IDENTIFICACIÓN AUTOMÁTICA DE RIESGO DE CÁNCER DE CUELLO UTERINO APLICANDO DEEP LEARNING EN IMÁGENES DE COLPOSCOPIA

Cesar Fuentes Esparza 8993331

Julián Alexis Correa Bustamante 0070408

Julián Correa Romero 066195

Proyecto Aplicado para optar al título de

Magister en Ciencia de Datos

Director

Hernán Darío Vargas Cardona

FACULTAD DE INGENIERÍA Y CIENCIAS

MAESTRÍA EN CIENCIA DE DATOS

SANTIAGO DE CALI, MAYO 19 DE 2025

TABLA DE CONTENIDO

LISTA DE FIGURAS

LISTA DE TABLAS

INTRODUCCIÓN

El cáncer de cuello uterino es una de las principales causas de muerte por cáncer entre las mujeres a nivel mundial, especialmente en países en vías de desarrollo donde el acceso a servicios de salud preventiva es limitado. La detección temprana de lesiones precancerosas mediante exámenes de colposcopia es esencial para la prevención y el tratamiento oportuno de esta enfermedad. Sin embargo, la interpretación de las imágenes de colposcopia es un proceso complejo que requiere de la experiencia y habilidad de especialistas, lo que puede llevar a variaciones en los diagnósticos y potencialmente a diagnósticos tardíos o erróneos.

Las técnicas de deep learning han emergido como una solución prometedora para la automatización y mejora de la precisión en el análisis de imágenes médicas. Los algoritmos de redes neuronales convolucionales (CNN) en particular han demostrado una capacidad notable para identificar patrones complejos en imágenes, superando en muchos casos la precisión de los métodos tradicionales y la observación humana.

Se implementó un modelo de deep learning para clasificar automáticamente imágenes de colposcopia en dos categorías: normales y con patología de cáncer de cuello uterino. La automatización de este proceso no solo tiene el potencial de aumentar la precisión diagnóstica, sino también de reducir la carga de trabajo de los especialistas y hacer que la detección sea más accesible en áreas con recursos limitados.

Para lograr esto, se gestionó una base de datos con imágenes de Colposcopia debidamente etiquetadas, se entrenaron algoritmos de deep learning en Python para la clasificación de las imágenes en normales o patológicas, y se validaron los modelos mediante la evaluación de métricas de clasificación como Exactitud (Accuracy), Sensibilidad, Especificidad, F1-score y AUC-ROC. Lo anterior, permitió desarrollar un sistema que apoya el proceso de tamización del cáncer de cuello uterino.

Este proyecto no solo contribuye al avance del conocimiento en el campo de la inteligencia artificial aplicada a la medicina, sino que también tiene el potencial de tener un impacto significativo en la salud pública, aportando a la detección temprana y el tratamiento del cáncer cervical.

1. DEFINICIÓN DEL PROBLEMA

1.1. Planteamiento del problema

El cáncer de cuello uterino, también conocido como cáncer cervicouterino, se origina en las células del cuello uterino, la porción inferior y estrecha del útero que conecta con la vagina [1]. A nivel mundial, representa un importante problema de salud pública, siendo una de las principales causas de mortalidad por cáncer en mujeres en diversas regiones [2]. Aunque este tipo de cáncer suele desarrollarse lentamente, lo que ofrece una ventana de oportunidad para su detección temprana, su identificación temprana resulta fundamental para mejorar significativamente las tasas de supervivencia y la efectividad del tratamiento [3]. Esta necesidad es aún más crítica en países de ingresos bajos y medios, donde las limitaciones en cobertura de programas de tamizaje y la escasez de personal especializado dificultan la detección temprana [4].

Actualmente los métodos más utilizados para la detección incluyen la prueba de Papanicolaou (Pap), la detección del virus del papiloma humano (VPH) y la colposcopia. Sin embargo, estos procedimientos enfrentan desafíos importantes, como la ocurrencia de falsos negativos, la subjetividad en la interpretación y la alta dependencia de profesionales altamente entrenados [3]. En particular, la interpretación de los hallazgos citológicos y colposcópicos, basados en la observación visual del cuello uterino tras la aplicación de ácido acético, depende en gran medida de la experiencia del examinador, lo que puede generar una considerable variabilidad interobservador [5].

El *deep learning* ha emergido como una herramienta con un potencial considerable para transformar el diagnóstico médico, especialmente en el campo de la ginecología oncológica [6]. En particular, su aplicación al análisis automatizado de imágenes de colposcopia busca mejorar la precisión y eficiencia tanto en la detección temprana del cáncer de cuello uterino como de la neoplasia intraepitelial cervical (NIC), actuando como un sistema de apoyo clínico para los profesionales de la salud [7][8]. El objetivo principal es complementar la pericia humana, reducir la subjetividad inherente a la interpretación visual y agilizar el proceso diagnóstico.

No obstante, el desarrollo de modelos de *deep learning* en este dominio enfrenta múltiples desafíos técnicos. Entre ellos se destacan: la escasez de datos etiquetados, la variabilidad en calidad de las imágenes, el desbalance de clases, la presencia de artefactos visuales (como reflejos especulares e instrumental médico) y la alta dimensionalidad de las representaciones extraídas [9][10]. Abordar estos retos requieren soluciones específicas desde la ciencia de datos, como la integración de fuentes heterogéneas, la aplicación de técnicas de balanceo de clases, la reducción de dimensionalidad y el diseño de pipelines modulares con validación cruzada y ajuste de hiperparámetros.

En este contexto, la presente investigación busca evaluar la viabilidad de un sistema automatizado para identificar el riesgo de cáncer de cuello uterino a partir de imágenes de colposcopia, empleando modelos de *deep learning* con énfasis en la interpretabilidad, la reducción de sesgos técnicos y la validación mediante métricas estandarizadas y análisis retrospectivos sobre datos reales. Si bien no se realizó una validación clínica directa, se incorporaron

enfoques orientados a mejorar la confiabilidad del modelo en escenarios médicos, priorizando la minimización de errores críticos y la utilidad potencial como herramienta de apoyo diagnóstico.

1.2. Formulación del problema

La formulación del problema en esta investigación se centra en la viabilidad y efectividad de los métodos de deep learning en la detección temprana de NIC (riesgo de cáncer) y cáncer de cuello uterino. Dado lo anterior se plantea la siguiente pregunta de investigación: ¿Cómo se pueden aplicar técnicas de deep learning para clasificar imágenes de Colposcopias como normales o con riesgo de cáncer de cuello uterino?

1.3. Preguntas de investigación.

Las preguntas específicas que guiarán esta investigación son:

1. ¿Cuáles son las estrategias más efectivas para gestionar y etiquetar una base de datos con imágenes de colposcopia de manera que puedan ser utilizadas en modelos de deep learning?
2. ¿Cómo se pueden identificar y seleccionar las técnicas o arquitecturas de deep learning más adecuadas para la clasificación de imágenes de colposcopias en dos grupos: normales y con riesgo de cáncer de cuello uterino?
3. ¿Cómo se pueden diseñar y aplicar métodos de evaluación y validación para determinar la efectividad de los modelos de deep learning en la clasificación de imágenes de colposcopias?

1.4. Justificación

El cáncer de cuello uterino representa una de las principales causas de mortalidad por cáncer en mujeres a nivel mundial. Según datos de la Organización Mundial de la Salud (OMS), en 2020 se estimaron aproximadamente 604,127 nuevos casos y 341,831 muertes, con una tasa de incidencia ajustada por edad de 13.3 por cada 100,000 mujeres-año [11]. Esta carga afecta de forma desproporcionada a los países de ingresos bajos y medianos, donde la incidencia puede triplicarse respecto a los países con muy alto índice de desarrollo humano.

De no implementarse intervenciones eficaces como la vacunación contra el virus del papiloma humano (VPH) y programas de tamizaje con cobertura universal, se proyecta que entre 2020 y 2069 se diagnosticarán más de 44 millones de casos de cáncer de cuello uterino, de los cuales dos terceras partes ocurrirán en regiones con desarrollo humano bajo o medio[12]. Estas cifras resaltan la urgencia de diseñar soluciones innovadoras, accesibles y escalables que permitan mejorar la detección temprana y reducir la carga global de esta enfermedad.

Frente a este panorama, el desarrollo de herramientas automatizadas basadas en inteligencia artificial, como las que se proponen en este trabajo, puede contribuir significativamente al fortalecimiento de los sistemas de salud pública, especialmente en entornos con recursos limitados. Estas tecnologías tienen el potencial de funcionar como sistemas de apoyo al diagnóstico clínico, mejorando la precisión, reduciendo la subjetividad en la interpretación de exámenes

colposcópicos y facilitando la detección oportuna del cáncer cervical.

Este proyecto propone una arquitectura robusta, con énfasis en la transparencia, la reducción de errores críticos (especialmente falsos negativos) y utilidad clínica, abordando de manera sistemática los principales desafíos reportados en el estado del arte. Su enfoque modular y validado retrospectivamente permite ser replicable, auditable y potencialmente escalable a escenarios reales, contribuyendo tanto al avance de la investigación biomédica como al fortalecimiento de los programas de tamizaje poblacional.

Por tanto, este proyecto se justifica en función de los siguientes aspectos:

- Explorar soluciones automatizadas que reduzcan la subjetividad diagnóstica mediante la aplicación de modelos robustos de deep learning.
- Superar las limitaciones técnicas identificadas en estudios previos, mediante un pipeline completo que abarca preprocesamiento, extracción de características, balanceo de clases, validación cruzada, búsqueda de hiperparámetros y análisis retrospectivos.
- Democratizar el acceso al diagnóstico temprano, habilitando herramientas que puedan ser utilizadas en contextos con escaso recurso humano especializado.
- Promover la interpretabilidad del modelo y el análisis detallado de errores, particularmente en la identificación de falsos negativos, es esencial en contextos clínicos donde las consecuencias de diagnósticos omitidos pueden ser críticas. Aunque muchas investigaciones priorizan métricas agregadas como el AUC o el F1-score, estudios recientes destacan la necesidad de enfoques que permitan comprender y explicar las decisiones del modelo, así como evaluar su desempeño en subgrupos clínicamente relevantes [13][14]. Esta perspectiva busca complementar los enfoques más comunes y aportar a la confiabilidad del modelo en escenarios reales.

2. OBJETIVOS

2.1. Objetivo general

Entrenar modelos de deep learning para clasificar automáticamente imágenes de colposcopias en dos grupos: Con y Sin riesgo de cáncer de cuello uterino para dar soporte al proceso de tamizaje de la enfermedad.

2.2. Objetivos específicos

- Gestionar una base de datos con imágenes de colposcopia debidamente etiquetadas en dos clases: normal y patológica.
- Entrenar algoritmos de deep learning en lenguaje Python que permitan la clasificación de imágenes de colposcopia en dos grupos con o sin riesgo de cáncer de cuello uterino.
- Validar los métodos de deep learning mediante la evaluación basada en métricas de clasificación como exactitud, sensibilidad, especificidad, F1-score, y AUC-ROC.

3. MARCO TEÓRICO Y ANTECEDENTES

Se presentan los conceptos teóricos relacionados con el desarrollo del proyecto, teniendo como enfoque la identificación de imágenes, siguiendo los pasos de un proyecto de ciencias de datos.

3.1. Cáncer de cuello uterino y pruebas de tamizaje.

El cáncer de cuello uterino es una proliferación de células que comienza en el cuello del útero. El cuello del útero es la parte inferior del útero que se conecta a la vagina.

Varias cepas del virus del papiloma humano juegan un papel importante en la causa de la mayoría de los tipos de cáncer del cuello del útero. El virus del papiloma humano es una infección frecuente que se transmite por contacto sexual "Fig 1.". Cuando se expone al virus del papiloma humano, el sistema inmunitario del cuerpo generalmente evita que el virus haga daño. Sin embargo, en un pequeño porcentaje de personas, el virus sobrevive durante años. Esto contribuye al proceso que hace que algunas células del cuello del útero se conviertan en células cancerosas [15].

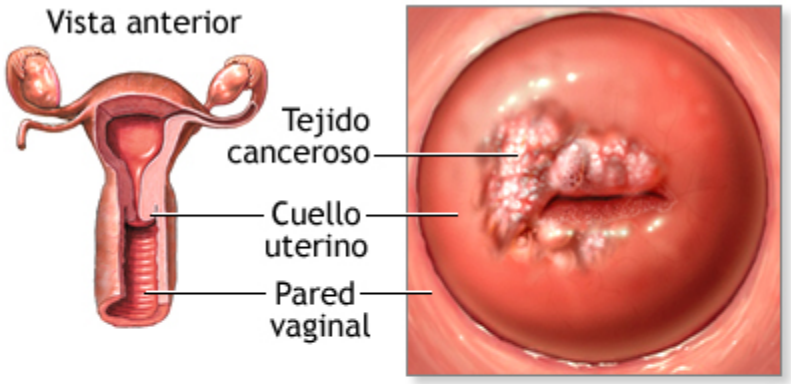


Fig 1. Cuello uterino y lesión cervical.
Nota: tomado de "https://medlineplus.gov/spanish/ency/article/000893.htm"

El tamizaje es fundamental para la detección temprana del cáncer de cuello uterino. Las pruebas de tamizaje como se observa en la figura 2, como la prueba de Papanicolaou (Pap) y las pruebas de detección del VPH, permiten identificar cambios celulares precancerosos antes de que evolucionen a cáncer. En muchos casos, el tratamiento temprano de estas lesiones puede prevenir el desarrollo del cáncer invasivo [15] [3].

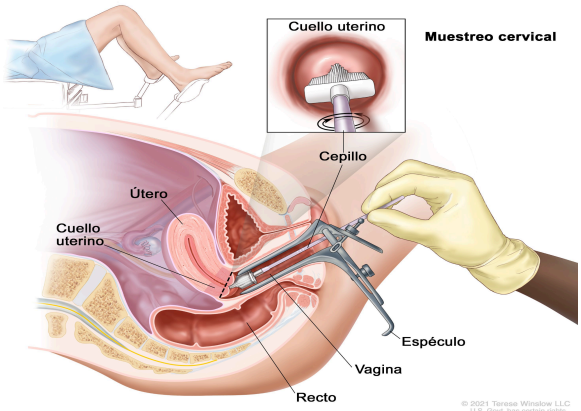


Fig 2. Ilustración del tamizaje.
Nota: tomado de "https://www.cancer.gov/espanol/tipos/cuello-uterino/deteccion"

3.2. Colposcopia e interpretación

Una colposcopia es un tipo de prueba para detectar el cáncer de cuello uterino (que es un tipo de cáncer de los órganos reproductivos). Utilizando un equipo de colposcopia como se observa en la figura 3. Permite que un doctor(a) o enfermera(o) pueda ver muy de cerca el cuello uterino, que es la entrada al útero. La colposcopia se usa para detectar células anormales en el cuello uterino [16].

La interpretación de los hallazgos colposcópicos requiere experiencia y conocimiento de los patrones normales y anormales del epitelio cervical. Los hallazgos comunes incluyen:

- Epitelio acetoblanco: Áreas que se vuelven blancas tras la aplicación de ácido acético, lo que puede indicar displasia.
- Punteado y mosaico: Patrones vasculares que sugieren cambios precancerosos o cancerosos.
- Lesiones sospechosas: Áreas con bordes irregulares, ulceraciones o vascularización anormal que requieren biopsia inmediata [17][18].

Fig 3. Equipo de colposcopia
Nota: tomado de https://susanaperez.es/que-es-la-prueba-de-la-colposcopia/

Tabla 1. Clasificación colposcópica de la International Federation of Cervical Pathology and Colposcopy [41]

Terminología colposcópica del cuello uterino de IFCPC 2011[1]			
Evaluación General		• Adecuada/ inadecuada a causa de... (por ej: cuello uterino no claro por inflamación, sangrado, cicatriz) • Visibilidad de la unión escamocolumnar: completamente visible, parcialmente visble, no visible Tipos de zona de transformación 1,2,3	
Hallazgos colposcópicos normales		Epitelio escamoso original: • Maduro • Atrófico Epitelio columnar • Ectopía Epitelio escamoso metaplásico • Quistes de Naboth • Aberturas glandulares y/o criptas glandulares Deciduosis en el embarazo	
Hallazgos colposcópicos anormales	**Principios generales**	**Ubicación de la lesión:** dentro o fuera de la zona de Transformación, ubicación de la lesión según las agujas del reloj **Tamaño de la lesión** Número de cuadrantes del cuello uterino que cubre la lesión, tamaño de la lesión en porcentajes del cuello uterino	
	Grado 1 (Menor)	Epitelio acetoblanco delgado. Borde irregular	Mosaico fino, Puntillado fino
	Grado 2 (Mayor)	Epitelio acetoblanco denso, Aparición rápida de epitelio acetoblanco. Orificios glandulares abiertos con bordes engrosados	Mosaico grueso, Puntillado grueso. Bordes delimitados, Signo del límite del borde interno, Signo de cresta o sobreelevado
	No específicos	Leucoplasia (queratosis, hiperqueratosis), Erosión Solución de Lugol (Test de Schiller): positivo/negativo	
Sospecha de invasión		Vasos atípicos Signos adicionales: Vasos delgados, superficie irregular, lesión exofítica, necrosis, ulceración (necrótica), tumoración nodular.	
Vasos atípicos		Zona de transformación congénita, Condiloma, Pólipo (exocervical / endocervical) Inflamación	Estenosis, Anomalía congénita, Anomalías post tratamiento, Endometriosis

La Clasificación Colposcópica más aceptada internacionalmente es la que propone la IFCPC.

La última actualización, elaborada por un comité constituido por 13 colposcopistas representantes de las diferentes Sociedades Científicas a nivel mundial, se presentó en Río de Janeiro en 2011 (Tabla 1) [19].

Esta clasificación, respecto a la previa, introdujo como principales novedades:

1. El concepto de exploración adecuada (sustituyendo el concepto clásico de colposcopia satisfactoria)
2. La descripción de la lesión en cuanto a tamaño, localización y ubicación con respecto a la zona de

transformación.

3. Incorpora 2 nuevos signos en el apartado de los cambios grado 2 (el signo del borde interno o blanco sobre blanco “inner border sign” y el signo de la cresta “ridge sign”).
4. Incorporó la clasificación y terminología para las lesiones de la vagina.

La tecnología empleada para la evaluación colposcópica se considera un factor que tiene una repercusión considerable. En la actualidad, con equipos de alta definición para colposcopia, es posible obtener imágenes de gran calidad que permiten mayor exactitud diagnóstica. Hay estudios recientes que utilizando colposcopios de última generación, obtienen una muy buena correlación entre la clasificación colposcópica y el resultado de biopsia (kappa 45,8% vs 74,1%) [20].

3.3. Neoplasia intraepitelial cervical (NIC)

La neoplasia intracervical, también conocida como neoplasia intraepitelial cervical (NIC), se refiere a la presencia de células anormales en el epitelio del cuello uterino. Estas células pueden variar en su grado de displasia, desde leves hasta severas, y su clasificación es crucial para determinar el riesgo y el tratamiento adecuado. La clasificación se realiza comúnmente mediante exámenes histológicos y citológicos, categorizando las lesiones en grados como NIC 1, NIC 2 y NIC 3, según la severidad y el potencial de progresión del cáncer cervical (Figura 4) [3].

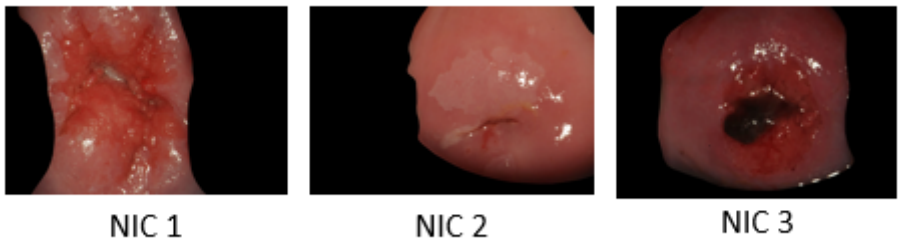


Fig 4. Ejemplo clasificación tipos de NIC

3.4. Técnicas de clasificación de imágenes médicas

Los métodos de detección para la neoplasia intracervical incluyen una variedad de técnicas tanto invasivas como no invasivas.

3.4.1. Métodos Tradicionales de Detección

Los métodos tradicionales de detección del cáncer de cuello uterino incluyen:

- Prueba de Papanicolaou (Pap): Detecta cambios celulares anormales en el cuello uterino. Es una herramienta clave en los programas de tamizaje.
- Prueba del VPH: Identifica la presencia de tipos de VPH de alto riesgo asociados con el desarrollo de cáncer

cervical.

- Colposcopia: Permite una visualización detallada del cuello uterino para identificar áreas sospechosas.
- Biopsia dirigida: Se utiliza para confirmar el diagnóstico en caso de hallazgos anormales [3][4].

3.4.2 Métodos Avanzados de Detección

Con el avance de la tecnología, se han desarrollado métodos más precisos y accesibles para la detección del cáncer cervical, entre ellos:

- Pruebas de ADN del VPH: Detectan directamente el material genético del virus en las células cervicales, lo que permite una identificación más precisa.
- Citología en medio líquido: Mejora la calidad de las muestras y reduce los errores de interpretación.
- Técnicas de imagen asistidas por inteligencia artificial: Utilizan algoritmos de aprendizaje profundo para analizar imágenes de colposcopia y citología, mejorando la precisión diagnóstica y reduciendo la subjetividad [21].

3.5. Deep learning y redes convolucionales

Las redes neuronales convolucionales (CNN) son la arquitectura más utilizada en el análisis de imágenes médicas. Una CNN estándar está compuesta por capas convolucionales, que extraen características espaciales locales, capas de pooling, que reducen la dimensionalidad y ayudan a controlar el sobreajuste, y capas densas, que realizan la clasificación final “Figura 5” [22][23]. En el ámbito de la medicina, las CNN han demostrado un desempeño sobresaliente en tareas de clasificación, segmentación y detección de anomalías en imágenes [24].

La arquitectura de una CNN consiste en múltiples capas, incluyendo[25]:

- Capas de Convolución (CONV): son fundamentales para la extracción de características de las imágenes de entrada [26]. Funcionan aplicando una serie de filtros (kernels) que se deslizan sobre la imagen, realizando una operación matemática llamada convolución. Cada filtro está diseñado para detectar un tipo específico de característica, como bordes, esquinas o texturas. La aplicación de estos filtros genera mapas de características que indican dónde se localizan esas características en la imagen. A medida que la información pasa por múltiples capas convolucionales, la red puede detectar características cada vez más complejas [25][27].
- Capas de Pooling (POOL): Estas capas se insertan generalmente entre las capas convolucionales y su función es reducir el tamaño de los mapas de características, disminuyendo así la cantidad de parámetros y la carga computacional. El Max-Pooling es una operación de pooling común que selecciona el valor máximo dentro de una ventana de filtro, preservando las características más importantes [26].
- Capas de Activación: Después de cada capa convolucional, se aplica una función de activación, que introduce no linealidad en la red, permitiéndole aprender relaciones complejas en los datos, una función de activación común es ReLU (Rectified Linear Units), que sustituye todos los valores negativos por ceros [26].

