciones verticales y horizontales, giros entre -15° y 15° , traspuestas y aplicación blur y contraste, con el objetivo de incrementar la diversidad del conjunto de datos y robustecer el modelo frente a variaciones en las imágenes.

Por último, en el notebook **4-UNet_128+Image Augmentation.ipynb** se aplica el data augmentation previamente mencionado y se entrena el modelo de segmentación semántica utilizando una configuración con 128 neuronas como profundidad máxima. En este notebook, se valida el rendimiento del modelo entrenado, evaluando su capacidad para segmentar correctamente las clases de interés y analizando los resultados obtenidos en comparación con las expectativas.

V. RESULTADOS

En esta sección se presentan los hallazgos obtenidos durante el análisis exploratorio de los datos, así como el desempeño del modelo seleccionado para la segmentación semántica. Se evalúa el impacto del procesamiento de datos y del aumento sintético de imágenes, destacando las métricas utilizadas para validar la efectividad del modelo, con énfasis en el F1-score general debido a la naturaleza desbalanceada de las clases. Adicionalmente, se analizan las limitaciones encontradas y las oportunidades de mejora para futuras implementaciones.

La Figura 1 muestra un esquema resumido del procedimiento seguido para la resolución del problema.

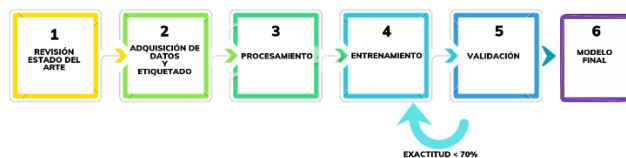


Figura 1. Distribución de clases a nivel de píxeles

A. Análisis exploratorio

El análisis inicial permitió identificar un desbalance significativo en la distribución de etiquetas. A nivel de regiones etiquetadas, el 84.5% correspondía a la clase *Bulk* y el 15.5% a la clase *Few Layer*. Sin embargo, al realizar un análisis a nivel de píxeles, el desbalance se torna aún más pronunciado: el 79.3% pertenece a la clase *Background* (fondo), el 20.3% a la clase *Bulk* y apenas el 0.4% a la clase *Few Layer*. como se muestra en la Figura 2. Este desbalance extremo motivó la priorización del *F1-score* como métrica principal de evaluación, ya que este indicador ofrece un balance adecuado entre precisión y sensibilidad en escenarios donde una clase es dominante.

En cuanto al análisis del espacio de color RGB, se observó que los tres canales presentaban un comportamiento muy similar, con diferencias insignificantes entre ellos. Esto sugiere que trabajar con imágenes en color o convertirlas a escala de grises no genera un cambio significativo en la calidad de la información visual utilizada para el entrenamiento del modelo. Por lo tanto, ambas opciones son igualmente válidas para este caso específico.

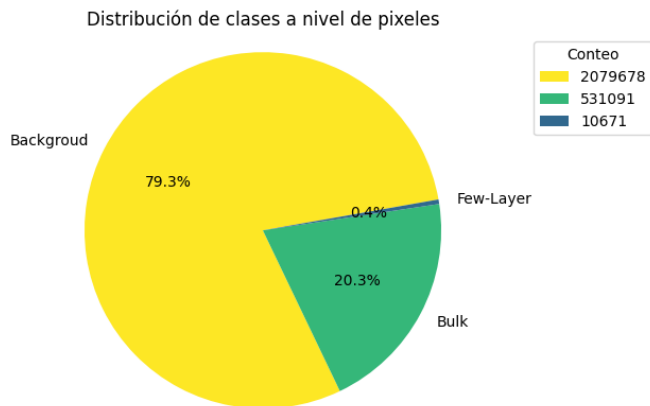


Figura 2. Distribución de clases a nivel de píxeles

B. Desempeño de la arquitectura

El modelo basado en la arquitectura *UNet* fue entrenado y evaluado utilizando las imágenes procesadas y aumentadas. Se probaron configuraciones con profundidades de 64, 128, 256 y 512 neuronas, donde cada modelo fue entrenado durante 100 épocas utilizando un conjunto de 32 imágenes para entrenamiento, 5 para prueba y 3 para validación. Entre las configuraciones evaluadas, el modelo con una profundidad de 128 mostró el mejor desempeño en la etapa de validación, alcanzando un *F1-score* general de 0.969. La Figura 3 se muestran los resultados de lo anteriormente expuesto.