- Capas Densely Connected (FC) o Capas de Clasificación: Al final de la fase de extracción de características, se encuentran las capas totalmente conectadas. Estas capas toman como entrada un vector de características aplanado de las capas precedentes y aplican una combinación lineal, seguida de una función de activación para clasificar la imagen en las diferentes categorías [25]. La capa de salida utiliza una función de activación como Sigmoide (para clasificación binaria) o Softmax (para clasificación multiclase) para producir las probabilidades de pertenencia a cada clase [28].

El aprendizaje por transferencia (TL) es una técnica de aprendizaje automático donde el conocimiento aprendido de una tarea se utiliza para mejorar el rendimiento en una tarea relacionada. En el contexto de la clasificación de imágenes, implica tomar arquitecturas de redes neuronales que han sido previamente entrenadas en grandes conjuntos de datos, como ImageNet (que contiene millones de imágenes etiquetadas con miles de categorías), y adaptarlas para una nueva tarea con un conjunto de datos más pequeño[27].

Las fuentes mencionan el uso de varias arquitecturas de estado del arte pre-entrenadas [27]:

- DenseNet: Esta arquitectura se caracteriza por conectar cada capa con todas las capas subsiguientes de la red, lo que permite una reutilización eficiente de características y ayuda a mitigar el problema del desvanecimiento del gradiente.
- ResNet (Residual Network): ResNet utiliza conexiones residuales que permiten que la información fluya directamente a través de las capas, facilitando el entrenamiento de redes muy profundas y abordando también el problema del desvanecimiento del gradiente.
- EfficientNet: Esta familia de redes neuronales se enfoca en escalar eficientemente las dimensiones de la red, utilizando un método de ajuste compuesto, buscando un equilibrio óptimo entre precisión y eficiencia en el uso de recursos computacionales. Como arquitectura base mejora el rendimiento y la eficiencia con respecto a otras redes neuronales. EfficientNetV2 es una evolución de esta familia.
- Inception (GoogLeNet) e InceptionV3: Estas arquitecturas se caracterizan por usar módulos Inception que permiten a la red elegir entre diferentes tamaños de filtros convolucionales en cada bloque, lo que la hace eficiente y capaz de capturar características a diferentes escalas. InceptionV3, una versión mejorada, introduce técnicas adicionales como dividir convoluciones y el uso de conexiones auxiliares, lo que la hace aún más eficiente y capaz de capturar características a diferentes escalas, mejorando el rendimiento en tareas de clasificación de imágenes.

El proceso de adaptación de un modelo pre-entrenado a una nueva tarea se conoce como fine-tuning. Esto implica congelar las capas iniciales del modelo pre-entrenado (que han aprendido características generales como bordes y texturas) y entrenar las capas finales en el nuevo conjunto de datos específico. A veces, también se descongelan y se ajustan algunas de las capas intermedias con una tasa de aprendizaje menor para adaptar las características aprendidas al nuevo dominio [27].

El uso de transfer learning ofrece varias ventajas, incluyendo la reducción del tiempo de entrenamiento, la necesidad de menores cantidades de datos etiquetados y, a menudo, un mejor rendimiento en la tarea final, ya que el modelo se beneficia de las características aprendidas previamente en un conjunto de datos mucho más grande[27].

Clasificadores Utilizados:

Después de extraer las características de las imágenes utilizando las capas convolucionales (ya sea de un modelo propio o de un modelo pre-entrenado), se utiliza un clasificador para asignar una etiqueta a cada imagen basándose en esas características [27]:

- SVM (Support Vector Machine): Es un algoritmo de aprendizaje supervisado que busca encontrar el hiperplano óptimo que mejor separa los datos de diferentes clases en un espacio de alta dimensión [26].
- Random Forest: es un algoritmo de ensemble learning que construye múltiples árboles de decisión durante el entrenamiento y luego promedia o vota las predicciones de los árboles individuales para obtener una predicción final más robusta y precisa.
- KNN (k-Nearest Neighbors): es un algoritmo de aprendizaje supervisado no paramétrico que clasifica una nueva muestra basándose en la clase mayoritaria de sus K vecinos más cercanos en el espacio de características [26].
- XGBoost (Extreme Gradient Boosting): es otro algoritmo de ensemble learning basado en el boosting de árboles de decisión. Implementando optimizaciones a nivel de algoritmo y sistema que lo hacen altamente eficiente y efectivo.
- Fully-connected: se refiere a una red neuronal multicapa (MLP) que se utiliza como clasificador. Después de aplanar las características extraídas por las capas convolucionales, estas se alimentan a una o varias capas totalmente conectadas para realizar la clasificación final.

La elección del clasificador puede influir en el rendimiento del sistema de clasificación de imágenes, y a menudo se experimenta con diferentes clasificadores para encontrar el que mejor se adapte a las características extraídas y al conjunto de datos específico [26].

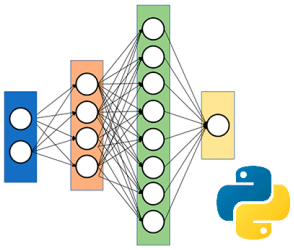


Fig 5. Diagrama red CNN
Nota: tomado de https://anderfernandez.com/blog/como-programar-una-red-neuronal-desde-0-en-python/

3.6. Aplicación de redes convolucionales densamente conectadas en imágenes de colposcopia

Las redes convolucionales densamente conectadas (DenseNets) han demostrado ser especialmente eficaces en el análisis de imágenes médicas, incluyendo las imágenes de colposcopia utilizadas para la detección de lesiones precancerosas y cáncer de cuello uterino. La capacidad de DenseNet para reutilizar características y facilitar el flujo de información entre capas permite identificar patrones complejos y sutiles que pueden pasar desapercibidos para el ojo humano o para modelos menos sofisticados [29][30].

En el contexto de la colposcopia, las imágenes presentan una gran variabilidad en cuanto a iluminación, textura y presencia de artefactos. DenseNet, al aprovechar la información de todas las capas previas, puede aprender representaciones más robustas y discriminatorias, lo que mejora la precisión en la clasificación de imágenes como normales o patológicas [31].

3.6.1. Beneficios de DenseNet en colposcopia

- Mejor discriminación de lesiones: DenseNet puede identificar diferencias sutiles entre tejido sano y tejido con lesiones precancerosas o cancerosas, lo que es fundamental para el diagnóstico temprano.
- Reducción de errores humanos: La automatización del análisis de imágenes de colposcopia mediante DenseNet puede disminuir la subjetividad y la variabilidad entre observadores, apoyando a los especialistas en la toma de decisiones clínicas.
- Eficiencia computacional: DenseNet requiere menos parámetros y recursos computacionales en comparación con otras arquitecturas profundas, facilitando su implementación en entornos clínicos con recursos limitados [29][31].

3.6. Revisión de trabajos previos relevantes

Se realizó una revisión de literatura referente a la detección y clasificación de cáncer de cuello uterino, que cumplieran con los siguientes criterios, uso de redes convolucionales y clasificación de imágenes.

- Automated Diagnosis of Cervical Intraepithelial Neoplasia in Histology Images via Deep Learning: Se desarrollaron, entrenaron y validaron modelos de deep learning para la clasificación automática de imágenes histológicas en la detección de la Neoplasia Cervical Intraepitelial. Se evaluaron las redes neuronales convolucionales DenseNet-161 y EfficientNet-B7 con un dataset de 1106 imágenes de 588 pacientes, logrando precisiones del 88.5% y 89.5% respectivamente para la clasificación en cuatro clases, y del 91.4% y 92.6% para tres clases de NIC. Estos resultados, comparables con el rendimiento de expertos humanos, demuestran que el deep learning es una herramienta eficaz para el diagnóstico automatizado de lesiones cervicales. Este enfoque es relevante para nuestro proyecto, que aplica técnicas de deep learning para clasificar imágenes obtenidas por la colposcopia y mejorar la detección del riesgo de cáncer de cuello uterino [30].
- Densely Connected Convolutional Networks: Introduce DenseNet, una arquitectura de redes neuronales convolucionales que conecta cada capa con todas las capas anteriores, mejorando significativamente la eficiencia

y precisión del entrenamiento. DenseNet aborda el problema del desvanecimiento de gradientes al mejorar la propagación de información y gradientes, permite la reutilización de características aprendidas por capas anteriores, y reduce la cantidad de parámetros necesarios al evitar la duplicación de mapas de características. Estas ventajas hacen que DenseNet sea especialmente eficaz para tareas complejas de clasificación de imágenes, lo cual es relevante para el proyecto, que utiliza deep learning para clasificar imágenes de colposcopia y mejorar la detección del riesgo de cáncer de cuello uterino [29].

- Accurate multi classification and segmentation of gastric cancer based on a hybrid cascaded deep learning model with a vision transformer from endoscopic images: El proyecto se enfoca en la clasificación y segmentación de imágenes del cáncer gástrico utilizando un modelo híbrido de aprendizaje profundo. Este modelo combina una red neuronal convolucional (CNN) mejorada basada en GoogLeNet y un transformador de visión (ViT). La CNN extrae características espaciales de las imágenes endoscópicas y clasifica en categorías: normales, cáncer gástrico temprano y avanzado. El ViT mejora la precisión mediante atención multicabezal. Además, se usa Faster R-CNN para segmentar y evaluar las lesiones cancerosas. Este proyecto se alinea con nuestros objetivos de mejorar la detección y tratamiento de esta enfermedad mediante técnicas avanzadas de deep learning [31].

4. GESTIÓN DE LA BASE DE DATOS

4.1. Fuentes de datos (NIH, IARC, CITOBOT.)

Se utilizaron imágenes y datos clínicos de tres fuentes principales: el NIC (Instituto Nacional de Salud de los EE.UU.), la IARC (Agencia Internacional para la Investigación sobre Cáncer) y el sistema CITOBOT (proyecto Universidad Javeriana). Estas bases contienen imágenes colposcópicas tomadas en campañas de tamizaje, junto con resultados citológicos y, en algunos casos, presencia confirmada del virus del papiloma humano (HPV).

Después del procesamiento independiente de cada fuente de datos (IARC, NIH y CITOBOT), se construyó un único dataset consolidado con las imágenes colposcópicas y sus respectivas etiquetas diagnósticas. En la tabla 2 se resumen los aportes de cada fuente:

Tabla 2. Caracterización de aportes de cada fuente de datos

Fuente de Datos	Total de imágenes procesadas	Etiquetas disponibles	Etiqueta binaria aplicada
IARC	1116 imágenes	Sin patología / Con patología	0: sin patología, 1: con patología
NIH	1197 imágenes	HPV_STATUS: -1, 0, 1, 2, 3	-1 descartado, 0: sin patología, ≥1: 1
CITOBOT	1614 imágenes	Normal, NIC1, NIC2, Carcinoma	Normal = 0, cualquier otro = 1

Las imágenes con valores no concluyentes (HPV_STATUS = -1) en la base NIH fueron eliminadas del dataset.
Este proceso permitió unificar imágenes provenientes de diferentes orígenes clínicos con criterio de clasificación distintos, unificando las etiquetas diagnósticas bajo un esquema binario común:

- 0 = Sin patología (normal)
- 1 = Con patología (patológica)

Además, se aplicaron filtros adicionales para:

- Asegurar que las imágenes referenciadas realmente existieran.
- Eliminar duplicados o registros sin información diagnóstica válida.
- Normalizar los nombres de las columnas y el esquema de etiquetas, además de agregar una columna que permite identificar la fuente de cada imagen.

Finalmente, las imágenes de todas las fuentes fueron copiadas al directorio case_images/, creando una estructura uniforme para los siguientes pasos del pipeline de preprocesamiento y entrenamiento. En total, se integraron **3927 imágenes**. Los detalles de este proceso se encuentran en el ***Anexo A.1***.

4.1.1. Caracterización inicial del dataset combinado

Las imágenes varían en resolución entre 640x480 px y 4256x2500 px, lo que implica la necesidad de normalización. También, se observa que el etiquetado de la variable objetivo (presencia o no de HPV) no es consistente en cuanto al número de clases, particularmente el caso de CITOBOT se tiene documentado que su clasificación es como resultado del estudio de biopsia, mientras que para las otras fuentes, no se tiene detalle del proceso de etiquetado "Tabla 3". Esta heterogeneidad plantea desafíos en la unificación de criterios para el modelo.

Tabla 3. Distribución de imágenes según su fuente

Fuente	Cantidad de imágenes	Etiquetas disponibles	Resolución promedio
NIH	1208	HPV, citología	1920x1080
IARC	2258	HPV	4256x2500
Citobot	461	HPV automático	800x600

Con respecto al balance de clases se encuentra que la base de datos de IARC presenta un buen balance, mientras que las bases de datos de NIH y CITOBOT si presentan un desbalance de clases, con un menor número de muestras para la clase positiva. En la figura 6, se presenta un gráfico de barras con la distribución de las clases positiva y negativa de acuerdo a la fuente de datos.

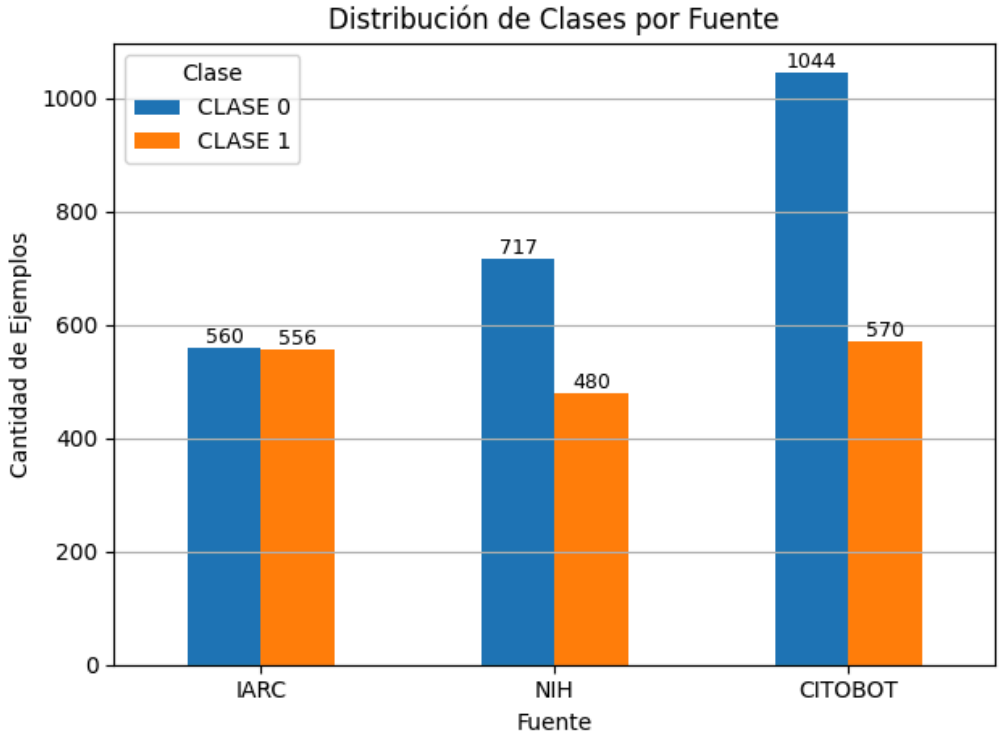


Fig 6. Distribución de clases por fuente de datos

4.1.2 Retos derivados de la integración de fuentes múltiples

Al tratarse de fuentes heterogéneas, se identificaron diferencias significativas en la calidad y estilo de las imágenes: CITOBOT produce imágenes más centradas y de menor resolución, mientras que NIH contiene registros segmentados manualmente, IARC es una base de datos sin segmentación con resolución de imágenes media y sin transformación alguna, es común encontrar imágenes con espéculos, e instrumental médico presentes en las imágenes. Estas diferencias introducen posibles sesgos en el aprendizaje del modelo. Además, se detectó una ligera desproporción en la cantidad de casos positivos y negativos para HPV, así como inconsistencias en los nombres de archivos y etiquetas asociadas, que fueron corregidas mediante procedimientos de limpieza y validación previa al entrenamiento.

En la figura 7 se presentan diferencias entre las imágenes provenientes de las tres fuentes de datos, tanto en términos de calidad como de condiciones de adquisición.

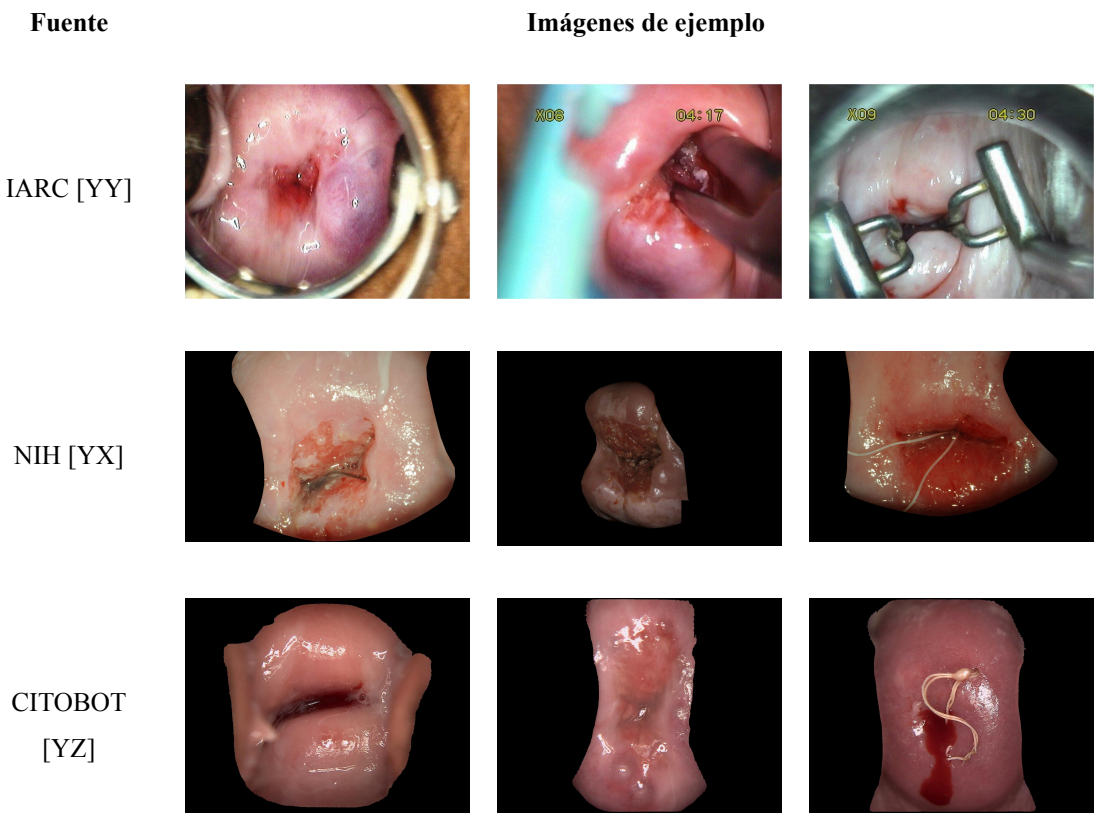


Fig 7. Ejemplos de colposcopia por fuente de datos

Como se puede observar en la figura 7, se pueden presentar artefactos clínicos como espéculos, pinzas u otros instrumentos, así como variaciones significativas en iluminación, enfoque y contraste. Esta heterogeneidad representa uno de los principales desafíos para lograr una base de datos homogénea que permita el entrenamiento efectivo del modelo. Un fenómeno recurrente en la mayoría de las imágenes es el reflejo especular, un artefacto óptico originado por la reflexión directa de la luz sobre superficies húmedas o brillantes, que carece de valor diagnóstico y puede inducir errores en el aprendizaje automático si no es identificado o mitigado adecuadamente.

4.2. Preprocesamiento de imágenes

4.2.1 Exploración de técnicas de preprocesamiento

En un primer experimento se exploraron varias estrategias de preprocesamiento para mejorar la calidad de las imágenes y la capacidad de los modelos para aprender patrones relevantes, ver **Anexo A.2**.

- **CLAHE con clip limit adaptativo**: Ajuste del parámetro clipLimit entre 2 y 8 en función del contraste y entropía. Esta técnica es ampliamente utilizada en imágenes médicas por su capacidad para mejorar detalles sin incrementar excesivamente el ruido [32].
- **Eliminación de reflejos especulares**: Mediante máscaras HSV y corrección con cv2.inpaint() usando el método

TELEA. Esta estrategia está alineada con trabajos previos que buscan eliminar artefactos ópticos que pueden interferir en el diagnóstico [10].

- **Normalización de color en espacio LAB**: Usando percentiles (5 y 95) y una transformación adaptativa de los canales L, A y B para conservar el rango dinámico. Esta estrategia busca evitar que el modelo aprenda patrones espurios derivados del equipo de adquisición [32].
- **Ajustes de iluminación y saturación**: Pequeñas transformaciones para reducir variabilidad entre dispositivos de captura. Esta estrategia sigue los enfoques propuestos en la literatura para reducir ruido visual no diagnóstico [33].

En un segundo experimento se aplicó un método de **segmentación para eliminar imágenes con espéculo visible**. Se utilizó una combinación de, ver **Anexo A.3**.

- Detección de regiones rojizas en HSV.
- Máscara elíptica centrada en el área de interés.
- Exclusión de zonas excesivamente brillantes.
- Conservación permisiva del centro de la imagen mediante un radio adaptativo.

Este filtrado ayuda a reducir el ruido en el dataset eliminando imágenes mal enfocadas o que incluían objetos clínicos no relevantes (como el espéculo) [32].

En un tercer experimento se diseñó un pipeline más estructurado para **detectar automáticamente el tipo de imagen colposcópica** (luz blanca, filtro verde, ácido acético, solución de Lugol o presencia de sangrado visible). A partir de dicha clasificación, se implementaron los siguientes pasos adaptativos, ver **Anexo A.4**.

- **Detección del orificio cervical y zonas anatómicas:** Incluye la clasificación automática del tipo de imagen colposcópica (luz blanca, filtro verde, ácido acético, sangrado), la eliminación de brillos, la normalización de iluminación, y la segmentación del **orificio cervical y la zona de transformación**. Este tipo de segmentación estructurada ha sido propuesto en investigaciones recientes con resultados prometedores [10].
- **Detección de zonas anormales:** Se implementaron filtros basados en gradientes, textura (Sobel, Frangi) y segmentación en espacio HSV para identificar **zonas potencialmente patológicas**, como áreas acetoblancas, sangrado o vasos atípicos. Estos enfoques son consistentes con técnicas avanzadas en análisis colposcópico automatizado [32].

Aunque este pipeline mostró ser el más detallado y clínicamente orientado, **se observó una pérdida significativa de información visual y la introducción de artefactos** durante las transformaciones, lo que afectó negativamente el entrenamiento del modelo. Por esta razón, su uso fue descartado en la versión final del preprocesamiento.

4.2.2. Resultados y decisión final

A pesar de la sofisticación de estas técnicas, los modelos entrenados con imágenes preprocesadas no superaron en rendimiento a los modelos entrenados con imágenes originales con muy poco procesamiento. La sensibilidad disminuyó y se incrementaron los falsos positivos. Se concluyó que las CNN eran sensibles a alteraciones cromáticas o estructurales artificiales.

Se optó finalmente por el siguiente procesamiento:

- **Segmentación:** Se aplicó segmentación manual en 592 imágenes utilizando software de edición. Esta tarea consistió en enmascarar visualmente elementos no anatómicos como el espéculo, pinzas y otros instrumentos médicos, así como eliminar imágenes con mal enfoque o alteradas por la presencia de lugol. El objetivo fue reducir el ruido visual y evitar que estos elementos introduzcan sesgos en el entrenamiento del modelo.
- **Conversión a RGB**: se normalizó el canal de color para garantizar homogeneidad en las entradas.
- **Redimensionamiento a 320x320 píxeles:** se utilizó interpolación **LANCZOS**, seleccionada por su capacidad para preservar detalles finos y reducir el aliasing en los bordes. Se elige este tamaño como un compromiso entre preservación de detalles anatómicos y eficiencia computacional, además se encuentra en un punto medio para entrenar con cualquiera de los cinco modelos utilizados. También, evaluaron dos enfoques de redimensionamiento:
- **Redimensionamiento con padding**: mantiene la relación de aspecto original de la imagen y rellena las áreas faltantes con color negro para completar las dimensiones requeridas. Este método evita distorsiones anatómicas.
- **Redimensionamiento con recorte (crop)**: también conserva la relación de aspecto, pero en lugar de añadir relleno, recorta la imagen para ajustarla al tamaño objetivo. Este enfoque es particularmente efectivo cuando el **orificio cervical se encuentra centrado** en la imagen, ya que permite preservar las regiones anatómicamente más relevantes sin introducir información artificial.
- **Reducción de reflejos especulares:** Se aplicó un filtro basado en la detección de regiones de alta luminosidad y baja saturación, seguido de técnicas morfológicas e inpainting (*TELEA*)[34] para reconstruir la zona afectada, mejorando la visibilidad de detalles anatómicos clave sin introducir artefactos visuales. ver **Anexo A.2.4**.

Esta estrategia resultó ser la más estable y eficaz en términos de desempeño del modelo, además de ser la más eficiente en términos de uso de recursos y tiempo de procesamiento.

El total de imágenes después del procesamiento fue de **3,397.**

La Figura 8 muestra un ejemplo de la diferencia entre las imágenes originales y las imágenes procesadas.

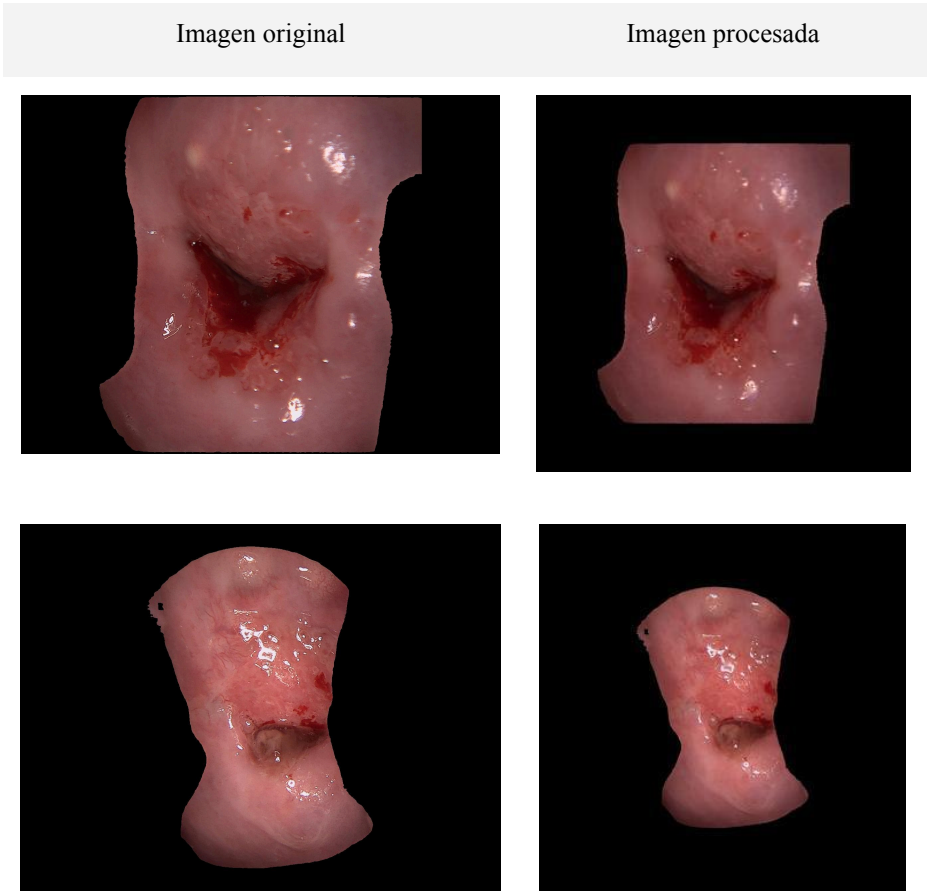


Fig 8. Comparativo imagen original y procesada

Los detalles técnicos de cada experimento de preprocesamiento, así como ejemplos visuales y fragmentos de código, se encuentran disponibles en el **Anexo A.5**.

4.3. Etiquetado, balanceo y partición de los datos

Las imágenes de colposcopia fueron etiquetadas manualmente a partir de metadatos de la variable HPV_STATUS, dividiéndolas en dos clases: **0 (sin patología)** y **1 (con patología asociada al VPH)**. Posteriormente, se validó la distribución de clases y se aplicaron las siguientes estrategias para garantizar un conjunto de datos balanceado y representativo:

a) Validación de la distribución de clases

Se exploró la distribución inicial del conjunto completo, encontrando un desbalance entre clases, como se mostró en la Figura 6. Para evitar que el modelo se sesgara hacia la clase mayoritaria, se optó por aplicar técnicas de balanceo.

b) Balanceo del dataset

Dado el carácter médico del problema y la necesidad de evitar sesgos diagnósticos, se utilizó **undersampling de la clase mayoritaria**, seleccionando aleatoriamente un número igual de muestras por clase. Este enfoque ha sido validado previamente como estrategia efectiva en datasets médicos pequeños para evitar el sobreajuste hacia la clase predominante [35].

c) División en conjuntos de entrenamiento, validación y prueba

Luego del balanceo, se dividió el conjunto de datos en tres subconjuntos:

- Entrenamiento (64%)
- Validación (16%)
- Prueba (20%)

Las divisiones se realizaron utilizando **stratified sampling**, para mantener la proporción de clases en cada subconjunto.

d) Validación cruzada estratificada

Para garantizar consistencia en el comportamiento del modelo durante el entrenamiento, se aplicó una **validación cruzada estratificada con K-Fold (k=3)** sobre el conjunto de entrenamiento. Se analizó la distribución de clases en cada fold y se verificó que se mantuviera equilibrada.

e) Pruebas estadísticas

Si bien con el muestreo estratificado y la validación cruzada con K-Fold se espera tener una separación balanceada de los conjuntos de entrenamiento, se tomó la decisión de realizar una validación empírica mediante **pruebas de homogeneidad de chi-cuadrado**, comparando la proporción de clases entre los subconjuntos. Esta prueba permite evaluar si las diferencias observadas en la distribución de clases entre los splits podrían deberse al azar o a un sesgo sistemático introducido durante la partición. Los resultados del test chi-cuadrado y otras métricas de equilibrio se presentan en el **Anexo A**.

4.4. Descripción final del dataset (tablas, figuras, proporciones)

El dataset se construyó a partir de tres fuentes con diferentes esquemas de clasificación diagnóstica:

- **NIH (Estados Unidos)**: Las etiquetas del virus del papiloma humano (HPV) siguen un esquema ordinal, con valores **-1** (no concluyente), **0** (sin patología), y **1, 2, 3** correspondientes a grados crecientes de lesión intraepitelial cervical (NIC).

- **IARC (Agencia Internacional para la Investigación del Cáncer)**: Las imágenes se encuentran etiquetadas como **con patología** o **sin patología**, sin distinción de grado. Esta fuente ofrece imágenes altamente controladas, a menudo acompañadas de segmentaciones manuales.
- **Citobot (sistema automatizado de diagnóstico portátil)**: Utiliza cuatro clases diagnósticas basadas en resultados de biopsia: **Normal**, **NIC1**, **NIC2**, y **Carcinoma** (que incluye casos equivalentes a NIC3). Se identificaron casos donde la clasificación inicial por sospecha visual fue posteriormente descartada mediante diagnóstico histológico, lo cual revela la importancia de contar con etiquetas confirmadas.

Para efectos del entrenamiento del modelo, todas las etiquetas fueron binarizadas en dos clases: **0** (sin patología) y **1** (con patología). Se descartaron los casos **-1** no concluyentes y se agruparon **1, 2, 3, NIC1, NIC2, Carcinoma** como clase **1**.

4.4.1. Distribución original de clases

La distribución inicial del dataset combinando las tres fuentes muestra un leve desbalance entre las clases positivas y negativas ver Tabla 4:

Tabla 4. Distribución clases conjunto original

Clase	Cantidad	Porcentaje
Clase 0 (sin patología)	2081	52.99%
Clase 1 (con patología)	1846	47.01%

Esta diferencia motivó la aplicación de una técnica de **undersampling** controlado de la clase mayoritaria, para garantizar que el modelo no aprenda sesgos relacionados con la frecuencia de las clases.

4.4.2. Distribución después del balanceo y partición del dataset

Tras aplicar el balanceo, los datos fueron divididos en tres conjuntos: **entrenamiento (64%)**, **validación (16%)** y **pruebas (20%)**, manteniendo una distribución estratificada visto en la Tabla 5. Los porcentajes de clase en cada conjunto fueron:

Tabla 5. Distribución de clases balanceada

Conjunto	Clase 0	Clase 1
Entrenamiento	50.00%	50.00%
Validación	49.92%	50.08%

Pruebas	50.07%	49.93%

Esto garantiza que el modelo sea entrenado, ajustado y evaluado sobre conjuntos **equilibrados**, evitando el sesgo de clase y asegurando mayor validez interna.

4.4.3. Validación cruzada estratificada

Para mejorar la robustez del entrenamiento, se utilizó **validación cruzada estratificada con 3 folds** "Tabla 6". Esto implica que, en cada iteración, se preservó la proporción de clases tanto en el conjunto de entrenamiento como en el de validación:

Tabla 6. Datasets con validación cruzada estratificada

Fold	Entrenamiento (Clase 0 / 1)	Validación (Clase 0 / 1)
1	50.00% / 50.00%	50.00% / 50.00%
2	49.97% / 50.03%	50.06% / 49.94%
3	50.03% / 49.97%	49.94% / 50.06%

Esta técnica permite evaluar la estabilidad del modelo ante particiones distintas, confirmando que la distribución de clases se mantiene prácticamente idéntica entre folds.

Gracias a este proceso de **curación, balanceo y validación estratificada**, se obtuvo un dataset confiable, equilibrado y representativo para entrenar modelos de clasificación binaria en imágenes colposcópicas. Esta base de datos consolidada permite evaluar el desempeño de los modelos sin sesgos introducidos por el desbalance o las etiquetas heterogéneas.

5. ENTRENAMIENTO DE MODELOS DE CLASIFICACIÓN

5.1. Arquitecturas exploradas

Se evaluaron dos arquitecturas para la clasificación de patologías cervicales a partir de imágenes colposcópicas:

- **Red neuronal convolucional (CNN) propia**: modelo construido desde cero, con una arquitectura convencional de varias capas convolucionales, capas de pooling y capas densas finales.
- **Transfer learning con modelos pre entrenados**: se usaron arquitecturas de estado del arte como DenseNet121, ResNet50V2, EfficientNetB3, InceptionV3 y EfficientNetV2S, y clasificador usando SVM, Random Forest, KNN, XGBoost, Fully-connected. Con esta estrategia se realizaron dos experimentos:
- **El primero**, comparar todos los modelos y todos los clasificadores para encontrar las mejores combinaciones extractor-clasificador.
- **El segundo**, usando datos sintéticos con la mejor combinación encontrada en el primer experimento.

Estos experimentos constituyen un estudio exploratorio cuyo propósito principal ha sido identificar arquitecturas, modelos y clasificadores que ofrezcan un desempeño prometedor en la detección de patologías.

Los hallazgos derivados de esta fase permitirán orientar experimentos con configuraciones más especializadas, con el objetivo de mejorar la precisión diagnóstica y la interpretabilidad clínica de los modelos. Se trata entonces de una aproximación iterativa, basada en evidencia empírica y validación para obtener el mejor modelo posible.

5.2. Arquitectura 1 - red convolucional

5.2.1 Características de la red convolucional

Se configuró un modelo secuencial de una red neuronal convolucional con las siguientes capas mostrado en la Tabla 7.

Tabla 7. Descripción de las capas de la estrategia 1 con un red convolucional propia

Capa	Tipo de Capa	Descripción
1	Conv2D	Capa convolucional con 32 filtros de tamaño (3x3). Aplica la función de activación tanh y espera una imagen de entrada de (224x224) píxeles con 3 canales. Utiliza regularización L2.
2	MaxPooling2D	Capa de max-pooling con una ventana de (2x2) para reducir la dimensionalidad de los mapas de características.

3	Conv2D	Capa convolucional con 64 filtros de tamaño (3x3) y función de activación tanh. Utiliza regularización L2.
4	MaxPooling2D	Capa de max-pooling con una ventana de (2x2).
5	Conv2D	Capa convolucional con 128 filtros de tamaño (3x3) y función de activación tanh. Utiliza regularización L2.
6	MaxPooling2D	Capa de max-pooling con una ventana de (2x2).
7	Conv2D	Capa convolucional con 256 filtros de tamaño (3x3) y función de activación tanh. Utiliza regularización L2.
8	MaxPooling2D	Capa de max-pooling con una ventana de (2x2).
9	Flatten	Capa que aplana la salida de la capa anterior en un vector unidimensional para conectar con las capas densas.
10	Dense	Capa densa (totalmente conectada) con 512 neuronas y función de activación tanh. Utiliza regularización L2.
11	Dropout	Capa de dropout con una tasa del 0.3 para prevenir el sobreajuste.
12	Dense	Capa densa con 256 neuronas y función de activación tanh. Utiliza regularización L2.
13	Dropout	Capa de dropout con una tasa del 0.2.
14	Dense	Capa densa con 128 neuronas y función de activación tanh. Utiliza regularización L2.
15	Dropout	Capa de dropout con una tasa del 0.2.
16 (Salida)	Dense	Capa densa de salida con 1 neurona y función de activación sigmoid para clasificación binaria.

5.2.2. Resultado

A continuación en la tabla se presentan los primeros resultados obtenidos

Tabla 8. Primeros resultados del entrenamiento del modelo

Indicador	Entrenamiento	Validación
Loss	~69%	~69%
Accuracy	~51%	~49%

5.3. Arquitectura 2 - Modelo híbrido: Feature Extractor + Clasificador externo

En este primer experimento se aplicó una estrategia de *transferencia de aprendizaje* con separación explícita entre la etapa de extracción de características y la clasificación. Esta aproximación modular permite explorar de forma sistemática diferentes combinaciones de extractor-clasificador y analizar su impacto en el rendimiento del modelo, especialmente en contextos clínicos.

5.3.1 Comparativa de modelos preentrenados para transferencia de aprendizaje

Para la extracción de características en imágenes colposcópicas, se utilizaron arquitecturas pre entrenadas en **ImageNet**, ampliamente reconocidas por su capacidad de generalización y eficiencia computacional. Estas redes fueron seleccionadas por su balance entre profundidad, número de parámetros, precisión en tareas de clasificación general, y compatibilidad con transfer learning [36].

A continuación en la tabla se presentan las características más relevantes de los modelos utilizados:

Tabla 9. Comparativo características modelos ImageNet evaluados

Modelo	**Num caract.**	**capas**	**Parámetros**	**Top1 accuracy en image net**	**Image size reference**	**última capa convolucional**
EfficientNetV2S	1280	481	~21.5 millones	~84.6%	384x384	top_conv
EfficientNetB3	1536	438	~12 millones	~81.6%	300x300	top_conv
ResNet50V2	2048	190	~25.6 millones	~78.4%	224x224	post_relu
DenseNet121	1024	427	~8 millones	~75.0%	224x224	conv5_block16_concat
InceptionV3	2048	311	~23.8 millones	~78.8%	299x299	mixed10

A partir de las características resumidas en la Tabla 9, se pueden identificar ventajas y limitaciones particulares de cada arquitectura. A continuación, se discuten estos aspectos en detalle, considerando implicaciones prácticas como la capacidad de representación, la eficiencia computacional y la compatibilidad con clasificadores tradicionales [37].

5.3.2. Análisis de características clave

- **Número de parámetros:**
  - Modelos como **ResNet50V2** y **InceptionV3** poseen más de 23 millones de parámetros, lo que puede implicar una mayor capacidad de representación, pero también un mayor riesgo de sobreajuste si no se cuenta con un conjunto de datos suficientemente amplio [38].
  - En contraste, **DenseNet121**, con ~8 millones de parámetros, ofrece una solución más ligera y eficiente.
- **Tamaño de imagen de entrada:**
  - **EfficientNetV2S** requiere imágenes de mayor resolución (384x384), lo cual permite preservar más detalles anatómicos, pero también incrementa el tiempo de procesamiento.
  - Modelos como **ResNet50V2 y DenseNet121,** al trabajar con imágenes más pequeñas (224x224), reducen la carga computacional, aunque pueden perder información espacial fina.
- **Top-1 Accuracy en ImageNet:**
  - Este valor sirve como una referencia general sobre la capacidad del modelo en tareas visuales**.**
  - **EfficientNetV2S** lidera con un rendimiento superior al 84%, seguido por **EfficientNetB3**, mientras que **DenseNet121** y **ResNet50V2** tienen valores por debajo del 80%.
- **Salida de la última capa convolucional:**
  - Este punto es importante para transfer learning, ya que define la dimensionalidad de las características que se extraerán.
  - **DenseNet121** entrega una menor cantidad de características (1024) respecto a **ResNet50V2** o **InceptionV3** (2048), lo cual puede afectar el rendimiento de clasificadores tradicionales si no se compensa con técnicas como fine-tuning o regularización adecuada.

La elección del modelo influye directamente en el equilibrio entre rendimiento, costo computacional y complejidad de ajuste.

5.3.2. Pipeline de entrenamiento sin data augmentation

Se diseñó un flujo de trabajo paso a paso que combina el uso de modelos ya entrenados con técnicas de optimización y evaluación. Este enfoque permitió aprovechar modelos existentes, ajustar sus parámetros y validar su rendimiento en distintas configuraciones de modelo - clasificador para el mismo conjunto de datos y así obtener resultados comparables.

1. **Preprocesamiento de imágenes**:
    - Redimensionamiento a 224x224 px.
    - Aplicación de filtro CLAHE adaptativo para mejorar contraste.
2. **Extracción de características**:
    - Utilizando modelos preentrenados, sin incluir la última capa de clasificación.
3. **División del dataset**:
    - Se aplicó partición estratificada con balanceo de clases para evitar sesgos.
    - Se crearon conjuntos de entrenamiento (*train*), validación (*val*) y prueba (*test*).
4. **Normalización/estandarización de características**:
    - Aplicada a *train* y *val* para homogeneizar las entradas del clasificador.
    - Dependiendo del modelo base se aplicó Normalización, Estandarización o ninguna.
        - **Normalización**: EfficientNetV2S, EfficientNetB3, DenseNet121
        - **Estandarización**: ResNet50V2
        - **Ninguna**: InceptionV3
5. **Búsqueda de hiper parámetros**:
    - Grid **Search** para los clasificadores clásicos (KNN, SVM, RandomForest, XGBoost).
    - Keras **tuner** para el clasificador Fully-connected.
6. **Evaluación final**:
    - Métricas visuales y cuantitativas: Curvas ROC, Heatmaps de desempeño, AUC, matriz de confusión e informe de clasificación.

5.3.3. Comparación de modelos y selección final

Se evaluaron cinco modelos de extracción de características (**DenseNet121**, **ResNet50V2**, **EfficientNetB3**, **InceptionV3** y **EfficientNetV2S**), combinadas con cinco clasificadores (**Fully-connected**, **SVM**, **KNN**, **Random Forest** y **XGBoost**). Algunos detalles adicionales de implementación, ajustes específicos y consideraciones técnicas sobre estos experimentos se describen en el **Anexo B1**.