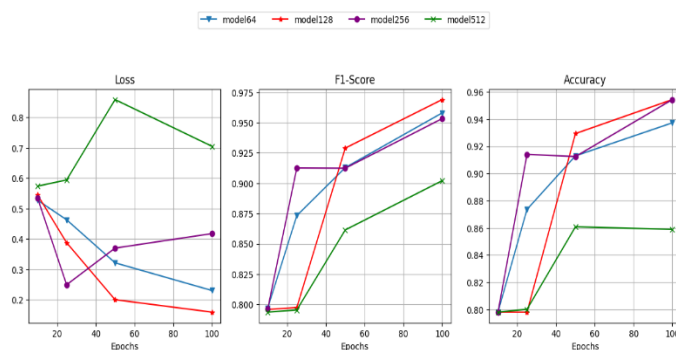


Figura 3. Desempeño de los modelos con diferentes profundidades en etapa de validación

Al analizar el rendimiento por clase en el mejor modelo, se observó que la clase *Bulk* obtuvo un *recall* de 0.62,

mientras que la clase *Few Layer* alcanzó un *recall* de 0.0 y finalmente la clase *Background* logró un *recall* de 0.97, como se muestra en la Figura 4.

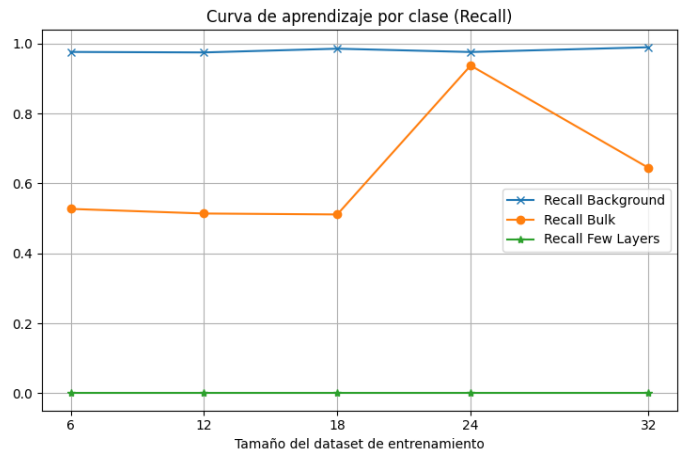


Figura 4. Curva de aprendizaje del modelo con 128 neuronas de profundidad evaluada para cada clase.

C. Aumento de datos

Adicionalmente, para fortalecer la capacidad de generalización del modelo, se aplicaron técnicas de aumento de datos inspiradas en métodos descritos en la literatura, como rotaciones, cambios de contraste y aplicación de métodos para difuminar ligeramente las imágenes. Estas estrategias incrementaron el conjunto de datos hasta alcanzar un volumen más adecuado para entrenar redes profundas. El análisis inicial del modelo entrenado destaca su eficacia en la identificación de estructuras *Few Layer* en condiciones experimentales controladas. Sin embargo, se requiere una mayor optimización del pipeline de preprocesamiento y la inclusión de más imágenes para mejorar la capacidad del modelo de segmentar estructuras complejas. Estos avances señalan la utilidad de *U-Net* como herramienta base y plantean el camino hacia desarrollos más robustos en el campo de la caracterización automatizada de nanomateriales.

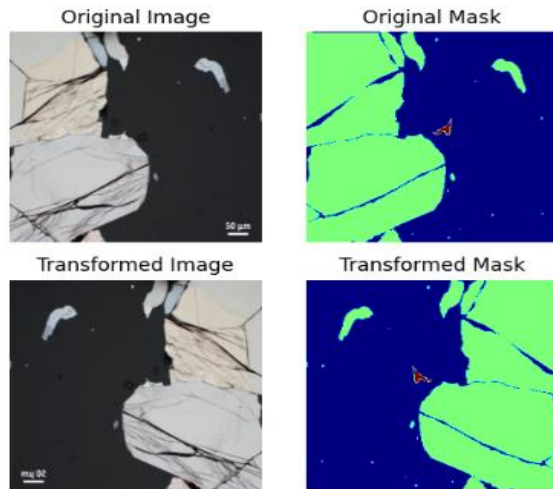


Figura 5. Muestra de una imagen origina y su máscara (Imágenes superiores), y una imagen generada con su respectiva máscara (dos imágenes inferiores).

Para la generación de imágenes, se implementó un pipeline de aumento de imágenes. Este proceso incluyó la creación de imágenes sintéticas mediante la aplicación de un filtro gaussiano combinado con la imagen original. Por cada imagen del conjunto de entrenamiento, se generaron 50 imágenes sintéticas, mientras que para el conjunto de prueba y el conjunto de validación se generaron 10 y 5 imágenes sintéticas por cada imagen original, respectivamente. Inicialmente, el conjunto de datos estaba compuesto por 32 imágenes de entrenamiento, 5 de prueba y 3 de validación. Después del aumento de datos, los conjuntos resultantes contenían 1632 imágenes para entrenamiento, 55 para prueba y 18 para validación.

A las máscaras de segmentación se les aplicó un proceso de codificación *one-hot encoding*. Sin embargo, debido a limitaciones de memoria RAM durante el entrenamiento, este proceso se ajustó para optimizar el rendimiento. Una vez preparado el conjunto de datos, las imágenes fueron normalizadas, mientras que las máscaras fueron codificadas. Se definió la arquitectura del modelo, se compiló y se creó una instancia de *EarlyStopping* para monitorizar la pérdida de validación. Finalmente, se definió la arquitectura del modelo, se compiló y se inició el proceso de entrenamiento.

D. Validación del modelo

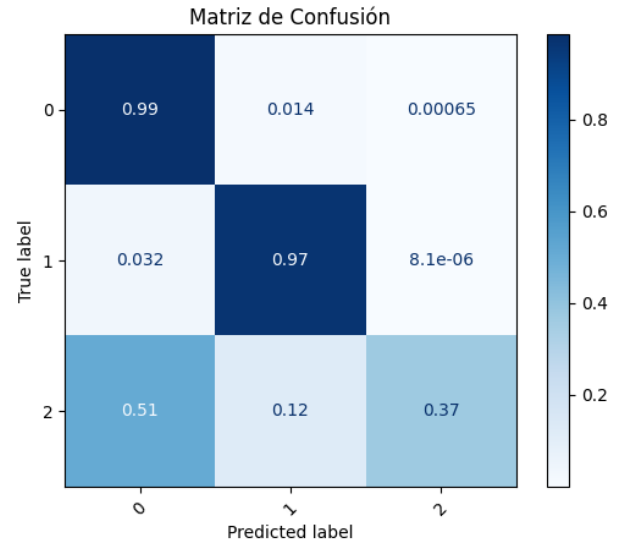


Figura 6. Matriz de confusión

Clase	Acc	Prec	Rec	F1 score	IoU
0	0.815	0.645	0.986	0.78	0.64
1	0.944	0.877	0.968	0.92	0.852
2	0.789	0.998	0.369	0.538	0.368
media	0.849	0.840	0.774	0.746	0.62
std	0.059	0.127	0.248	0.137	0.171

Tabla 1. Tabla resumen de métricas de validación del modelo. Acc significa exactitud, Prec es la precisión, Rec es el recall, IoU es la intersección sobre la unión, y std es la desviación estándar.