La siguiente tabla resume los resultados obtenidos:

Tabla 10. Resumen valoración modelos y clasificadores

			Precision (weight avg)		Recall (weighted avg)							
Model	**Classifier**	**Accuracy**	**Sin patologia (0)**	**Con patologia (1)**	**Sin patologia (0)**	**Con patologia (1)**	**F1-score (weighted avg)**	**AUC**	**TP**	**FP**	**TN**	**FN**
DenseNet121	SVM	0.81	0.81	0.82	0.82	0.80	0.81	0.88	297	67	303	72
EfficientNetV2S	SVM	0.80	0.81	0.80	0.80	0.81	0.80	0.89	301	73	297	68
ResNet50V2	SVM	0.79	0.80	0.78	0.77	0.81	0.79	0.87	299	70	285	85
DenseNet121	KNN	0.79	0.77	0.80	0.81	0.76	0.79	0.89	282	71	299	87
DenseNet121	Xgboost	0.79	0.80	0.79	0.78	0.80	0.79	0.87	295	80	290	74
EfficientNetB3	Fully-connected	0.78	0.82	0.75	0.72	0.84	0.78	0.85	310	104	266	59
EfficientNetB3	Random Forest	0.78	0.78	0.78	0.78	0.78	0.78	0.87	286	83	287	83
DenseNet121	Fully-connected	0.78	0.77	0.79	0.81	0.75	0.78	0.86	278	72	298	91
DenseNet121	Random Forest	0.78	0.78	0.78	0.78	0.78	0.78	0.87	286	83	287	83
EfficientNetB3	KNN	0.76	0.74	0.79	0.81	0.71	0.76	0.85	263	69	301	106
EfficientNetB3	Xgboost	0.76	0.82	0.73	0.68	0.85	0.76	0.85	312	118	252	57
ResNet50V2	Fully-connected	0.76	0.77	0.79	0.81	0.75	0.76	0.84	295	100	270	74
InceptionV3	SVM	0.76	0.78	0.74	0.73	0.80	0.76	0.85	295	101	269	74
EfficientNetV2S	Fully-connected	0.75	0.77	0.74	0.72	0.78	0.75	0.83	289	102	268	80
EfficientNetV2S	Xgboost	0.75	0.78	0.73	0.71	0.80	0.75	0.82	295	109	261	74
EfficientNetV2S	KNN	0.74	0.73	0.75	0.76	0.72	0.74	0.83	265	87	283	104
EfficientNetV2S	Random Forest	0.73	0.72	0.75	0.74	0.72	0.73	0.82	259	87	283	110
InceptionV3	KNN	0.73	0.71	0.75	0.78	0.68	0.73	0.82	251	83	287	118
ResNet50V2	Xgboost	0.72	0.79	0.68	0.60	0.84	0.72	0.83	310	147	223	59
ResNet50V2	KNN	0.70	0.70	0.70	0.70	0.70	0.70	0.82	258	102	261	111
ResNet50V2	Random Forest	0.70	0.78	0.66	0.56	0.84	0.69	0.81	311	163	207	58
InceptionV3	Random Forest	0.68	0.82	0.62	0.45	0.90	0.68	0.78	333	204	166	36

El modelo que presentó mejor desempeño fue **DenseNet121,** se puede apreciar en la tabla que en comparación con los otros modelos es el que mejor desempeño presenta al combinarlo con los clasificadores.

El mejor rendimiento global se obtuvo con **DenseNet121 + SVM**, logrando un **AUC de 0.88**, **F1-score de 0.81** y un excelente equilibrio entre sensibilidad y especificidad. Se observaron mejoras consistentes en la detección de casos positivos.

En este punto se consideró que el paso a seguir era realizar el experimento con aumento de datos, para ello se exploró una técnica de generación de datos sintéticos.

5.4. Arquitectura 2 Experimento 2 - Data augmentation sintética - Extractor - Clasificador

5.4.1. Estrategia data augmentation DCGAN

Se utilizaron las imágenes del dataset para entrenar dos modelos que permitieron generar 8000 imágenes con un modelo DCGAN (Red Generativa Antagónica Convolucional Profunda) que fueron usadas en el entrenamiento del modelo de clasificación.

5.4.2. Pipeline de entrenamiento usando data augmentation sintética

Para la generación de imágenes se realizaron los siguientes pasos

- Se hizo un resize del dataset original a 224x224 pixels
- Se filtraron las imágenes en dos grupos con patología y sin patología
- Se generaron las imágenes para cada grupo
- Se mezclan imágenes reales e imágenes sintéticas en el dataset de pruebas
- Se separan 739 imágenes reales solo para la validación del modelo
- Se usó **DenseNet121** para la extracción de categorías
- Se usó una estrategia de random search con un clasificador binario **SVM** con **K-folds**
- Se validó el modelo

5.4.3. Data Augmentation con imágenes sintéticas

Aca podemos ver el resultado de las imágenes generadas

Sin patología	Con patología

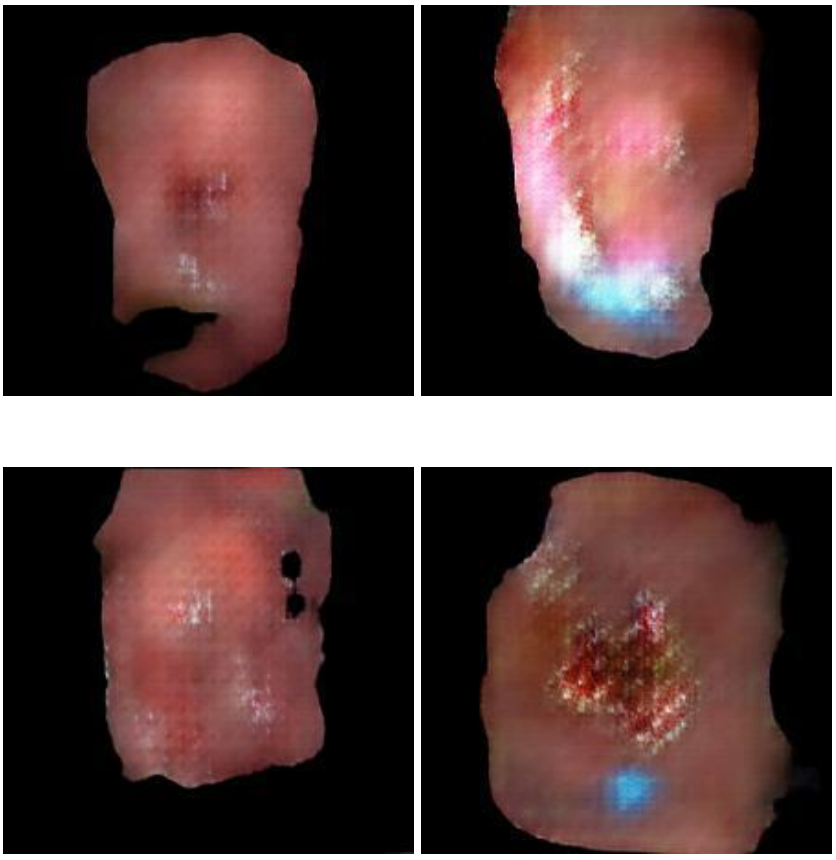


Fig 9. Imágenes sintéticas generadas por modelo DCGAN

5.4.4. Resultados experimento 1 con data augmentation sintética

Considerando la combinación **DenseNet121 + SVM** se presentarán los resultados en la Tabla 11, al aplicar fine-tuning y data augmentation con el fin de evaluar si estas estrategias mejoran el desempeño de esta arquitectura.

Tabla 11. Resultados modelo entrenado con imágenes sintéticas

Clase	Precisión	Recall	F1-score	Soporte
Sin Patología	~84%	~83%	~83%	370
Con Patología	~83%	~85%	~83%	369
Accuracy			~84%	739
Macro Avg	~84%	~84%	~84%	739
Weighted Avg	~84%	~84%	~84%	739

5.5. Evaluación modelos seleccionado (DenseNet121 + SVM)

Después de analizar los resultados de los experimentos realizados a saber:

- **Experimento 1**: Entrenamiento de una red neuronal convolucional propia.
- **Experimento 2**: Evaluación de cinco modelos preentrenados para extracción de características, combinados con cinco clasificadores tradicionales.
- **Experimento 3**: Uso de datos sintéticos con la mejor combinación modelo-clasificador identificada previamente.

Se obtuvo una comprensión más profunda del problema, lo que permitió ajustar el pipeline general y definir una estrategia más sólida para el desarrollo del modelo final. A continuación, se destacan algunos hallazgos clave derivados de estos experimentos:

- **Limitación de datos reales:** Se considera que la cantidad de imágenes (3927) disponibles en los experimentos 1 y 2 no fue suficiente para lograr una adecuada generalización. Esto se evidenció al observar que los mejores resultados se alcanzaron cuando se incorporaron datos sintéticos en el experimento 3, por este motivo se incluirá en el pipeline un paso de data augmentation.
- **Importancia del preprocesamiento:** Se identificó la necesidad de mejorar el procesamiento de imágenes, especialmente en lo relacionado con el enfoque en las zonas anatómicas relevantes. El objetivo es mantener la mayor fidelidad visual posible, optimizando al mismo tiempo las áreas de atención que influencian las decisiones del modelo, adicionalmente, eliminar o segmentar aquellas imágenes que introducen confusión al modelo al contener artefactos o instrumental médico.
- **Adaptación del modelo preentrenado al dominio clínico:** Se manejó la hipótesis que los modelos preentrenados en ImageNet no estaban optimizados para comprender las particularidades visuales de las imágenes colposcópicas. Ante esta limitación, se propuso aplicar fine-tuning descongelando las últimas 50 capas

del modelo DenseNet121. Esta estrategia busca que el modelo aprenda representaciones más especializadas a partir de los datos médicos disponibles, permitiendo mejorar la calidad de las características extraídas para las tareas de clasificación.

- **Uso de técnicas de interpretabilidad:** Se implementó Grad-CAM [39] para visualizar las regiones más relevantes que activan la red neuronal durante la predicción. Esto aporta transparencia y confianza en la toma de decisiones del modelo.
- **Optimización de la búsqueda de hiperparámetros:** Dado que GridSearch puede ser muy lento, se considera beneficioso incluir un paso previo de exploración mediante RandomSearch. Esta estrategia permite reducir el espacio de búsqueda y enfocar el GridSearch en los rangos más prometedores.
- **Ajuste del umbral de decisión:** Se exploró la optimización del umbral de clasificación utilizando el F1-score como métrica objetivo, con el fin de equilibrar la sensibilidad y la precisión del sistema.

5.5.1 Pipeline modelo final

A continuación se presenta el pipeline modificado con los hallazgos de los experimentos previos que se aplicó a la combinación ***DenseNet121 + SVM***.

5.5.1.1. Procesamiento de Imágenes

Se realizó el procesamiento de imágenes mediante el redimensionamiento a 320x320px, aplicación de filtro Clahe adaptativo para mejorar el contraste y reducción de brillo especular.

5.5.1.2. Extracción de Características – DenseNet121 con Fine-Tuning

Para mejorar la calidad de las representaciones extraídas de las imágenes colposcópicas, se implementó una estrategia avanzada de fine-tuning con atención canal basada en arquitecturas preentrenadas en ImageNet. El procedimiento se desarrolló en varias fases:

- **Carga del modelo base sin capa superior**: Se cargó el modelo **DenseNet121** sin la capa de clasificación final y con los pesos preentrenados.
- **Integración de un bloque de atención tipo SE (Squeeze-and-Excitation)**: Se añadió una capa de atención por canal posterior a las salidas convolucionales del modelo base, con el objetivo de enfatizar los canales más relevantes antes del paso de agregación espacial.
- **Congelamiento inicial y entrenamiento leve**: Inicialmente, el modelo base fue congelado y se entrenó solo la capa densa final durante una época, para estabilizar el entrenamiento antes del ajuste fino.
- **Uso de callbacks para control del entrenamiento**:
  Se implementaron dos mecanismos automáticos para mejorar el proceso de entrenamiento:
- **EarlyStopping**, para detener el entrenamiento si no se observa mejora en la pérdida.

- **ReduceLROnPlateau**, para reducir dinámicamente la tasa de aprendizaje en caso de estancamiento.
- **Fine-tuning parcial**: Se descongelaron las últimas *50* capas del modelo base y se entrenaron junto con la capa final, permitiendo que las capas más profundas se ajustaran a los patrones específicos de las imágenes médicas.
- **Uso de Focal Loss**: Para abordar el desbalance entre clases y mejorar la sensibilidad ante casos positivos, se utilizó una función de pérdida Focal Loss con parámetros ajustados (alpha=0.25, gamma=2.0).
- **Agregación mediante Global Average Pooling**: Las activaciones de la red fueron embebidas en vectores representativos utilizando GlobalAveragePooling2D, lo que permite capturar la esencia espacial de la imagen reducida a un vector de características.
- **Construcción de un extractor de características**: Una vez finalizado el entrenamiento, se construyó un modelo exclusivo para la extracción de características, eliminando la capa de salida binaria y manteniendo la estructura convolucional, atención y pooling global.
- **Extracción y almacenamiento**: Las características fueron extraídas en lotes desde todas las imágenes disponibles, generando representaciones numéricas que posteriormente fueron usadas por clasificadores externos como SVM.
- **Visualización con Grad-CAM**: Posteriormente, se utilizó Grad-CAM para identificar las regiones de atención del modelo, proporcionando interpretabilidad sobre qué zonas de la imagen influyeron en la predicción.

En las Figuras 10 y 11 se presentan dos ejemplos, uno correspondiente a una imagen con diagnóstico de patología (Clase 1) y otro a una imagen sin patología (Clase 0).

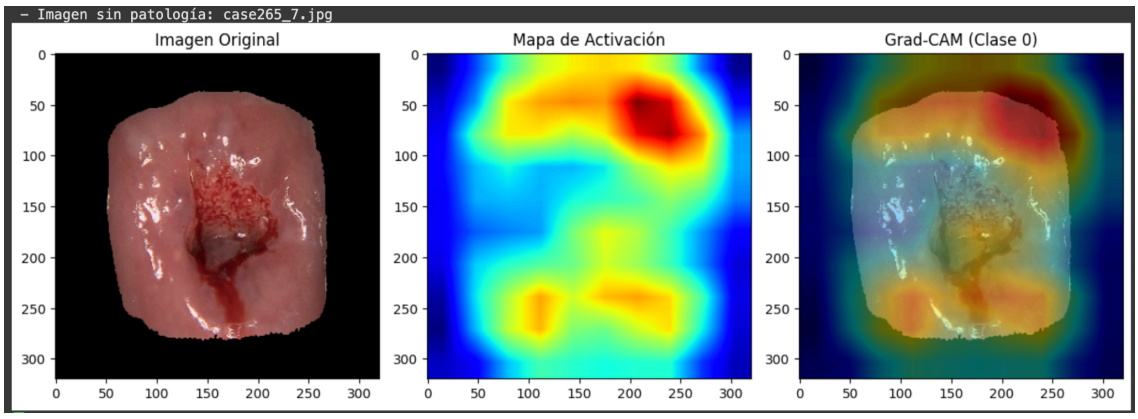


Fig 10. Grad-CAM - imagen con patología

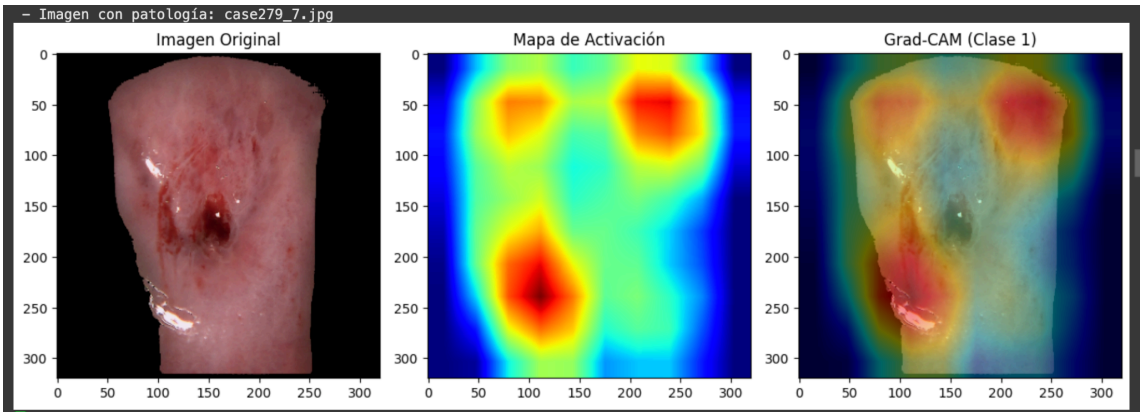


Fig 11. Grad-CAM - imagen sin patología

- En la imagen con patología (*fig 10*), se observa que la atención del modelo está correctamente centrada en la zona del orificio cervical, especialmente en áreas donde hay presencia de lesiones visibles (rojo oscuro). Esta concentración sugiere que el modelo aprendió a enfocar sus predicciones en regiones anatómicamente relevantes.
- Por otro lado, en la imagen sin patología (*fig 11*), el mapa de activación también se concentra en el centro de la imagen, pero con menor intensidad. El Grad-CAM indica que la atención se distribuye de forma más difusa, posiblemente reflejando la ausencia de lesiones destacadas. No obstante, en ambos casos, se mantiene una atención coherente hacia la zona central del cuello uterino, lo que respalda que el modelo no está tomando decisiones arbitrarias basadas en artefactos externos o bordes de la imagen.

5.5.1.3. División de conjuntos del dataset

El conjunto completo de características (X_features) y etiquetas (y_labels), previamente extraídas mediante un modelo con ajuste fino (fine-tuning), fue dividido en tres subconjuntos: entrenamiento, validación y prueba. Para garantizar una evaluación justa del modelo y evitar filtración de datos, se siguió el siguiente procedimiento:

- Balanceo **previo por *undersampling***: Dado que el dataset presentaba un desbalance entre clases, se aplicó un muestreo aleatorio estratificado de la clase mayoritaria, igualando el número de muestras entre clases antes de realizar cualquier partición. Esto es especialmente relevante en el contexto médico, donde el desbalance puede sesgar el modelo.
- División estratificada:
  - Se reservó un 20% de las muestras balanceadas para conformar el conjunto de prueba (*test set*), que no se utilizó en ninguna etapa del entrenamiento.
  - El 80% restante se subdividió nuevamente en entrenamiento y validación, destinando 20% para validación (*validation set*) y el 60% final para entrenamiento (*training set*).
  - En cada partición, se utilizó estratificación por clase para asegurar que la proporción de etiquetas se mantuviera constante en todos los subconjuntos.
  - Resultados **de la partición**:

La partición final arrojó las siguientes estadísticas:

- **Conjunto de Entrenamiento**:
  - Total de muestras: 1999
  - Clase 0: 49.97%
  - Clase 1: 50.03%
- **Conjunto de Validación**:
  - Total de muestras: 500
  - Clase 0: 50.00%
  - Clase 1: 50.00%
- **Conjunto de Prueba**:
  - Total de muestras: 625
  - Clase 0: 50.08%
  - Clase 1: 49.92%

- **Persistencia y reutilización:** Los conjuntos generados se almacenan en disco en formato .npy para garantizar reproducibilidad y eficiencia en ejecuciones posteriores.
- **Validación de consistencia:** Se aplicó validación cruzada estratificada (*Stratified K-Fold*) sobre el conjunto de entrenamiento para confirmar la distribución homogénea de clases entre las particiones internas.

5.5.1.4. Data Augmentation Realista

Para incrementar la diversidad del conjunto de entrenamiento y mejorar la capacidad de generalización del modelo, se aplicó una estrategia de *data augmentation* basada en transformaciones realistas, sin alteración de la orientación anatómica (es decir, sin flip horizontal ni vertical).

Se definió un conjunto de transformaciones usando la librería Albumentations, incluyendo:

- Ajustes de brillo y contraste (RandomBrightnessContrast),
- Rotaciones leves (Rotate, ShiftScaleRotate),
- Desenfoque por movimiento (MotionBlur),
- Distorsión óptica (OpticalDistortion),
- Cambios en canales de color (RGBShift).

Cada imagen del conjunto de entrenamiento fue aumentada generando **4 nuevas versiones** (*on the fly*) con diferentes combinaciones aleatorias de estas transformaciones, manteniendo la resolución final en 320x320 píxeles.

Posteriormente, se utilizaron los modelos previamente entrenados (con fine-tuning y atención por canal) para extraer características de estas imágenes aumentadas. Las representaciones generadas se combinaron con las originales para conformar el nuevo conjunto completo de entrenamiento.

- **Total de imágenes originales**: 1999

- **Total de imágenes aumentadas**: 7996
- **Total de muestras de entrenamiento final**: 9995
- **Dimensión de las características**: 1024 por muestra

Esta técnica permitió aumentar el volumen del dataset sin necesidad de adquirir nuevas imágenes, reforzando la robustez del modelo ante pequeñas variaciones en el enfoque, iluminación o distorsión, manteniendo al mismo tiempo la fidelidad anatómica de las imágenes colposcópicas.

5.5.1.5. Procesamiento de Características

- Para los modelos cuyo espacio opera naturalmente en el rango [0, 1], (como EfficientNet y DenseNet121), se aplicó una estrategia de normalización mediante MinMaxScaler. Esta técnica se aplicó únicamente si las características extraídas no estaban ya normalizadas.
- El escalador fue ajustado exclusivamente sobre el conjunto de entrenamiento y posteriormente aplicado a los conjuntos de validación y prueba.
- Para más detalles sobre la implementación de esta estrategia, ver **Anexo B.1.3.**

5.5.1.6. Entrenamiento del Clasificador SVM

- **Búsqueda de Hiper parámetros**
- **Random Search**: exploración inicial de combinaciones amplias de parámetros como C, kernel, gamma.
- **Grid Search**: ajuste fino en el espacio de hiper parámetros más prometedores.
- **Se** utilizaron validaciones cruzadas y se almacenaron métricas clave:
  - Accuracy, F1-score, curva ROC-AUC.
  - Matriz de confusión y heatmaps de resultados.

5.5.1.7 Selección de Threshold Óptimo

- **Se** ajustó el **umbral de decisión** para maximizar el **F1-score**, en lugar de usar el valor estándar de 0.5.
- **Se** utilizó la **curva precision-recall** para determinar el punto de corte más adecuado, mejorando el balance entre sensibilidad y precisión.

5.5.1.8. Evaluación del modelo en el conjunto de Test.

- **Se** evalúa el modelo y se obtiene informe de clasificación, matriz de confusión y curva ROC_AUC.
- **Se** realiza análisis con Grad-CAM

5.5.2. Resultados modelo final

En esta sección se presentan los resultados de los pasos 6 y 7 del pipeline correspondientes a la búsqueda de los hiperparámetros y el entrenamiento del modelo.

5.5.2.1. Random Search

Para reducir el tiempo computacional de una búsqueda exhaustiva con GridSearch, se implementó una estrategia de búsqueda aleatoria de hiperparámetros (RandomizedSearchCV) como paso preliminar. Esta técnica permite explorar

eficientemente un espacio amplio de combinaciones posibles con menor costo computacional, ayudando a identificar regiones prometedoras del espacio de búsqueda para su posterior refinamiento.

5.5.2.1.1. Parámetros Evaluados

Se exploraron dos tipos de kernel para la SVM: **rbf** (base radial) y **poly** (polinómico). Los rangos seleccionados para cada hiperparámetro se definieron en función del conocimiento previo del dominio:

- **C:** [1, 1000] en escala logarítmica (np.logspace(0, 3, 15)). Controla el grado de penalización a errores en el margen. Un rango amplio permite explorar desde regularización fuerte (valores pequeños) hasta márgenes muy ajustados (valores grandes).
- **Gamma:** [1e-4, 3.16] en escala logarítmica (np.logspace(-4, 0.5, 15)) Define la influencia de una sola muestra. Valores bajos generan decisiones más suaves; valores altos tienden a sobre ajustar el modelo.
- **Kernel**: ["rbf"] o ["poly"] Se limita la búsqueda a un tipo de kernel por iteración para enfocar los recursos computacionales.
- **Degree (solo para poly):** [2, 3, 4]. Se evaluaron polinomios de bajo grado, considerando su mejor adecuación a conjuntos de datos con estructura no lineal pero sin ruido excesivo.
- **Coef0 (solo para poly):** [0, 0.5, 1.0]. Afecta el comportamiento del kernel polinómico, particularmente su suavidad.
- **Class Weight:** Se probaron pesos personalizados para mitigar el impacto del desbalance de clases, incluyendo:
  - "balanced"
  - {0: 1.0, 1: 1.5}, {0: 1.0, 1: 2.0}, {0: 1.0, 1: 3.0}
  - None

5.5.2.1.2. Resultados

Se ejecutaron entre 100 y 300 iteraciones aleatorias, cada una evaluada mediante validación cruzada estratificada con 3 particiones. La métrica objetivo fue el AUC (Área Bajo la Curva ROC), lo que permite capturar el desempeño global del modelo frente a ambos tipos de error (falsos positivos y falsos negativos).