Los resultados del proceso de entrenamiento y validación del modelo se encuentran en la Figura 5, la matriz de confusión, y en la Tabla 1, exponiendo métricas de validación sobre un conjunto de datos que el modelo nunca ha visto. De la matriz de confusión se observa que el modelo tiene buena capacidad para detectar el fondo y la clase Bulk, pero la clase Few Layer tiene un desempeño mediocre, y se observa que se confunde fácilmente con el fondo. Esto se respalda con la Tabla 1, donde tanto en las métricas F1 score y IoU muestran que el modelo tiene dificultades diferenciando estas dos clases. son alentadores, ya que la diferencia en valores de píxeles entre el fondo y Few Layer al menos en escala de color RGB, es muy pequeña, y aun así, el modelo da indicios de ser capaz de identificar algunas regiones Few Layer reales.

La Figura 6 muestra un ejemplo de comparación entre la máscara real de la máscara predicha por el modelo. Se observa que el modelo se aproxima bien a la posición real de la región correspondiente a Few Layer. Y como muestran los

valores de la Figura 4 y la Tabla 1, confunde los pixeles de Few Layer con el Fondo.



Figura 7. Imagen para segmentar (izquierda). Máscara real (centro) y máscara predicha (derecha). Región en morado: Background. Región amarilla: Few Layer. Región azul aguamarina: Bulk.

Es importante resaltar que hubo casos donde no el modelo no fue capaz de detectar la clase Few Layer en lo absoluto, contribuyendo a los resultados expuestos en la Figura 5 y la Tabla 1. También se debe destacar que hubo limitación de recursos computacionales, ya que, si se generaban muchas imágenes sintéticas, no alcanzaba la memoria RAM para entrenar, el sistema colapsaba (se entrenó en Google Colab con una GPU T4, y el límite de RAM es de 12GB).

Finalmente, la Figura 7 muestra la evolución de la función de pérdida versus el número de épocas del modelo. Se observa un leve decrecimiento para el conjunto de entrenamiento a medida que aumentan las épocas. Sin embargo, para el conjunto de validación la pérdida permanece oscilando aproximadamente entre 0.1 y 0.125, con una ligera tendencia a empeorar a medida que aumentan las épocas. La Figura 8 muestra la pérdida a medida de que el proceso de entrenamiento del modelo avanza.

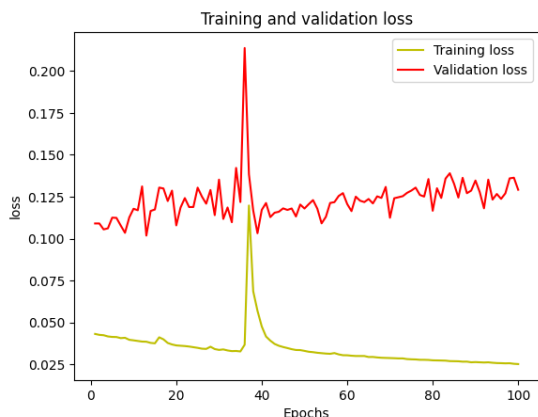


Figura 8. Curva de pérdida de entrenamiento del modelo, para conjunto de entrenamiento y de prueba.

VI. DISCUSIÓN

Los resultados obtenidos en este estudio demuestran tanto el potencial como las limitaciones actuales de la aplicación de técnicas de aprendizaje profundo para la segmentación semántica de nanomateriales. El desbalance significativo en la distribución de clases (79.3% Background, 20.3% Bulk y 0.4% Few Layer) y la dificultad para diferenciar

entre el fondo y la clase de interés representan las dificultades principales. Además, para este estudio, la limitación de recursos computacionales y la poca cantidad de imágenes influyeron en los resultados presentados. Sin embargo, los resultados son alentadores, pues la detección automática de las regiones Few Layer, por lo menos en su posición, está comenzando a aparecer.

Las estrategias de aumento de datos implementadas, incluyendo la generación de imágenes sintéticas mediante filtros gaussianos, demostraron ser fundamentales para incrementar el conjunto de datos de entrenamiento. No obstante, las limitaciones de recursos computacionales (12GB RAM) restringieron la escala del aumento de datos, lo que podría haber impactado en la capacidad del modelo para aprender características más sutiles de las regiones Few Layer.

Para trabajo futuro, se recomienda explorar arquitecturas más sofisticadas que puedan manejar mejor el desbalance de clases. Además, sería valioso investigar técnicas de preprocesamiento más avanzadas que puedan resaltar las diferencias entre las regiones Few Layer y el fondo. La exploración de funciones de pérdida personalizadas que tengan en cuenta el peso de las clases también podría mejorar significativamente el rendimiento en la detección de la clase minoritaria. Finalmente, por necesidades de negocio, también se podría plantear el enfoque a detección de objetos, ya que sólo se busca detectar posibles regiones como Few Layer, y no una segmentación precisa como se intentó en el presente artículo.

VII. CONCLUSIONES

Este estudio demuestra la viabilidad de utilizar redes neuronales convolucionales, específicamente la arquitectura U-Net, para la segmentación semántica de nanomateriales, aunque con limitaciones importantes en la detección de regiones Few Layer.

El modelo desarrollado muestra un rendimiento aceptable en la identificación de regiones del fondo y Bulk, como se muestra en la matriz de confusión de la Figura 5.

Las técnicas de aumento de datos implementadas, aunque limitadas por restricciones computacionales, demostraron ser cruciales para mejorar el rendimiento general del modelo.

La detección de regiones Few Layer continúa siendo un desafío significativo debido a su similitud con el fondo, por lo que un pretratamiento de imágenes para tratar de aumentar el contraste entre estas dos clases es recomendado.

Se debe resaltar que para aumentar la efectividad de los modelos de DL, se requiere suministrar una gran cantidad de datos (en este caso, imágenes), por lo que se debe aumentar la cantidad de imágenes para robustecer los modelos, y esto implica, además, una mayor capacidad computacional para entrenar estos modelos. A pesar de las limitaciones, el procedimiento empleado para la solución del problema planteado es bastante prometedor.

Finalmente, se debe resaltar que, para aumentar la efectividad de los modelos de DL, se requiere suministrar una gran cantidad de datos (en este caso, imágenes), por lo que se debe aumentar

la cantidad de imágenes para robustecer los modelos, y esto implica, además, una mayor capacidad computacional para entrenar estos modelos. A pesar de las limitaciones, el procedimiento empleado para la solución del problema planteado es bastante prometedor.

VIII. REPOSITORIO

El enlace a los notebooks de código se puede encontrar [aquí](#).

AGRADECIMIENTOS

Se agradece a Sergio Castrillón Salazar, estudiante egresado del instituto de Física, actualmente estudiante de Doctorado en la Universidad de Mackenzie, por suministrar las imágenes empleadas en este proyecto. También se agradece a Cristian David Gutiérrez Céspedes por sus asesorías en los temas de DL.