Posteriormente, se seleccionó el mejor modelo según AUC y se evaluó su rendimiento en el conjunto de validación externa. Además, se generaron visualizaciones como:

- Dispersión de AUC en función de log10(C) y log10(Gamma)
- Heatmap de AUC para combinaciones C vs Gamma (solo kernel RBF)
- Boxplot comparativo entre kernels (en caso de búsquedas mixtas)

Esta exploración permitió reducir el espacio de búsqueda para la siguiente fase de *GridSearch*, enfocando los esfuerzos computacionales en los rangos más prometedores detectados por *RandomSearch*.

El resultado de *RandomSearch* se muestra a continuación:

Desempeño del mejor modelo:

Accuracy: 0.8200 | AUC: 0.9023 | Recall: 0.7800 | Specificity: 0.8600

Este resultado nos indica que el modelo discrimina bien entre las clases AUC > 0.9, lo que es un buen indicador, la exactitud global es buena 82%, hay un compromiso entre Recall (78%) y Especificidad (86%), lo que sugiere que el modelo está algo más inclinado a evitar los falsos positivos que a capturar todos los positivos.

A continuación se presenta la tabla con las 10 mejores combinaciones de hiperparámetros:

Tabla 12. Top mejores 10 combinaciones de hiperparámetros RandomSearch

Iteración	C	Gamma		Kernel	AUC
1	1000	0.037276		rbf	0.899871
11	138.949549	0.037276		rbf	0.899871
8	1000	0.037276		rbf	0.899871
40	138.949549	0.037276		rbf	0.899871
50	84.83429	0.037276		rbf	0.899871
41	19.306977	0.037276		rbf	0.899626
31	227.584593	0.078137		rbf	0.898622
0	1000	0.078137		rbf	0.898622
13	19.306977	0.078137		rbf	0.898622
34	84.83429	0.078137		rbf	0.898622

En esta tabla se observa que hay estabilidad en el AUC, ya que para valores distintos de C el AUC es casi constante, esto sugiere que el modelo es robusto en ese rango y que hay una meseta de buen desempeño cuando Gamma = 0.037276. Cuando se baja el valor de Gamma, ocurre lo mismo con AUC.

Vemos que la penalización (C) no está influyendo mucho en el intervalo ya que valores entre 84 y 1000 producen el mismo AUC.

Con este resultado se puede definir un rango para la búsqueda con *GridSearch* que sería así:

- C entre 85 y 1000
- Gamma alrededor de 0.037
- kernel rbf

Continuando con el análisis de resultados para *RandomSearch* se presenta la gráfica de AUC vs log10(C) coloreado por log10(gamma)

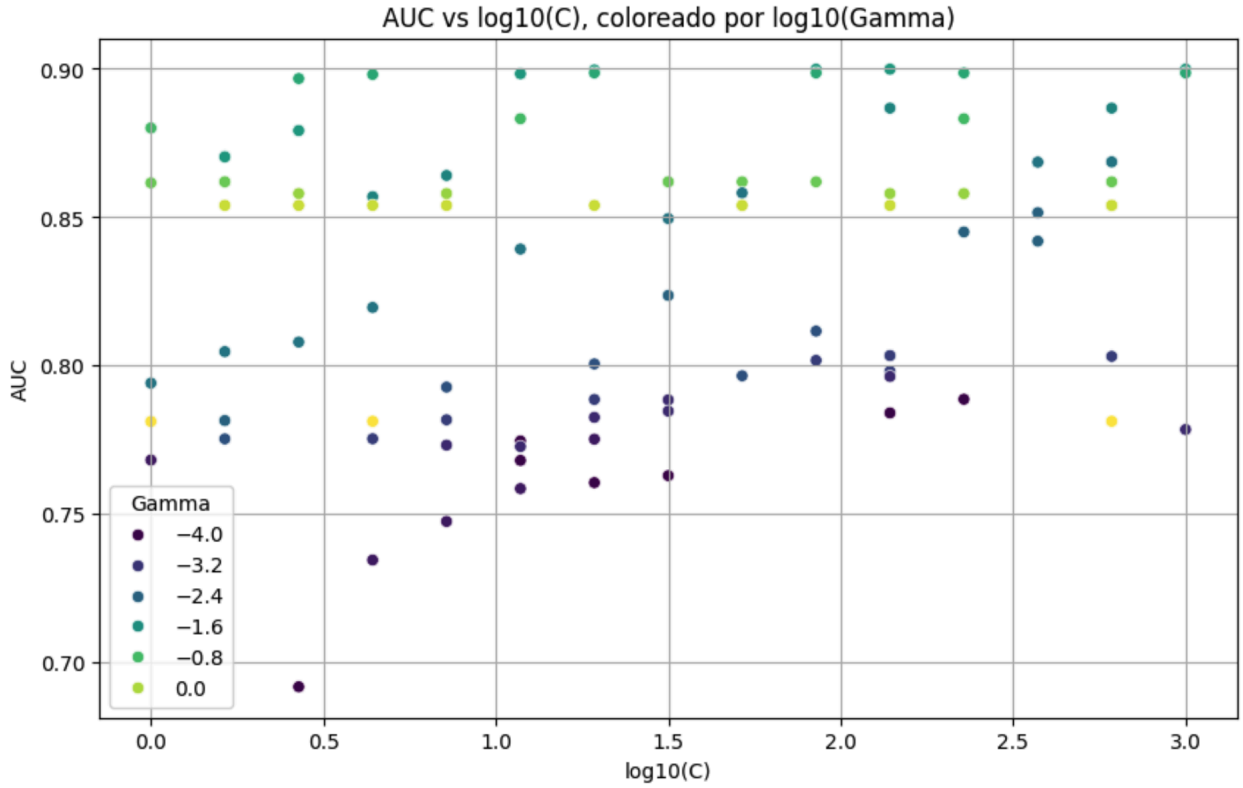


Fig 12. AUC vs log10(C) coloreado por log10(gamma)

Que representa esta figura:

- En el eje X se tiene log10(C) que es la regularización (0 a 3, es decir, C de 1 a 1000)
- En el eje Y AUC (de 0.68 a 0.9)
- Color log10(gamma) desde valores pequeños 1e-4 en morado oscuro, hasta 1 en amarillo (gamma grande)

Análisis de resultados:

- Los puntos más altos en el eje Y (AUC ≈ 0.90) están asociados a log10(gamma) cercanos a -1.6 o 0, que en escala lineal corresponde a gamma ≈ 0.025 a 0.1, esto coincide con el gamma = 0.037 de la tabla.
- Se confirma el buen rendimiento de AUC para valores de regularización de 30 a 1000
- Gamma muy pequeño o muy alto perjudica, el rango intermedio óptimo es aproximadamente entre 0.037 y 0.08.

En la gráfica de heatmap que se muestra a continuación se puede confirmar el desempeño del AUC para diferentes valores de gamma y C.

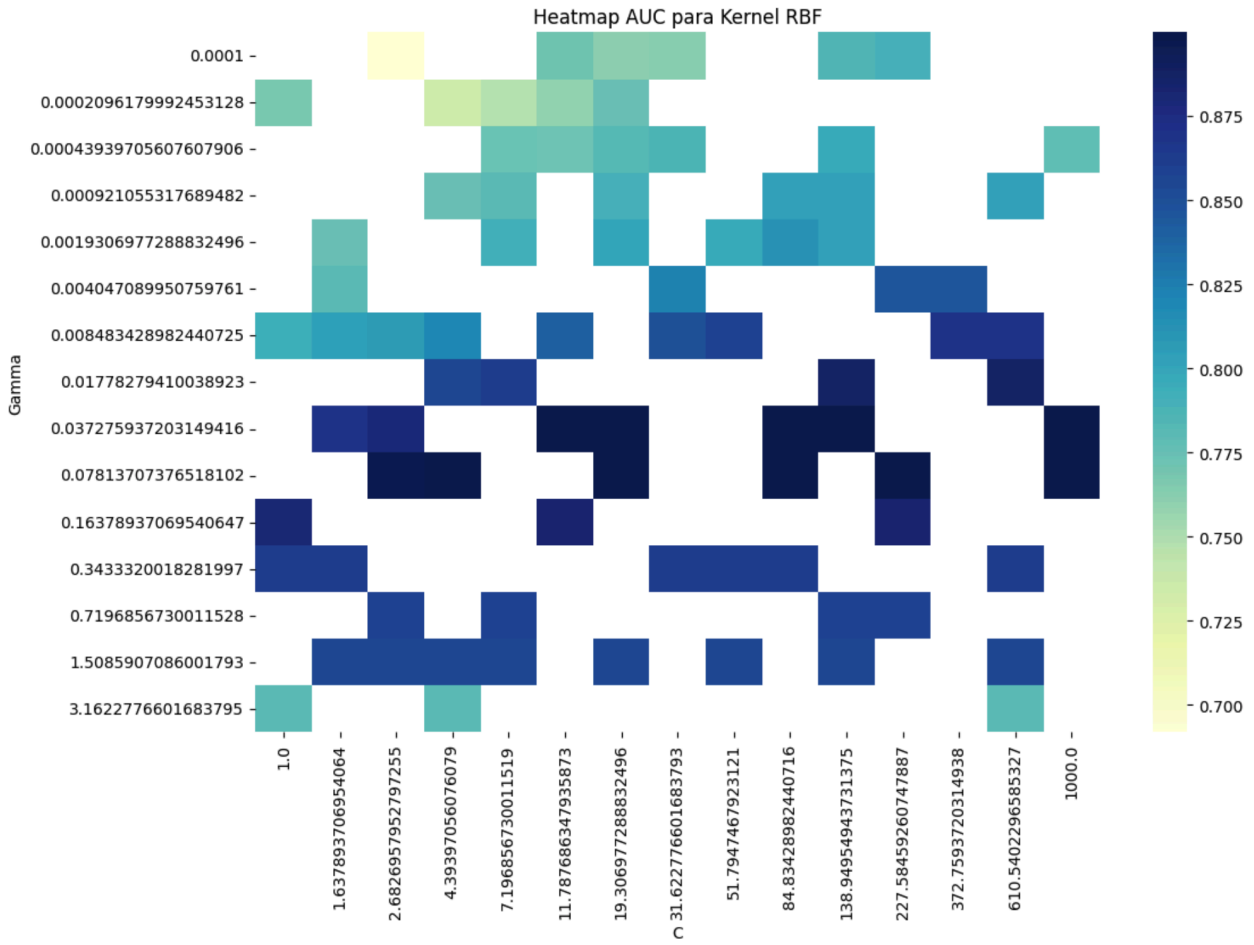


Fig 13. Heatmap AUC para el kernel RBF

Esta gráfica refuerza el resultado obtenido hasta ahora con respecto a los rangos más prometedores para realizar una búsqueda fina con *GridSearch*, al identificar la zona más oscura se puede observar que no solo confirma el resultado sino que indica que hay robustez en esa zona, es decir, hay más valores en ese rango que producen AUC alto, lo que se traduce en un buen margen de estabilidad.

Un detalle interesante de esta gráfica es que a partir de gamma > 0.163 se produce una disminución grande del AUC sin importar el valor de C, esto indica sobreajuste o pérdida de capacidad de generalización, es decir, el modelo se enfoca mucho en puntos individuales.

Mejores rangos según esta gráfica:

- C entre 138 y 372
- gamma entre 0.037 y 0.078.

5.5.2.2. Grid Search

Con los rangos obtenidos con RandomSearch procedimos a realizar una búsqueda de hiperparámetros con

GridSearch, se consideró enfocar el análisis en la parte baja de C, ya que se identificó estabilidad en este parámetro, pero se deseaba una mayor regularización para generalizar mejor y evitar el sobreajuste, la figura siguiente muestra el espacio de búsqueda:

```
svm_params_grid = {
        "C": [
            19.306977, 84.834290, 1000.0, 138.949549, 227.584593
        ],
        "gamma": [
            0.030, 0.035, 0.037, 0.040, 0.045, 0.050, 0.060, 0.078, 0.085
        ],
        "kernel": ["rbf"],
        "class_weight": [
            "balanced",
            {0: 1.0, 1: 1.5},
            {0: 1.0, 1: 2.0}
        ]
    }
```

Fig 14. Espacio de búsqueda de hiper parámetros para GridSearch

5.5.2.2.1. Descripción del proceso:

- Dado el elevado número de combinaciones posibles, se dividió la evaluación en bloques de 10 configuraciones, procesados en paralelo con joblib.Parallel, usando múltiples núcleos (n_jobs), para acelerar la búsqueda.
- Luego de cada bloque, los resultados fueron almacenados en disco (checkpoint), permitiendo reanudar el proceso en cualquier momento sin pérdida de progreso.
- Se implementó un mecanismo para eliminar combinaciones duplicadas, garantizando una evaluación única por conjunto de hiperparámetros. Se implementó una función que convierte los parámetros en estructuras hashables y elimina repeticiones.

5.5.2.2.2 Validación Cruzada

- Cada configuración fue evaluada usando validación cruzada estratificada (*StratifiedKFold*) con 5 particiones (CV=5), preservando la proporción de clases en los subconjuntos de entrenamiento y validación. Las métricas fueron promediadas entre los folds.
- Las métricas promediadas para cada configuración fueron:
- **Accuracy** = (TP + TN) / (TP + TN + FP + FN)
- **Recall (Sensibilidad)** = TP / (TP + FN)
- **Specificity** = TN / (TN + FP)
- **AUC** = Área bajo la curva ROC, refleja la capacidad discriminativa del modelo

5.5.2.2.3. Selección del Mejor Modelo

- Se seleccionó la configuración con el mayor AUC promedio como modelo final.
- Posteriormente, se entrenó nuevamente sobre todo el conjunto de entrenamiento con dicha configuración óptima.
- El desempeño fue reportado sobre el conjunto de validación.

5.5.2.2.4. Visualizaciones

- Para facilitar la interpretación del proceso se generaron:
- Gráficas de desempeño promedio por valor de C.
- Mapas de calor (heatmaps) de AUC en función de C y gamma.

5.5.2.2.5. Resultados

La búsqueda de hiperparámetros identificó la siguiente configuración óptima:

- **C:** 19.306977
- **class_weight:** balanced
- **gamma:** 0.078
- **kernel:** rbf

Esta configuración alcanzó las siguientes métricas de rendimiento:

- **AUC:** 0.9912
- **Recall:** 0.9648
- **Specificity:** 0.9646
- **Accuracy:** 0.9647

Como se observa en la Figura 15, las diferentes métricas de rendimiento muestran un comportamiento estable en relación con el valor de C. El AUC se mantiene consistentemente alto (alrededor de 0.991) a través de todo el rango de valores de C evaluados, lo que indica un rendimiento robusto del modelo independientemente del nivel de regularización aplicado, con un buen desempeño en la tarea de clasificación binaria, además mantiene equilibrio entre sensibilidad (recall) y especificidad.

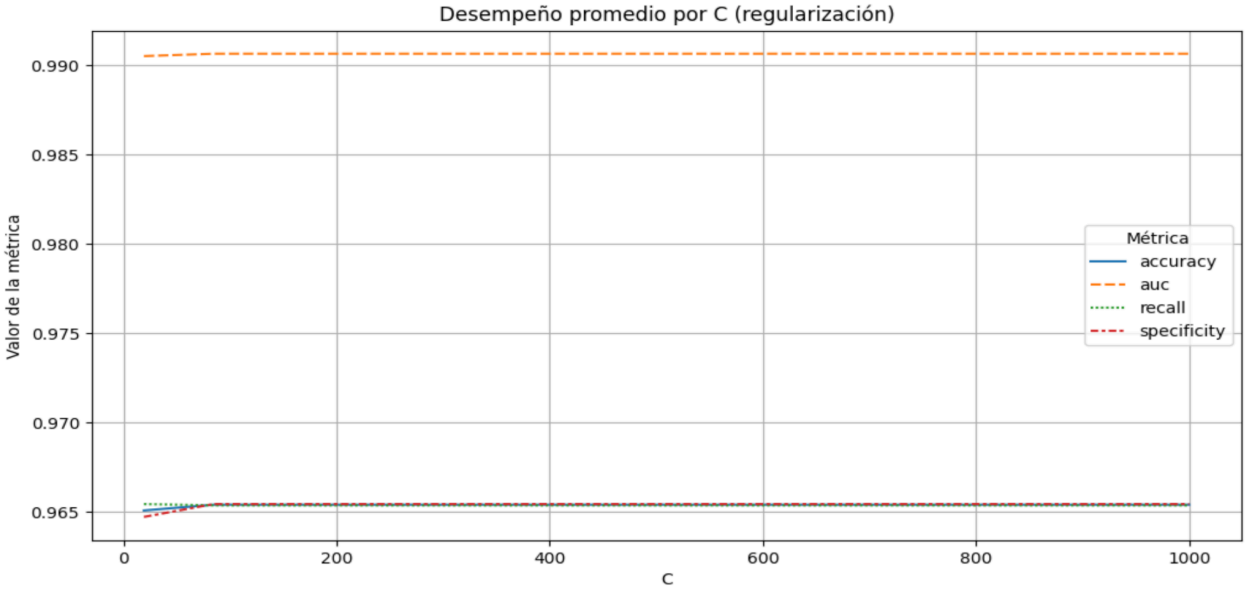


Fig 15. Desempeño promedio por regularización (C)

La métrica **AUC** se mantiene prácticamente constante para valores de C mayores a 20, indicando que el modelo es poco sensible a grandes cambios en la penalización de margen.

Tanto **Recall** como **Specificity** también se estabilizan, lo que sugiere un punto de saturación del modelo.

- En cuanto a la relación entre kernel y el parámetro C, en la siguiente figura se puede ver un mapa de calor que muestra el AUC promedio para diferentes valores de kernel (en este caso rbf) y C

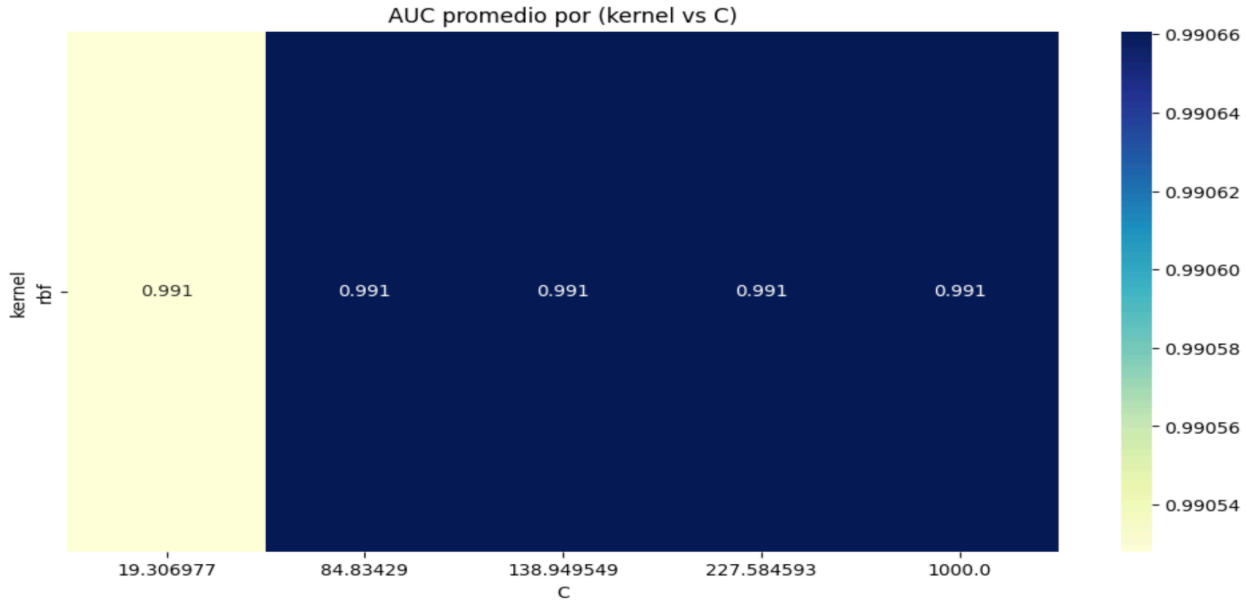


Fig 16. AUC promedio por kernel vs C

Se observa que todas las combinaciones producen un AUC muy similar, cercano a 0.991, esto confirma que el modelo ha alcanzado una meseta de estabilidad.

Se puede decir que el conjunto de datos es relativamente bien separable en el espacio de características transformadas por el kernel rbf, y que no se requiere un ajuste extremadamente preciso para lograr un buen rendimiento.

La saturación de las métricas, cuando Accuracy, Recall, Specificity y AUC se mantienen constantes para un amplio rango de valores de C, puede tener distintos significados:

- **Caso positivo:** que el modelo ha aprendido patrones robustos, y pequeñas variaciones en los hiperparámetros no afectan mucho el rendimiento.
- **Caso negativo** (se comprueba en la evaluación): que el modelo ha ajustado en exceso a los datos de validación cruzada, aprendiendo patrones que no se generalizan bien. Esto es compatible con una caída de rendimiento en el conjunto de prueba.

5.5.3 Retos encontrados (overfitting, desbalance de clases)

Durante el entrenamiento se enfrentaron varios retos clave:

- **Overfitting**: las arquitecturas Fully-connected tendieron a sobre ajustar los datos sin regularización ni augmentación.
- **Reflejos especulares**: a pesar de aplicar técnicas avanzadas (U-Net, Mask R-CNN, inpainting), no fue posible eliminarlos sin perder información relevante. Afectaron negativamente la interpretabilidad visual con Grad-CAM.
- **Desbalance inicial de clases**: fue necesario aplicar undersampling estratificado para evitar sesgos.
- **Largo tiempo de entrenamiento**: algunos modelos como InceptionV3 o versiones con augmentation completa presentaron tiempos >6 horas en CPU.
- posible **sobreajuste al conjunto de entrenamiento y validación**, es decir, el modelo "memoriza" bien estos datos pero pierde capacidad predictiva en nuevos ejemplos.