REFERENCIAS

- [1] A. Alqarni, "Characterization and Application of Graphene and Other Two-Dimensional Materials in Nanotechnology and Electronics," Apr. 01, 2024, *Research Square*. doi: 10.21203/rs.3.rs-4192806/v1.
- [2] S. Usha, M. Karunakaran, K. Kasirajan, and S. Prabakaran, "Two-dimensional materials and their applications," in *Futuristic Trends in Chemical Material Sciences & Nano Technology Volume 3 Book 23*, First., Iterative International Publishers, Selfypage Developers Pvt Ltd, 2024, pp. 244–251. doi: 10.58532/V3BECS23P4CH2.
- [3] P. Kumbhakar *et al.*, "Prospective applications of two-dimensional materials beyond laboratory frontiers: A review," *iScience*, vol. 26, no. 5, p. 106671, May 2023, doi: 10.1016/j.isci.2023.106671.
- [4] T. Wei, Z. Han, X. Zhong, Q. Xiao, T. Liu, and D. Xiang, "Two dimensional semiconducting materials for ultimately scaled transistors," *iScience*, vol. 25, no. 10, p. 105160, Oct. 2022, doi: 10.1016/j.isci.2022.105160.
- [5] S. Tufail *et al.*, "2D nanostructures: Potential in diagnosis and treatment of Alzheimer's disease," *Biomed. Pharmacother.*, vol. 170, p. 116070, Jan. 2024, doi: 10.1016/j.biopha.2023.116070.
- [6] J. Sanchez-Juarez, M. Granados-Baez, A. A. Aguilar-Lasserre, and J. Cardenas, "Automated system for the detection of 2D materials using digital image processing and deep learning," *Opt. Mater. Express*, vol. 12, no. 5, p. 1856, May 2022, doi: 10.1364/OME.454314.
- [7] S. Masubuchi *et al.*, "Deep-learning-based image segmentation integrated with optical microscopy for automatically searching for two-dimensional materials," *Npj 2D Mater. Appl.*, vol. 4, no. 1, p. 3, Mar. 2020, doi: 10.1038/s41699-020-0137-z.
- [8] Y. Qi, D. Hu, Z. Wu, M. Zheng, G. Cheng, and Y. P. Chen, "Deep Learning Assisted Raman Spectroscopy for Rapid Identification of 2D Materials".
- [9] J. Yang and H. Yao, "Automated identification and characterization of two-dimensional materials via machine learning-based processing of optical microscope images," *Extreme Mech. Lett.*, vol. 39, p. 100771, Sep. 2020, doi: 10.1016/j.eml.2020.100771.
- [10] Y. Saito *et al.*, "Deep-learning-based quality filtering of mechanically exfoliated 2D crystals," *Npj Comput. Mater.*, vol. 5, no. 1, p. 124, Dec. 2019, doi: 10.1038/s41524-019-0262-4.
- [11] B. Han *et al.*, "Deep-Learning-Enabled Fast Optical Identification and Characterization of 2D Materials," *Adv. Mater.*, vol. 32, no. 29, p. 2000953, Jul. 2020, doi: 10.1002/adma.202000953.
- [12] S. Qiao, L.-C. Chen, and A. Yuille, "DetectoRS: Detecting Objects with Recursive Feature Pyramid and Switchable Atrous Convolution," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2021, pp. 10208–10219. doi: 10.1109/CVPR46437.2021.01008.
- [13] E. Greplova *et al.*, "Fully Automated Identification of Two-Dimensional Material Samples," *Phys. Rev. Appl.*, vol. 13, no. 6, p. 064017, Jun. 2020, doi: 10.1103/PhysRevApplied.13.064017.
- [14] X. Lin *et al.*, "Intelligent identification of two-dimensional nanostructures by machine-learning optical microscopy," *Nano Res.*, vol. 11, no. 12, pp. 6316–6324, Dec. 2018, doi: 10.1007/s12274-018-2155-0.
- [15] P. A. Leger, A. Ramesh, T. Ulloa, and Y. Wu, "Machine-Learning-Enabled Fast Optical Identification and Characterization of 2D Materials," Jun. 23, 2024, *arXiv: arXiv:2406.16211*. Accessed: Aug. 12, 2024. [Online]. Available: <http://arxiv.org/abs/2406.16211>
- [16] L. Zichi, T. Liu, E. Drueke, L. Zhao, and G. Xu, "Physically informed machine-learning algorithms for the identification of two-dimensional atomic crystals," *Sci. Rep.*, vol. 13, no. 1, p. 6143, Apr. 2023, doi: 10.1038/s41598-023-33298-6.