A pesar de estos retos, la arquitectura final demostró ser robusta y consistente, con buen poder de generalización sobre el conjunto de pruebas.

5.5.4 Prueba con reducción de dimensionalidad con PCA

Durante la fase de ajuste de hiperparámetros se identificaron signos de saturación en las métricas de validación cruzada, donde múltiples combinaciones diferentes arrojaban valores idénticos de AUC, Recall y Specificity. Este comportamiento puede indicar un riesgo de **sobreajuste o memorización del conjunto de entrenamiento**,

afectando negativamente la capacidad del modelo para generalizar a datos nuevos.

Para mitigar este efecto, se incorporó un paso de **reducción de dimensionalidad utilizando PCA (Análisis de Componentes Principales)**. PCA es una técnica estadística que transforma el espacio original de características en un conjunto reducido de componentes ortogonales que explican la mayor parte de la varianza de los datos. Este procedimiento permite:

- Reducir la **complejidad del modelo**
- Disminuir el riesgo de **overfitting**
- Mejorar la capacidad de **generalización**

De acuerdo con Jolliffe y Cadima [40], PCA es especialmente útil en contextos donde se trabaja con una alta cantidad de variables correlacionadas, como es el caso de vectores de características obtenidos por redes convolucionales.

A continuación se presentan los resultados de tres experimentos utilizando PCA, con diferentes niveles de reducción:

Tabla 13. Comparación de resultados para diferentes valores de número de componentes (PCA)

PCA	C	gamma	class_weight	AUC	Recall	Specificity
500	19.306	0.078	balanced	0.9912	0.9648	0.9646
300	19.306	0.078	balanced	0.9905	0.9624	0.9632
100	19.306	0.085	{0: 1.0, 1: 1.5}	0.9868	0.9562	0.9528
50	20	0.1	{0: 1.0, 1: 1.5}	0.977	0.9396	0.9407

Se observa que a pesar de la reducción de dimensiones desde 1024 hasta 50, el desempeño del modelo (AUC, Recall, Specificity) **permanece elevado y relativamente estable**, especialmente con 300 y 100 componentes, lo cual indica que una parte significativa de la información fue retenida. Sin embargo, se mantiene el fenómeno de **saturación**, especialmente con 300 y 500 componentes, lo que sugiere que **aunque el modelo aprende bien los patrones dominantes, podría estar perdiendo sensibilidad a diferencias sutiles** en la data.

La ligera caída en desempeño a 50 componentes puede indicar una **posible pérdida de información relevante**, lo que sugiere que ese nivel ya compromete parte del contenido para realizar la discrimación de clases, esto se comprueba al evaluar la curva ROC_AUC que desciende de 0.91 a 0.86.

6. VALIDACIÓN DEL MODELO

6.1 Metodología de evaluación

La validación del modelo final se enfocó exclusivamente en la combinación DenseNet121 + SVM, que demostró el mejor rendimiento general durante la etapa de experimentación. Para asegurar una evaluación robusta y clínicamente relevante, se aplicaron dos estrategias complementarias:

- **División Hold-out Estratificada:** El conjunto total fue dividido en entrenamiento (64%), validación (16%) y prueba (20%), manteniendo la proporción de clases en cada subconjunto.
- **Validación Cruzada Estratificada (cv=5):** Se aplicó sobre el conjunto de entrenamiento para verificar la estabilidad del modelo y prevenir el sobreajuste.

La selección del threshold de decisión se realizó sobre el conjunto de validación, buscando maximizar el F1-score pero priorizando un alto recall para la clase positiva, debido a su importancia clínica.

6.1.1 Relación de gráficos por clasificador

Para cada clasificador se utilizaron algunas gráficas durante el entrenamiento y evaluación de los modelos, estas gráficas ayudan a identificar estabilidad del modelo, sobreajuste y rangos óptimos para la elección de los hiper parámetros. El detalle de las gráficas por clasificador se relaciona en la siguiente tabla.

Tabla 14. Relación de gráficos por clasificador para validación

Clasificador	Gráficas/Métricas clave	¿Qué muestran?	¿Para qué sirven?
fully connected (FCN)	- Curva de pérdida (loss) - Accuracy - ROC - Precision-Recall (PR)	Muestran cómo aprende el modelo durante el entrenamiento y cómo clasifica con distintas probabilidades. Las curvas muestran precisión y sensibilidad.	Detectar overfitting, ajustar arquitectura, evaluar estabilidad del aprendizaje y necesidad de regularización
KNN	- Accuracy vs. k-vecinos - ROC - Precision-Recall	Cómo cambia el rendimiento al variar el número de vecinos (k). ROC/PR muestran calidad de predicción. Revelan sensibilidad y especificidad del clasificador.	Elegir el mejor valor de k y ver si el modelo es robusto a valores cercanos.

Random Forest	- Importancia de variables - ROC - Precision-Recall - Overfitting	Qué características extraídas son más relevantes. Posible sobreajuste si accuracy en train es muy alto.	Ajustar hiperparámetros como max_depth, n_estimators, min_samples_leaf para evitar overfitting.
SVM	- Boxplot Recall por gamma y C - ROC - PR Curves - Heatmap de desempeño por hiper-parámetro	Relación entre sensibilidad y regularización. Mapas visuales permiten ubicar combinaciones óptimas.	Elegir C y gamma que maximicen F1 o recall. Evaluar compromiso entre precisión y sensibilidad.
XGBoost	- Overfitting (Train vs Validation AUC) - Feature Importance - ROC/PR	Cuánto aprende el modelo y si se está "memorizando". Qué variables son más relevantes, precisión y sensibilidad del modelo.	Ajustar max_depth, aumentar subsample, ajustar scale_pos_weight si hay overfitting.

6.2. Métricas de Evaluación e Interpretación Clínica

Para evaluar el desempeño del modelo se utilizaron las siguientes métricas:

- **Accuracy (Exactitud):**

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$$

Proporción de predicciones correctas.

- **Recall (Sensibilidad):**

$$Recall = \frac{TP}{TP+FN}$$

Proporción de verdaderos positivos detectados correctamente. En este caso, se refiere a la **capacidad de detectar pacientes con patología**. Una métrica crítica, pues un bajo recall implica **falsos negativos** (casos omitidos).

- **Precisión (Precisión):**

$$Precisión = \frac{TP}{TP+FP}$$

Mide cuántas de las predicciones positivas fueron correctas. Ayuda a reducir alarmas innecesarias por falsos

positivos.

- **F1-score:**

$$F1 = 2 * \frac{Precision * Recall}{Precision + Recall}$$

Media armónica que resume precisión y recall. Útil cuando hay un desbalance en las clases.

6.3. Resultados del modelo final

La siguiente tabla muestra el informe de clasificación del mejor modelo (con threshold óptimo):

Tabla 15. Informe de clasificación mejor modelo

Informe de Clasificación:	precisión	recall	f1-score	support
Sin Patología	0.87	0.80	0.84	313
Con Patología	0.82	0.88	0.85	312
accuracy			0.84	625
macro avg	0.85	0.84	0.84	625
weighted avg	0.85	0.84	0.84	625

El modelo entrenado con DenseNet121 + SVM logró un desempeño global aceptable en el conjunto de prueba, con una exactitud del 84%y un AUC de 0.91.

Con respecto a las métricas se puede decir que:

- **Recall (sensibilidad)** para clase con patología = 0.88, es el resultado **más clínicamente relevante**, ya que refleja la capacidad del modelo para detectar verdaderos positivos, es decir, pacientes con lesiones reales. Un valor alto en esta métrica significa **bajo riesgo de falsos negativos (FN)**, que es crítico en contextos médicos.
- **Precision (Precisión)** en la clase sin patología = 0.87, indica que cuando el modelo predice que **no hay patología**, acierta el 87% de las veces. Clínicamente, esto contribuye a **minimizar alarmas innecesarias** (falsos positivos) y reduce la sobrecarga diagnóstica en pacientes sanos.
- **F1-score** balancea la precisión y la sensibilidad con valor promedio de 0.84, lo que sugiere que el modelo mantiene un buen compromiso entre **no omitir casos graves** y **no sobre-diagnosticar.**

Otra métrica que permite evaluar de forma detallada el comportamiento del modelo es la matriz de confusión, al

mostrar cómo se distribuyen las predicciones en términos de verdaderos positivos, falsos positivos, verdaderos negativos y falsos negativos. La siguiente Figura representa la matriz de confusión del mejor modelo.

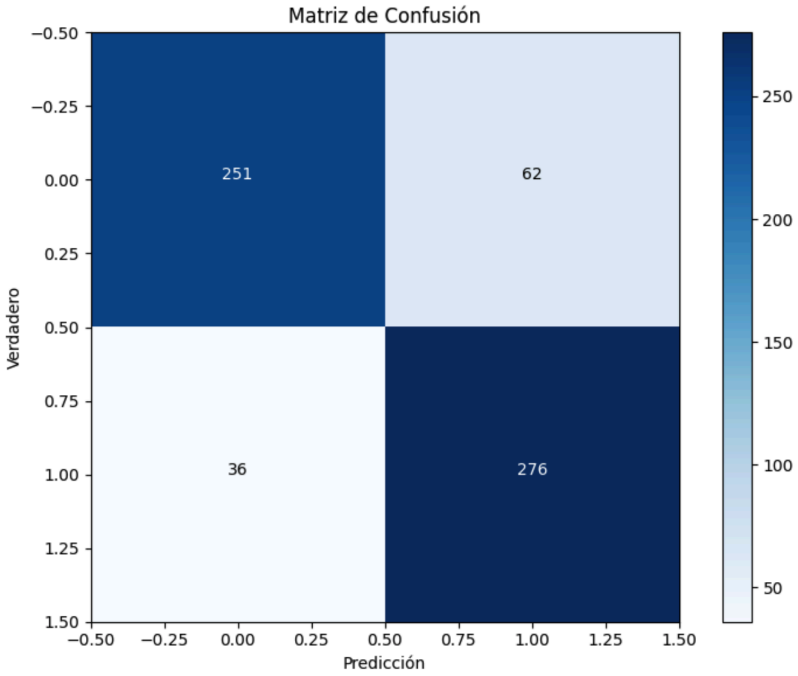


Fig 17. Matriz de confusión

- Los **falsos negativos (FN)** = 36, representan los casos con patología que el modelo no detecta, lo cual implica un riesgo clínico alto al no alertar sobre una posible lesión.
- Los **falsos positivos (FP)** = 62, pueden derivar en alarmas innecesarias o exámenes adicionales, pero son clínicamente más aceptables si se logra reducir al mínimo los FN.
- Los **verdaderos positivos (TP)** = 276, reflejan detecciones correctas de patología, y los **verdaderos negativos (TN)** = 251 aseguran que los casos normales no son sobre-diagnosticados.

6.4. Evaluación del Threshold de Decisión

En modelos que emiten **probabilidades de clase**, como el caso del clasificador SVM con probability=True, es necesario seleccionar un **umbral de decisión (threshold)** que defina a partir de qué probabilidad se clasifica una muestra como positiva (con patología). Por defecto este umbral es 0.5, pero su ajuste puede mejorar significativamente el **equilibrio entre sensibilidad (recall)** y **precisión (precisión)**, métricas clave en contextos clínicos.

6.4.1. Metodología

Se implementó una función de evaluación que prueba múltiples valores de threshold en el rango [0, 1], con incrementos de 0.01. Para cada valor se calculan:

- **Accuracy**: tasa global de aciertos.
- **Precisión**: proporción de verdaderos positivos sobre todos los predichos como positivos.
- **Recall (sensibilidad)**: proporción de positivos reales correctamente identificados.
- **F1-score**: media armónica entre precisión y recall.
- **Specificity**: proporción de negativos correctamente clasificados.

El objetivo fue identificar el umbral que maximiza el F1-score sin afectar de manera crítica el recall para la clase positiva, que representa los casos con patología.

6.4.2. Resultados

A continuación en la tabla se presentan los valores para cada métrica usando el umbral óptimo.

Tabla 16. Umbral de decisión óptimo

Threshold que maximiza F1:	value
threshold	0.250000
accuracy	0.842000
precisión	0.832685
recall	0.856000
f1	0.844181
specificity	0.828000

El umbral óptimo encontrado fue **threshold = 0.25**, como se muestra en la **Figura 18**. Este valor proporciona un equilibrio adecuado:

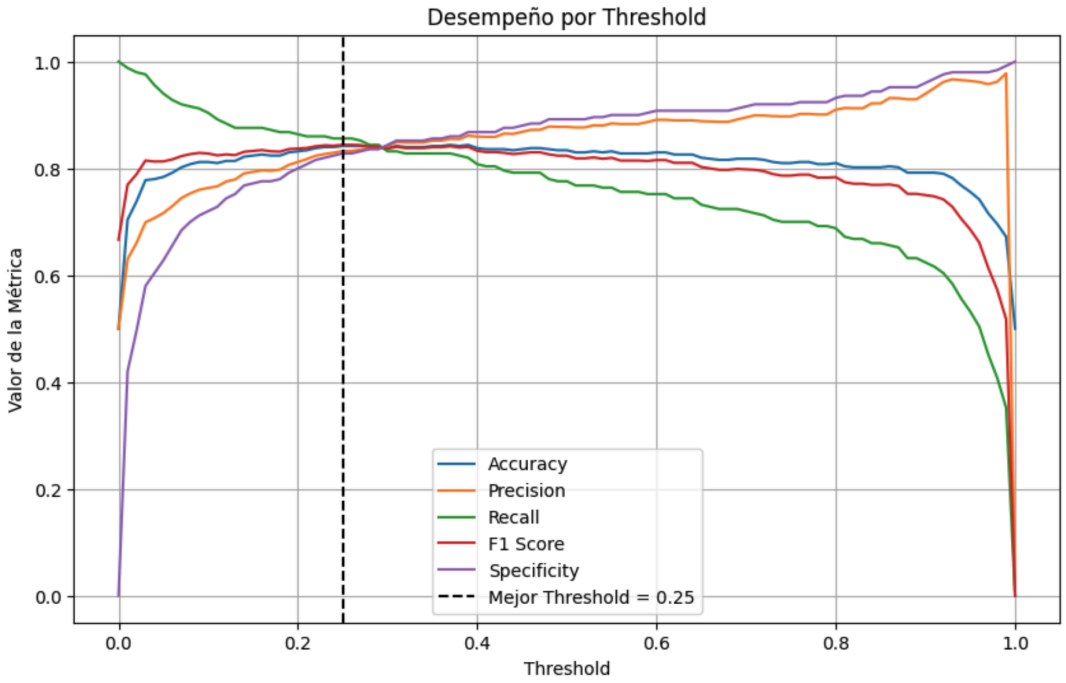


Fig 18. Desempeño de las métricas por threshold

En la figura, la línea vertical punteada indica el umbral óptimo (0.25).

Se observa que:

- A valores bajos de threshold (< 0.2), el recall aumenta considerablemente, pero a costa de una caída en la especificidad y la precisión, lo cual incrementa los falsos positivos.
- A valores altos (> 0.5), la precisión y especificidad aumenta, pero se eleva el número de falsos negativos, lo cual representa un riesgo clínico, pues el modelo dejaría de detectar casos patológicos.

Reducir los **falsos negativos (FN)** es prioritario en el tamizaje de patología cervical, ya que un error en este sentido podría llevar a omitir el seguimiento de una lesión premaligna. Por tanto, ajustar el threshold permite personalizar el modelo para que actúe con mayor sensibilidad, aunque ello implique sacrificar ligeramente la precisión.

En la siguiente tabla se muestran los resultados de cada una de las métricas al variar el umbral de decisión, lo que permite que el modelo se adapte a las necesidades clínicas o de diagnóstico automático.

Tabla 17. Comparativa de desempeño con base en el umbral de decisión.

Umbral	Accuracy	Precisión - Sin patologia (0)	Precisión - Con patologia (1)	Recall - Sin patologia (0)	Recall - Con patología (1)	F1-score (weighted avg)	TP	FP	FN	TN
0.01	0.71	0.95	0.64	0.45	0.97	0.71	140	173	8	304
0.05	0.78	0.9	0.72	0.63	0.93	0.78	198	115	21	291
0.12	0.82	0.89	0.77	0.73	0.91	0.82	227	86	27	285
0.15	0.83	0.89	0.78	0.74	0.91	0.82	233	80	29	283
0.18	0.83	0.88	0.8	0.77	0.9	0.83	241	72	32	280
0.2	0.84	0.88	0.81	0.79	0.89	0.84	246	67	35	277
0.25	**0.84**	**0.87**	**0.82**	**0.8**	**0.88**	**0.84**	**251**	**62**	**36**	**276**
0.3	0.85	0.86	0.84	0.83	0.87	0.85	260	53	41	271

El umbral con mejor equilibrio es 0.25, sin embargo, al reducirlo a 0.12 se puede lograr un recall de 91% en la clase positiva, lo cual puede ser deseable en entornos clínicos.

6.5 Interpretabilidad Clínica con Grad-CAM

Para validar si el modelo enfoca su atención en las regiones anatómicamente relevantes, se utilizó Grad-CAM (Gradient-weighted Class Activation Mapping)[39], una técnica de interpretabilidad que permite visualizar las áreas de una imagen que más contribuyeron a la decisión del modelo.

Grad-CAM genera un **mapa de calor sobre la imagen original**, resaltando con colores cálidos (rojo/amarillo) las zonas que activaron con mayor intensidad la salida del modelo. Esto resulta particularmente útil en aplicaciones médicas, ya que permite verificar si el modelo basa su decisión en **regiones clínicamente coherentes**, como la **zona de transformación del cuello uterino** o el **orificio cervical interno**.

En esta sección se presentan ejemplos representativos de:

- **Falsos Positivos (FP)**: imágenes normales clasificadas como patológicas.
- **Falsos Negativos (FN)**: imágenes patológicas no detectadas por el modelo.
- **Verdaderos Positivos (TP)**: imágenes patológicas correctamente clasificadas.
- **Verdaderos Negativos (TN)**: imágenes normales correctamente clasificadas.

Cada imagen va acompañada de su mapa de activación y análisis correspondiente, lo cual permite identificar posibles patrones de error, presencia de artefactos como reflejos especulares, o regiones mal interpretadas por el modelo.

6.5.1. Falsos positivos

En la siguiente figura se muestran tres ejemplos de imágenes que fueron clasificadas como positivas, cuando en realidad no lo eran.

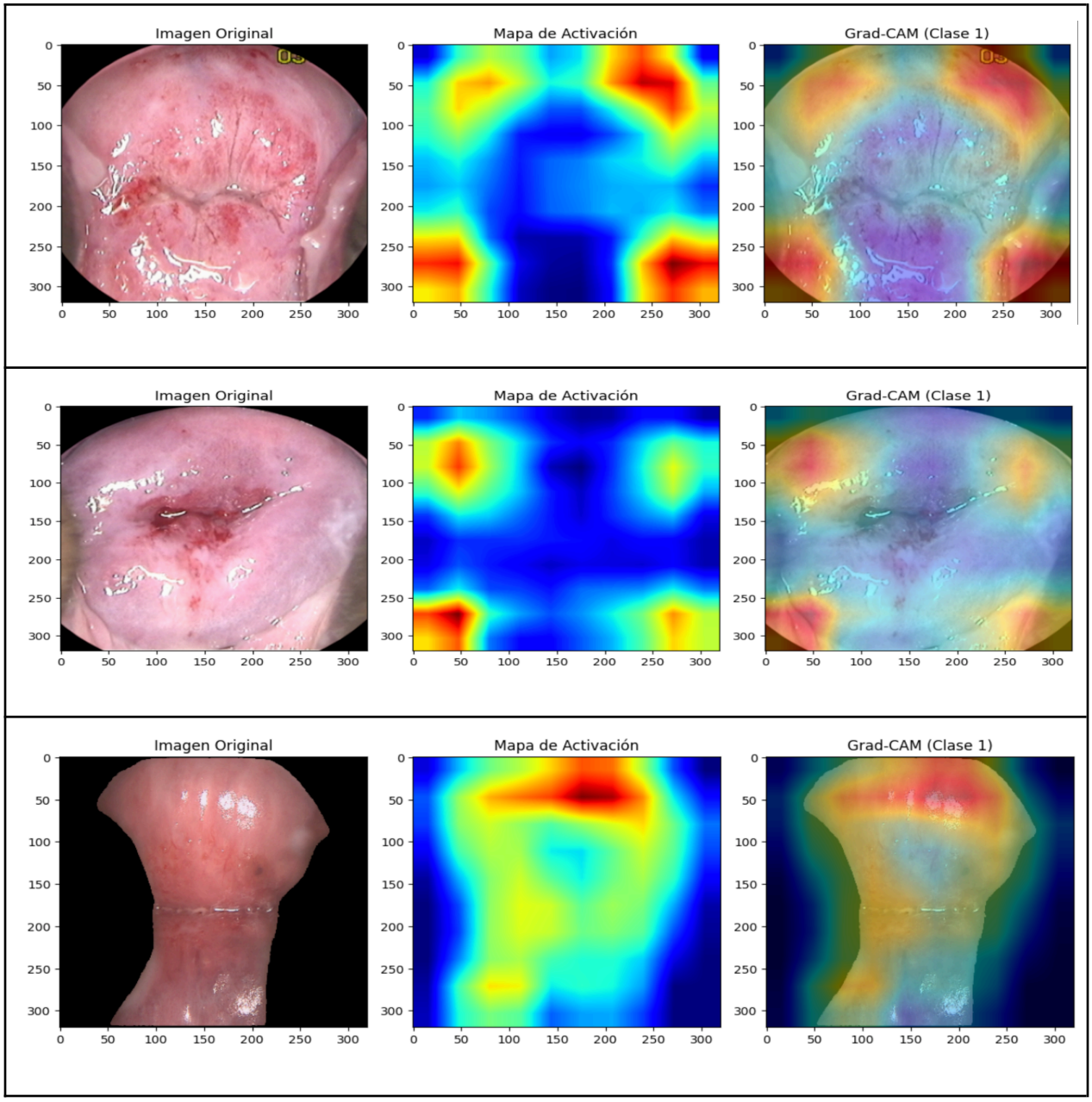


Fig 19. Grad-CAM de falsos positivos

- En la primera imagen se observa un patrón difuso, con énfasis en las regiones laterales y superiores, Grad-CAM indica que la atención al parecer se centró en reflejos especulares ubicados fuera del orificio cervical.
    - El modelo podría estar interpretando estos reflejos como regiones de interés, debido a similitudes en la textura con lesiones reales durante el entrenamiento, por ejemplo, regiones acetoblancas de la colposcopia.
- En la segunda imagen, aunque hay una zona central más homogénea, el modelo no enfoca el orificio cervical, las regiones destacadas corresponden a áreas con contraste visual fuerte, bordes brillantes.
- En la tercera imagen, el modelo concentra su atención en la parte superior, sobre una región lisa y brillante sin aparente presencia de patología.
    - Podría decirse de forma hipotética que el modelo podría estar sobre ajustado a ciertos patrones lumínicos o reflejos que aparecían en imágenes positivas durante el entrenamiento.

6.5.2 Falsos Negativos

En la siguiente figura se muestran tres ejemplos de imágenes que fueron clasificadas como negativas, cuando en realidad no lo eran.

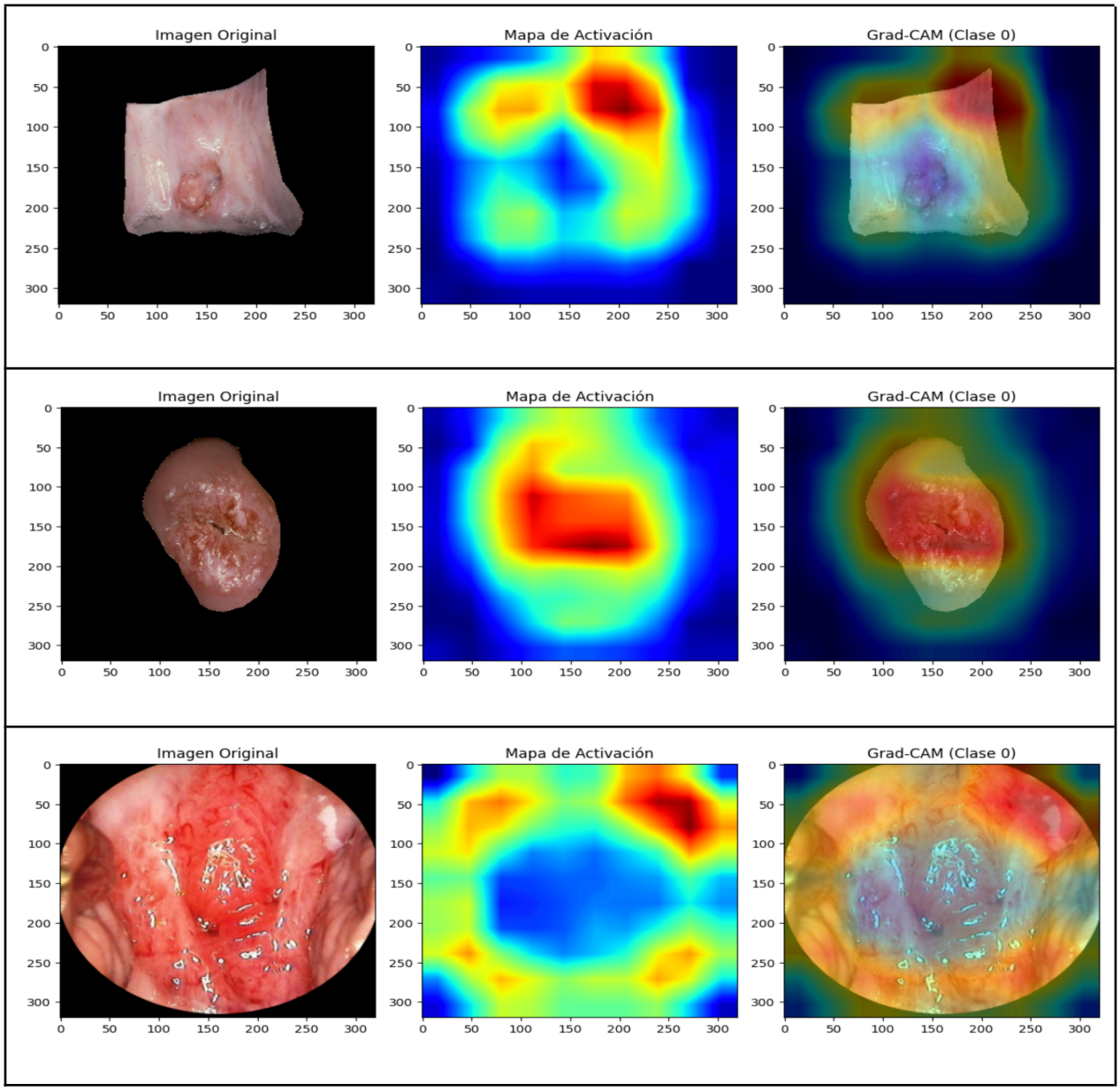


Fig 20. Grad-CAM de falsos negativos

- En la primera imagen, se observa en el centro una lesión bien definida, sin embargo el modelo no focaliza su atención en dicha zona, al parecer las zonas lisas en las cuales el modelo puso atención fueron asociadas como zonas sanas y no lo clasificó como patológico.
- En la segunda imagen, la lesión ocupa una gran porción central de la imagen, con textura rugosa y variación en el color, el mapa de activación de Grad-CAM indica que el modelo si prestó atención a la región central, pero la respuesta de activación fue difusa y cubrió toda la imagen, al parecer la probabilidad asignada no fue suficiente para superar el umbral de decisión y la clasificó erróneamente como sin patología.

- En la tercera imagen, se ven múltiples “artefactos visuales”, como reflejos, texturas del canal vaginal que podrían haber confundido al modelo, además se ve que el mapa de activación se enfocó en la zona periférica y solo de forma parcial cubrió la región cervical.

6.5.3. Verdaderos positivos

En la siguiente figura se muestran tres ejemplos de imágenes que fueron clasificadas correctamente como positivas.

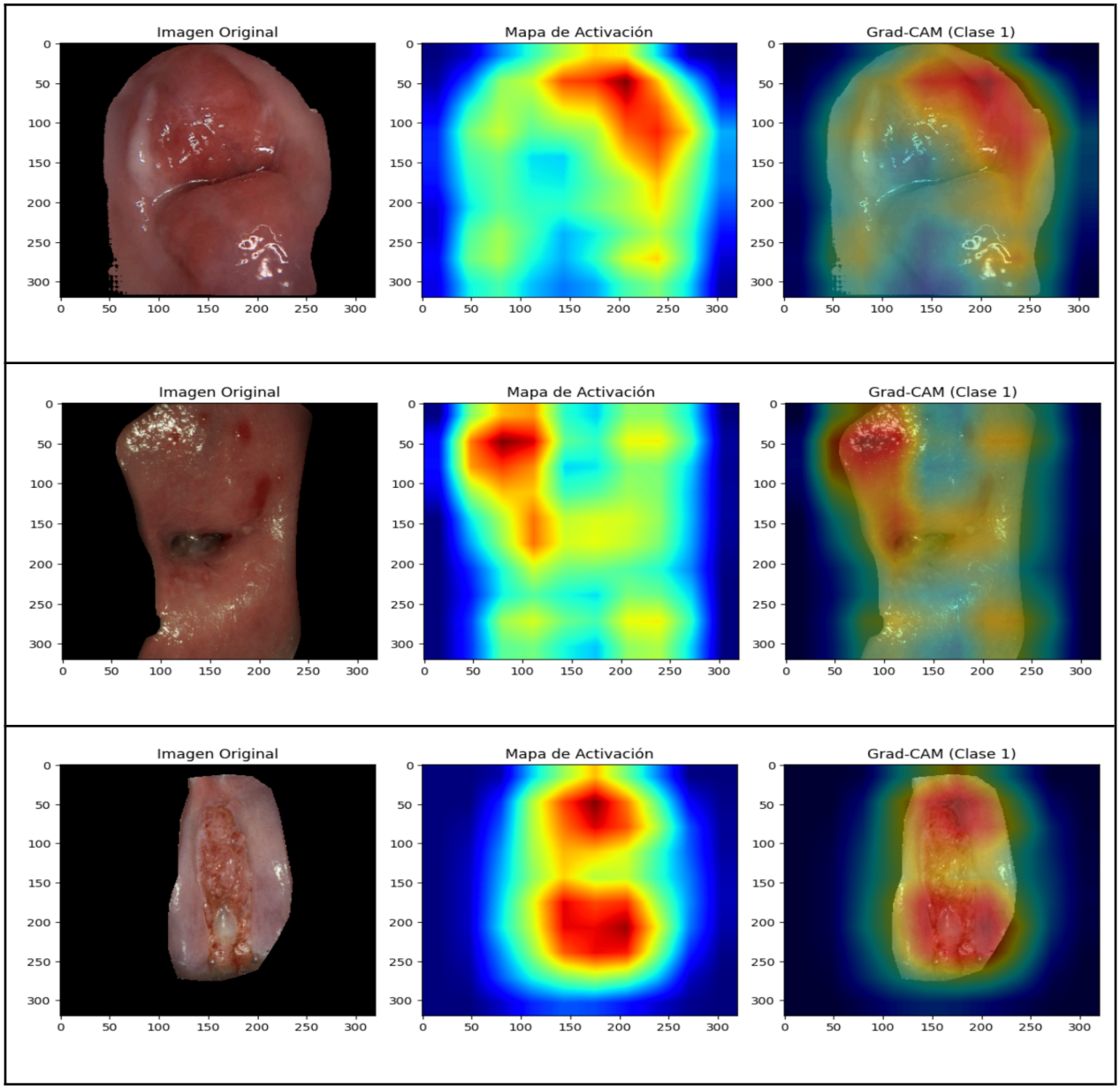


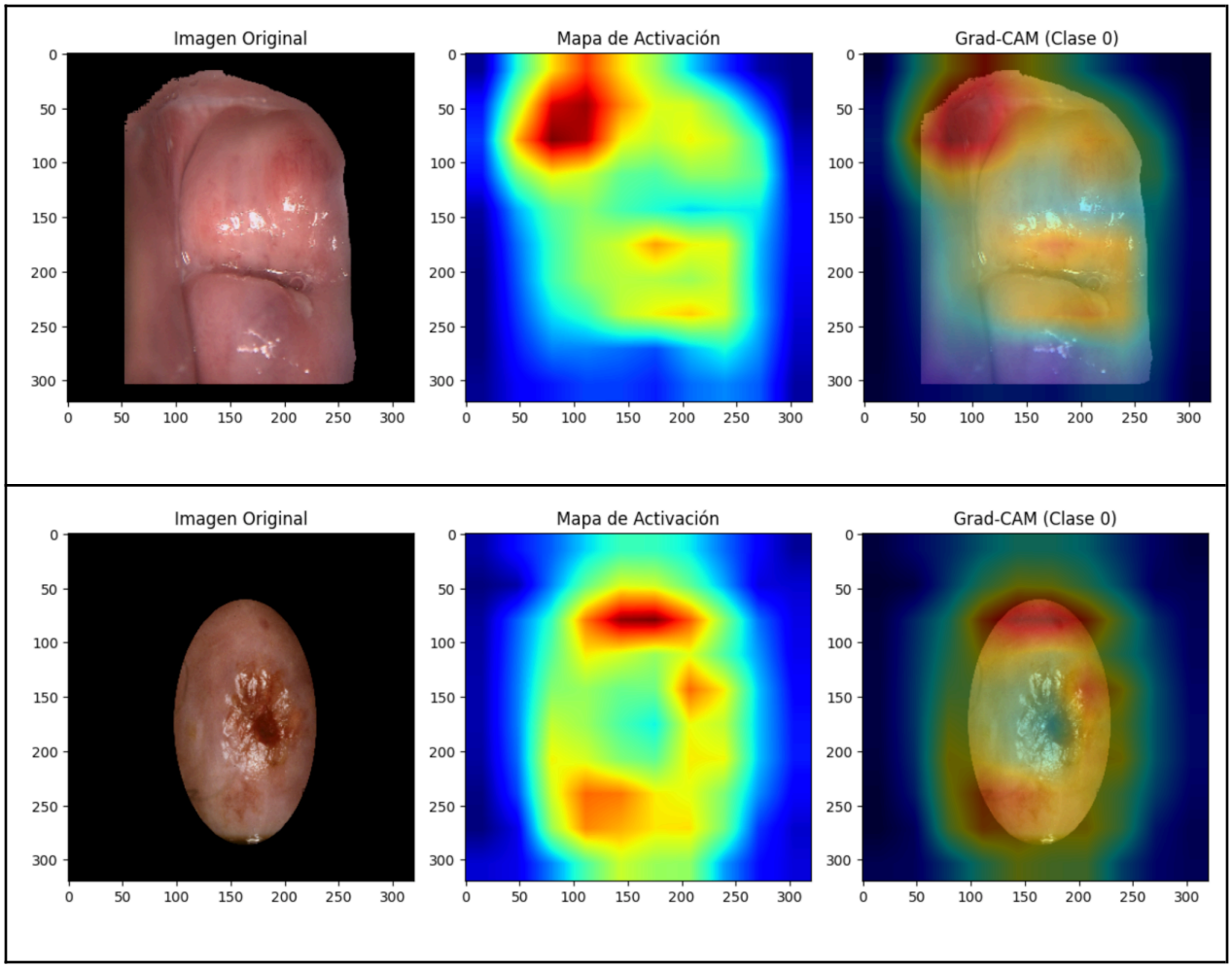
Fig 21. Grad-CAM de verdaderos positivos

En estas imágenes el modelo clasificó correctamente la presencia de patología. El análisis de activación a través de Grad-CAM permite identificar qué regiones fueron más relevantes en la decisión del modelo, en este caso para las tres imágenes se destacan los siguientes patrones:

- Zona de activación en el orificio cervical y en zonas con cambios de coloración o textura, permitiendo al modelo identificar las lesiones.
- En la primera imagen, se puede apreciar que el modelo no se vió afectado por los reflejos especulares, puede interpretarse que hubo un mejor aprendizaje en estos casos.
- En la tercera imagen se aprecia que la zona de activación tiene un fuerte foco en la región con la lesión, ubicada en el centro del cuello uterino.

6.5.4. Verdaderos negativos

En la siguiente figura se muestran tres ejemplos de imágenes que fueron clasificadas correctamente como negativas.



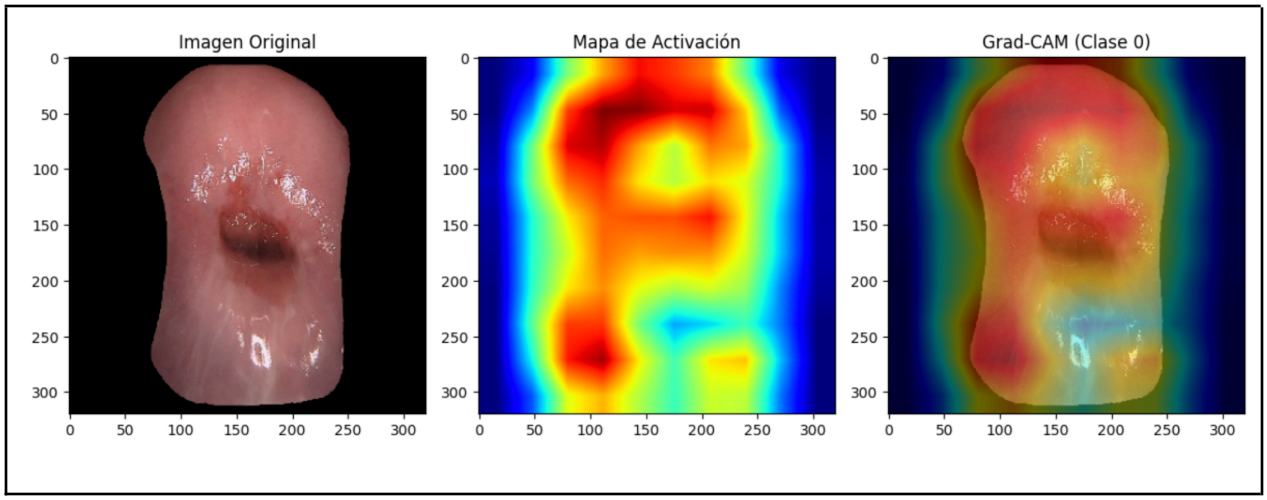


Fig 22. Grad-CAM de verdaderos negativos

- En este caso los mapas de activación tienden a desviarse del orificio cervical en las zonas de más interés (imagen 1 y 2) sin embargo, en la tercera imagen si hay una focalización clara en el orificio cervical. Esto podría indicar que el modelo está tomando decisiones basado en patrones periféricos, lo cual puede ser riesgoso, ya que en ocasiones podría estar ignorando la zona de transformación.
- Al parecer el modelo ignora zonas que podrían parecer anómalas por su color o iluminación sino presenta patrones que se asocian con patología.
- En general se ve una zona de activación periférica, lo que es común a todos los casos, es posible que al no haber presencia de zonas acetoblancas o con cambios de coloración o texturas el modelo las interprete como negativas.

7. CONCLUSIONES Y TRABAJOS FUTUROS

7.1 Conclusiones

- Objetivo General: Se logró entrenar modelos de deep learning para clasificar automáticamente imágenes de colposcopia en dos grupos: con y sin riesgo de cáncer de cuello uterino, con el fin de dar soporte al proceso de tamizaje de la enfermedad. El modelo que demostró el mejor desempeño fue la combinación de la arquitectura DenseNet121 con un clasificador SVM, alcanzando un AUC de 0.92 y un F1-score de 0.85 en el conjunto de prueba
- Gestionar una base de datos con imágenes de colposcopia debidamente etiquetadas en dos clases: normal y patológica. En el Capítulo 4 ["Gestión de la base de datos"], se describe detalladamente el proceso de integración de imágenes de tres fuentes diferentes (NIH, IARC, CITOBOT). Se logró compilar un dataset de aproximadamente 3927 imágenes. Se abordaron los retos de la heterogeneidad de las fuentes en cuanto a calidad, resolución y etiquetado. Se realizó un preprocesamiento mínimo que incluyó segmentación manual de artefactos, redimensionamiento, conversión a RGB, padding y aplicación de filtro CLAHE suave. Finalmente, se llevó a cabo un proceso de etiquetado, balanceo (mediante undersampling de la clase mayoritaria) y partición estratificada de los datos en conjuntos de entrenamiento, validación y prueba. Se validó la distribución de clases tras el balanceo y la partición. Por lo tanto, se concluye que se gestionó exitosamente una base de datos etiquetada en dos clases, abordando los desafíos iniciales.
- Entrenar algoritmos de deep learning en lenguaje Python que permitan la clasificación de imágenes de colposcopia en dos grupos con o sin riesgo de cáncer de cuello uterino. En el Capítulo 5 ["Entrenamiento de modelos de clasificación"], se exploraron diversas arquitecturas, incluyendo CNN propia y transfer learning con modelos pre-entrenados (DenseNet121, ResNet50V2, EfficientNetB3, InceptionV3 y EfficientNetV2S). Se determinó que la estrategia de transfer learning ofrecía mejores resultados. Se implementó un pipeline de entrenamiento avanzado basado en la combinación de DenseNet121 con un clasificador SVM. Este pipeline incluyó preprocesamiento, extracción de características con fine-tuning, data augmentation, normalización de características, búsqueda de hiperparámetros y selección del umbral óptimo de clasificación. Se logró entrenar un modelo de deep learning en Python capaz de clasificar imágenes de colposcopia en dos grupos. La elección de DenseNet121 se basó en su buen desempeño general, a pesar de tener un menor número de parámetros en comparación con otros modelos evaluados.
- Validar los métodos de deep learning mediante la evaluación basada en métricas de clasificación como exactitud, sensibilidad, especificidad, F1-score, y AUC-ROC. En el Capítulo 6 ["Validación de los modelos"], se evaluó el rendimiento del modelo seleccionado (DenseNet121 + SVM) utilizando una metodología de hold-out estratificado y validación cruzada estratificada (K-Fold). Sobre el conjunto de prueba, el modelo alcanzó un AUC de 0.92, un F1-score de 0.85, una exactitud de 0.85, un recall para la clase positiva (con patología) del 0.89 y una especificidad del 0.89. Se realizó un análisis comparativo con otros modelos, donde la combinación de DenseNet121 y SVM demostró un rendimiento superior. Se discutieron los resultados desde una perspectiva

clínica, resaltando la capacidad del modelo para minimizar los falsos negativos. Por lo tanto, se validó el método de deep learning, obteniendo métricas de clasificación que sugieren una buena capacidad para la identificación automática de riesgo de cáncer de cuello uterino en imágenes de colposcopia.

- Este trabajo presenta un avance en la aplicación de deep learning para la identificación automática de riesgo de cáncer de cuello uterino a partir de imágenes de colposcopia. La implementación y validación de un modelo basado en DenseNet121 y un clasificador SVM demostró resultados prometedores en la clasificación de imágenes en normales y patológicas, con un AUC de 0.92. Este desarrollo tiene el potencial de mejorar la precisión y accesibilidad del diagnóstico temprano de esta enfermedad, la cual es una de las principales causas de muerte por cáncer entre mujeres en países en desarrollo. La automatización de este proceso podría reducir la variabilidad en los diagnósticos debido a la subjetividad humana y disminuir la carga de trabajo de los especialistas, facilitando la detección en áreas con recursos limitados. Además, el uso de Grad-CAM como herramienta de interpretabilidad, aunque con limitaciones debido a artefactos en las imágenes, proporciona una visión sobre las regiones de atención del modelo.

7.2 Trabajos futuros

7.2.1 Recomendaciones Metodológicas:

- Explorar arquitecturas de deep learning más avanzadas o combinaciones híbridas que puedan mejorar el rendimiento o la generalización del modelo. Esto podría incluir investigar arquitecturas más recientes o la integración con otros tipos de modelos como transformers, tal como se menciona en la revisión de trabajos previos con el modelo híbrido para cáncer gástrico.
- Investigar técnicas de entrenamiento más sofisticadas, como el uso de learning rates adaptativos avanzados, optimizadores de última generación o estrategias de regularización más efectivas para evitar el sobreajuste.
- Evaluar el impacto de diferentes estrategias de aumento de datos, incluyendo técnicas más avanzadas o específicas para imágenes de colposcopia que preserven las características relevantes y no introduzcan artefactos.
- Considerar la incorporación de información multimodal, si estuviera disponible, como datos clínicos, resultados de pruebas de VPH o información demográfica, para enriquecer el proceso de clasificación.

7.2.2 Propuestas de Mejora en Calidad de Datos, Segmentación e Interpretabilidad

- Mejorar la calidad y diversidad del conjunto de datos, incluyendo imágenes de diferentes poblaciones, equipos de colposcopia y grados de severidad de las lesiones. Esto podría abordar la limitación de la calidad y diversidad de los datos encontrada en la discusión general. Se podría explorar la posibilidad de colaborar con más fuentes de datos o realizar campañas de recolección más amplias.
- Desarrollar técnicas de segmentación más robustas para aislar la región de interés del cuello uterino y eliminar artefactos como el espéculo de manera más efectiva, lo cual fue un reto encontrado. Lo anterior podría mejorar la calidad de las imágenes de entrada y la interpretabilidad de los modelos (por ejemplo, con Grad-CAM). Se podrían investigar arquitecturas de segmentación más avanzadas o adaptar las existentes al dominio de la colposcopia.

- Trabajar en la mejora de la interpretabilidad de los modelos para comprender mejor qué características de las imágenes son relevantes para la clasificación. Esto podría incluir la exploración de técnicas de interpretabilidad más allá de Grad-CAM o el desarrollo de métodos específicos para imágenes de colposcopia.

REFERENCIAS BIBLIOGRÁFICAS

[1] Sociedad Americana contra el Cáncer, "Cáncer de cuello uterino," 2023. [Online]. Available: https://www.cancer.org/es/cancer/tipos/cancer-de-cuello-uterino.html

[2] Organización Mundial de la Salud, "Cáncer de cuello uterino," Nov. 17, 2023. [Online]. Available: https://www.who.int/es/news-room/fact-sheets/detail/cervical-cancer

[3] Instituto Nacional de Cancer, "Detección del cáncer de cuello uterino," NIH, May 22, 2024. [Online]. Available: https://www.cancer.gov/espanol/tipos/cuello-uterino/deteccion

[4] World Health Organization, "Cervical cancer: Key facts," 2022. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/cervical-cancer

[5] J. Kim, C. M. Park, S. Y. Kim, and A. Cho, "Convolutional neural network-based classification of cervical intraepithelial neoplasias using colposcopic image segmentation for acetowhite epithelium," Scientific Reports, vol. 12, p. 17228, 2022. [Online]. Available: https://doi.org/10.1038/s41598-022-21692-5

[6] A. A. Taddese, B. C. Tilahun, T. Awoke, A. Atnafu, A. Mamuye, and S. A. Mengiste, "Deep-learning models for image-based gynecological cancer diagnosis: a systematic review and meta-analysis," Frontiers in Oncology, vol. 13, p. 1216326, 2023. [Online]. Available: https://doi.org/10.3389/fonc.2023.1216326

[7] P. Xue, J. Wang, D. Qin, H. Yan, Y. Qu, S. Seery, Y. Jiang, and Y. Qiao, "Deep learning in image-based breast and cervical cancer detection: A systematic review and meta-analysis," npj Digital Medicine, vol. 5, no. 1, p. 19, 2022. [Online]. Available: https://doi.org/10.1038/s41746-022-00559-z

[8] Instituto Nacional del Cancer, "Neoplasia intraepitelial cervical," NIH. [Online]. Available: https://www.cancer.gov/espanol/publicaciones/diccionarios/diccionario-cancer/def/neoplasia-intraepitelial-cervical

[9] F. Hashem, I. Kobagi, M. Othman, and M. Abou Ali, "Exploring data imbalance challenges in cervical cancer detection using advanced deep learning models," in Proc. 2024 Int. Conf. Smart Systems and Power Management (IC2SPM), 2024. [Online]. Available: https://doi.org/10.1109/IC2SPM62723.2024.10841356

[10] L. Jimenez-Martin, D. A. Valdés Pérez, A. M. Solares Asteasuainzarra, L. Leonard, and M. L. Baguer Díaz-Romañach, "Specular reflections removal in colposcopic images based on neural networks: Supervised training with no ground truth previous knowledge," arXiv preprint arXiv:2106.02221, 2021. [Online]. Available: https://doi.org/10.48550/arXiv.2106.02221

[11] Organización Mundial de la Salud, "Human papillomavirus (HPV) and cervical," Nov. 24, 2022. [Online].

Available: https://www.who.int/news-room/fact-sheets/detail/human-papillomavirus-(hpv)-and-cervical-cancer

[12] K. Canfell, J. J. Kim, M. Brisson, A. Keane, K. T. Simms, M. Caruana, ... and R. Hutubessy, "Mortality impact of achieving WHO cervical cancer elimination targets: a comparative modelling analysis in 78 low-income and lower-middle-income countries," The Lancet, vol. 395, no. 10224, pp. 591-603, 2020. [Online]. Available: https://doi.org/10.1016/S0140-6736(20)30157-4

[13] L. Oakden-Rayner, J. Dunnmon, G. Carneiro, and C. Ré, "Hidden Stratification Causes Clinically Meaningful Failures in Machine Learning for Medical Imaging," arXiv preprint arXiv:1909.12475, 2020. [Online]. Available: https://arxiv.org/abs/1909.12475

[14] M. Ennab and H. Mcheick, "Advancing AI Interpretability in Medical Imaging: A Comparative Analysis of Pixel-Level Interpretability and Grad-CAM Models," Machine Learning and Knowledge Extraction, vol. 7, no. 1, p. 12, 2025. [Online]. Available: https://doi.org/10.3390/make7010012

[15] Planned Parenthood, "¿Qué es una colposcopia?" [Online]. Available: https://www.plannedparenthood.org/es/temas-de-salud/cancer/cancer-cervical/que-es-una-colposcopia

[16] Ejaz Ul Haq, Q. Yong, Z. Yuan, J. Jianjun, H. Rizwan Ul Haq, and X. Qin, "Accurate multiclassification and segmentation of gastric cancer based on a hybrid cascaded deep learning model with a vision transformer from endoscopic images," Information Sciences, vol. 670, p. 120568, 2024.

[17] H. Cho, S. Lee, J. Choi, and K. Lee, "Automated diagnosis of cervical intraepithelial neoplasia in histology images via deep learning," IEEE Trans. Med. Imaging, vol. 41, no. 3, pp. 676–687, 2022. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/35204638/

[18] Centers for Disease Control and Prevention, "HPV and cancer," Dec. 13, 2022. [Online]. Available: https://www.cdc.gov/hpv/parents/cancer.html

[19] Asociación Española de Patología Cervical y Colposcopia, "Guía de colposcopia: Estándares de calidad," 2018. ISBN: 978-84-09-06631-5.

[20] A. S. Mousavi, F. Fakour, M. M. Gilani, et al., "A prospective study to evaluate the correlation between Reid colposcopic index impression and biopsy histology," Journal of Lower Genital Tract Disease, vol. 11, no. 3, pp. 147–150, 2007.

[21] National Cancer Institute, "Cáncer de cuello uterino (PDQ®)–Versión para pacientes: prevención," Dec. 13, 2023. [Online]. Available: https://www.cancer.gov/types/cervical/patient/cervical-prevention-pdq

[22] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," Proceedings of the IEEE, vol. 86, no. 11, pp. 2278–2324, 1998. [Online]. Available: https://doi.org/10.1109/5.726791

[23] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," Neural Computation, vol. 27, no. 8, pp. 1916–1929, 2015. [Online]. Available: https://doi.org/10.1162/neco_a_00990

[24] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," Medical Image Analysis, vol. 35, pp. 518–531, 2017. [Online]. Available: https://doi.org/10.1016/j.media.2017.07.005

[25] F. Lubinus Badillo, C. A. Rueda Hernández, B. Marconi Narváez, and Y. E. Arias Trillos, "Redes neuronales convolucionales: Un modelo de deep learning en imágenes diagnósticas. Revisión de tema," Revista Colombiana de Radiología, vol. 32, no. 3, pp. 5591–5599, 2021. [Online]. Available: https://doi.org/10.53903/01212095.161

[26] DataScientest, "Convolutional Neural Network: definición y funcionamiento," Dec. 16, 2023. [Online]. Available: https://datascientest.com/es/convolutional-neural-network-es

[27] A. García Hernández, "Ingeniería de las tecnologías de la telecomunicación (Trabajo Fin de Grado)," Universidad de Sevilla, 2021. [Online]. Available: https://biblus.us.es/bibing/proyectos/abreproy/93800/fichero/TFG-3800+GARC%C3%8DA+HERN%C3%81NDEZ%2C+ALBERTO.pdf

[28] Aprende e Ingenia, "¿CÓMO FUNCIONAN LAS REDES NEURONALES CONVOLUCIONALES? | Crea tu propia CNN en Python," YouTube, Jun. 7, 2021. [Online]. Available: https://www.youtube.com/watch?v=5fiBLJeAFGg

[29] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely Connected Convolutional Networks," in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), pp. 4700-4708, 2017.

[30] B.-J. Cho, J.-W. Kim, J. Park, G.-Y. Kwon, M. Hong, S.-H. Jang, H. Bang, G. Kim, and S.-T. Park, "Automated diagnosis of cervical intraepithelial neoplasia in histology images via deep learning [Preprint]," Research Square, 2022. [Online]. Available: https://doi.org/10.21203/rs.3.rs-877842/v1

[31] L. Hu, M. Schiffman, et al., "An observational study of deep learning and automated evaluation of cervical images for cancer screening," Journal of the National Cancer Institute, Jan. 10, 2019. [Online]. Available: https://doi.org/10.1093/jnci/djy225

[32] M. Bhat and T. M. S. Patil, "Adaptive clip limit for contrast limited adaptive histogram equalization (CLAHE) of medical images using least mean square algorithm," in Proc. Int. Conf. Advances in Computing, Communications

and Informatics (ICACCI), 2013, pp. 1751-1756. [Online]. Available: https://2024.sci-hub.se/3703/d4bf56c9d5d474d95cf874eef98af45c/bhat2014.pdf

[33] H. Yu, Y. Fan, H. Ma, H. Zhang, C. Cao, X. Yu, J. Sun, Y. Cao, and Y. Liu, "Segmentation of the cervical lesion region in colposcopic images based on deep learning," Frontiers in Oncology, vol. 12, p. 952847, 2022. [Online]. Available: https://doi.org/10.3389/fonc.2022.952847

[34] A. Telea, "An image inpainting technique based on the fast marching method," Journal of Graphics Tools, vol. 9, no. 1, pp. 23–34, 2004. [Online]. Available: https://www.olivier-augereau.com/docs/2004JGraphToolsTelea.pdf

[35] A. Fernández, S. García, M. Galar, R. C. Prati, B. Krawczyk, and F. Herrera, "Learning from imbalanced data sets," Springer, 2018. [Online]. Available: https://doi.org/10.1007/978-3-319-98074-4

[36] B.-J. Cho, Y. J. Choi, M.-J. Lee, J. Y. Kim, J. H. Kim, S. M. Kim, and E. Y. Park, "Classification of cervical neoplasms on colposcopic photography using deep learning," Scientific Reports, vol. 10, p. 13652, 2020. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/32788635/

[37] TensorFlow, "Keras Applications," Keras.io. [Online]. Available: https://keras.io/api/applications/

[38] Encord, "Overfitting in machine learning explained," Encord Blog, Mar. 22, 2023. [Online]. Available: https://encord.com/blog/overfitting-in-machine-learning/

[39] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual explanations from deep networks via gradient-based localization," in Proc. IEEE Int. Conf. Computer Vision (ICCV), pp. 618–626, 2017. [Online]. Available: https://ieeexplore.ieee.org/document/8237336

[40] I. T. Jolliffe and J. Cadima, "Principal component analysis: A review and recent developments," Philosophical Transactions of the Royal Society A, vol. 374, no. 2065, p. 20150202, 2016. [Online]. Available: https://doi.org/10.1098/rsta.2015.0202

[41] International Agency for Research on Cancer - World Healt Organzation, “Atlas de colposcopia - Principios y práctica” Available: https://screening.iarc.fr/atlascolpoesdetail.php?Index=028&e=,0,1,2,3,8,10,15,19,30,31,43,46,47,60,61,68,73,83,88,89,93,96,102,105,111

Anexos

Anexo A - Apéndice del capítulo 4: Gestión de la base de datos

A.1 Fuentes de datos (NIH, IARC, CITOBOT.) Unificación del dataset

A continuación se presenta el proceso para la unificación de las tres fuentes de datos.

A.1.1 Unificación y limpieza del dataset (IARC)

1. **Carga y depuración de metadatos**

   Se trabajó inicialmente con dos archivos en formato Excel:

   - Un archivo con metadatos de los casos (cases_meta_data.xlsx) que incluía el ID del caso y su diagnóstico visual con ácido acético (VIA).
   - Un segundo archivo con los nombres y tipos de imágenes (cases_images.xlsx)

   De estos dos archivos, se extrajeron únicamente las columnas relevantes: identificadores diagnóstico (CaseNumber, CaseID, VIA). Se aplicó una transformación para convertir las etiquetas de VIA en valores binarios:

   - Negative $\rightarrow 0$
   - Positive y Suspicious of cancer $\rightarrow$ 1

   Esto permitió una unificación de etiquetas para el problema de clasificación binaria.

2. **Unificación de metadatos con imágenes**

   Los datos de metadatos y de imágenes se fusionaron usando la clave CaseNumber. A su vez, se filtraron todas aquellas imágenes que fueran del tipo "*Lugol*", ya que no eran relevantes para el análisis con ácido acético, que era el objetivo del estudio.

3. **Filtrado y copia de imágenes válidas**

   Se realizó una exploración de carpetas para buscar y copiar únicamente aquellas imágenes que:

   - Estaban referenciadas en el archivo de metadatos fusionado.
   - Existían físicamente en las carpetas de origen del banco de imágenes.

   Se creó una carpeta unificada (case_images/) donde se almacenan todas las imágenes válidas para su posterior análisis. Las imágenes que no se encontraron fueron descartadas del dataset final.

4. **Exportación del conjunto final**

   Se generó un archivo CSV (dfVia_filtered.csv) que contiene el listado de las imágenes validadas con sus respectivas etiquetas y se agregó una nueva columna Source = 'VIA' para poder rastrear el origen de los datos durante el análisis.

A.1.2 Unificación y limpieza del conjunto de datos NIH

El conjunto de datos proporcionado por el National Institutes of Health (NIH) contenía imágenes colposcópicas

etiquetadas clínicamente, almacenadas en subdirectorios organizados por estudios y acompañadas de un archivo .csv (trainingDatasetSEG.csv) con información de diagnóstico.

1. **Filtrado de rutas válidas**

   Se conservaron únicamente las entradas cuyas imágenes se encontraban en los subdirectorios training_SEG/Biopsy_Study/ y training_SEG/NHS/, ya que el resto no estaban disponibles en el entorno de ejecución.
2. **Eliminación de casos no clasificados**

   Se descartaron registros donde el valor de HPV_STATUS era igual a -1, indicando una clasificación inconclusa o desconocida. Esta limpieza inicial aseguró la validez clínica de los datos para la clasificación binaria, equivalente a eliminar valores NA.
3. **Codificación de clases**

   Se unificó la variable HPV_STATUS en una etiqueta binaria denominada HPV
   - Se asignó 0 a los casos sin evidencia de patología.
   - Se asignó 1 a los casos con clasificación de riesgo patológico (HPV_STATUS ≥ 1), incluyendo lesiones de bajo y alto grado.
4. **Verificación de disponibilidad de imágenes**

Se comprobó que cada entrada del dataset tuviera su archivo de imagen correspondiente en disco, descartando cualquier fila que hiciera referencia a archivos inexistentes. Este paso fue crítico para evitar errores en el flujo de entrenamiento del modelo.

5. **Reubicación de archivos**

Todas las imágenes válidas fueron copiadas a una carpeta centralizada (case_images/) con el fin de unificar la estructura del dataset y facilitar su posterior preprocesamiento.

6. **Estandarización del esquema final**

Se conservaron únicamente las columnas *IMAGE_ID*, *HPV* y se agregó una nueva columna Source = 'NIH' para poder rastrear el origen de los datos durante el análisis.

A.1.3 Unificación y limpieza del conjunto de datos CITOBOT

El conjunto de datos proporcionado por CITOBOT contenía imágenes colposcópicas organizadas en carpetas individuales por paciente (anonimizado), acompañadas de una hoja de cálculo (OFICIAL_BASE_DE_DATOS_LADERA.xlsx) que contenía el diagnóstico clínico confirmado por biopsia.

1. **Carga y estandarización de columnas**

   Se cargó el archivo .xlsx y se renombraron las columnas clave para estandarizar el formato. En particular:
   - "39. INTERPRETACIÓN DE HALLAZGOS DE BIOPSIA" se renombró a HPV, ya que contenía la

clasificación diagnóstica.

- "PACIENTE ANONIMA" se renombró a FOLDER, dado que era el identificador del paciente y correspondía al nombre de la carpeta que contenía sus imágenes.

2. **Filtrado de datos válidos**

   Se eliminan filas con valores nulos en la columna FOLDER, asegurando que todas las entradas tuvieran una carpeta asociada con imágenes.

3. **Codificación binaria de etiquetas**

   Se asignó la etiqueta 0 a los casos con diagnóstico NORMAL, y la etiqueta 1 a cualquier otro hallazgo patológico, como NIC1, NIC2 o Carcinoma. Esta codificación binaria refleja la condición patológica general del paciente, independientemente del grado.

4. **Unificación del esquema de datos**

   Se recorrieron las carpetas correspondientes a cada paciente (según el campo FOLDER) y se recopiló el nombre de cada archivo de imagen asociado. Esta operación permitió construir un DataFrame con los campos IMAGE_ID y FOLDER, que luego se unificó con el *DataFrame* original para asociar cada imagen con su diagnóstico.

   Se eliminaron las columnas auxiliares y se conservó únicamente:

   - IMAGE_ID: nombre de archivo de imagen.
   - HPV: etiqueta binaria del diagnóstico.
   - Source: campo adicional con valor "CITOBOT" para trazar el origen de cada imagen.

5. **Copia de imágenes a la carpeta centralizada**

   Finalmente, se copiaron las imágenes correspondientes desde sus carpetas de origen a la carpeta general case_images/, para unificar el conjunto de datos en un solo directorio accesible para el flujo de entrenamiento.

A.1.4 Combinación final del dataset y verificación de integridad

Una vez normalizadas y etiquetadas las tres fuentes de datos (IARC, NIH, CITOBOT), se realizó la unión final con el objetivo de consolidar toda la información diagnóstica y vincularla directamente con las imágenes disponibles.
El proceso se llevó a cabo de la siguiente manera:

- Se concatenaron los *DataFrames* individuales (dfVia, dfColpo, dfNIH y dfCitobot) en un único *DataFrame* llamado df_combined, conteniendo todas las imágenes y sus etiquetas binarias unificadas.
- Se generó el archivo labels.csv, el cual contiene los campos IMAGE_ID, HPV y Source. Este archivo sirve como referencia principal para el entrenamiento del modelo.
- Se verificó la correspondencia entre los registros del CSV y los archivos efectivamente presentes en el directorio case_images/. Esto permitió detectar si alguna imagen listada en el dataset no estaba físicamente

presente en la carpeta de trabajo.

A.2 Técnicas de preprocesamiento - Experimento 1: Clahe y normalización

A.2.1 Redimensionamiento de imágenes

Durante los experimentos se evaluaron dos estrategias de redimensionamiento para adaptar las imágenes colposcópicas al tamaño requerido por los modelos pre entrenados. Esta etapa fue crítica para preservar la información diagnóstica sin distorsionar las estructuras anatómicas.

1. Redimensionamiento directo (sin padding)

Se implementó una función para redimensionar directamente las imágenes al tamaño objetivo (224x224 o 320x320) utilizando el algoritmo *LANCZOS*, conocido por su calidad en escalamiento.

```
# Función para redimensionar directamente a IMAGE_SIZE sin añadir relleno
def resize_image(img, target_size=IMAGE_SIZE):
    return img.resize(target_size, Image.Resampling.LANCZOS)
```

- Ventajas: rápida, directa, compatible con todos los modelos y dependiendo de la resolución se optimiza la capacidad del modelo.
- Limitaciones: distorsión de la relación de aspecto, lo que puede alterar estructuras como el orificio cervical.

2. Redimensionamiento con relleno (preservando aspecto)

Para evitar distorsiones, se desarrolló una segunda función que primero escala la imagen manteniendo su aspecto original y luego añade bordes negros (padding) hasta alcanzar el tamaño objetivo.

```
def resize_with_padding(img, target_size=IMAGE_SIZE, fill_color=(0, 0, 0)):
    """
    Redimensiona manteniendo aspecto y rellena con bordes negros hasta target_size.
    Ideal para imágenes médicas sin distorsión anatómica.
    """
    original_size = img.size  # (width, height)
    ratio = min(target_size[0] / original_size[0], target_size[1] / original_size[1])
    new_size = (int(original_size[0] * ratio), int(original_size[1] * ratio))
    img = img.resize(new_size, Image.Resampling.LANCZOS)

    new_img = Image.new("RGB", target_size, fill_color)
    new_img.paste(img, ((target_size[0] - new_size[0]) // 2,
                        (target_size[1] - new_size[1]) // 2))
    return new_img
```

- Ventajas: mantiene la proporción original, útil en imágenes médicas donde las proporciones anatómicas son relevantes.

- Observación: las pruebas mostraron que este método ayudó a conservar la precisión de los modelos

Inicialmente se trabajó con 224x224px, resolución estándar para modelos como ResNet y DenseNet.
Posteriormente se migró a 320x320px.
Este cambio también permitió mayor detalle visual en zonas críticas (e.g. zona de transformación), mejorando los resultados del modelo sin afectar la velocidad de entrenamiento de manera significativa.

A.2.2 Filtro CLAHE adaptativo

Durante el procesamiento de las imágenes colposcópicas se empleó ***CLAHE* (Contrast Limited Adaptive Histogram Equalization)**, una técnica ampliamente usada en imágenes médicas por su capacidad para mejorar el contraste local sin amplificar el ruido.

- *CLAHE* mejora el contraste en regiones específicas de la imagen, lo que **resalta detalles anatómicos importantes** como vasos atípicos, zonas acetoblancas o el orificio cervical.
- A diferencia de histogramas globales, *CLAHE* divide la imagen en pequeñas ventanas (tiles) y aplica ecualización local, lo que se puede ajustar con el parámetro GridSize.
- Su parámetro clipLimit evita sobresaturar el contraste en regiones homogéneas.

Aplicación en el espacio YCrCb

Aunque el espacio **LAB** es común en tareas de normalización de color, en este caso se optó por trabajar sobre el espacio **YCrCb** debido a:

- **Separación más limpia entre luminancia (Y)** y crominancia (Cr, Cb), permitiendo aplicar CLAHE únicamente sobre la **intensidad (Y)** sin alterar la coloración original.
- **Resultados más estables visualmente**, especialmente en tejidos con coloraciones rojizas (importantes para el diagnóstico colposcópico), los cuales tendían a tornarse marrones con LAB + CLAHE.

Implementación técnica

1. La imagen se redimensiona con padding para mantener proporción anatómica.
2. Se convierte de RGB a **YCrCb**, extrayendo los canales Y (luminancia), Cr y Cb (crominancia).
3. Se aplica CLAHE solo al canal Y, con un clipLimit **adaptativo**, determinado por:
    - **Entropía**: mide la cantidad de información visual.
    - **Coeficiente de variación**: relación entre desviación estándar y media.

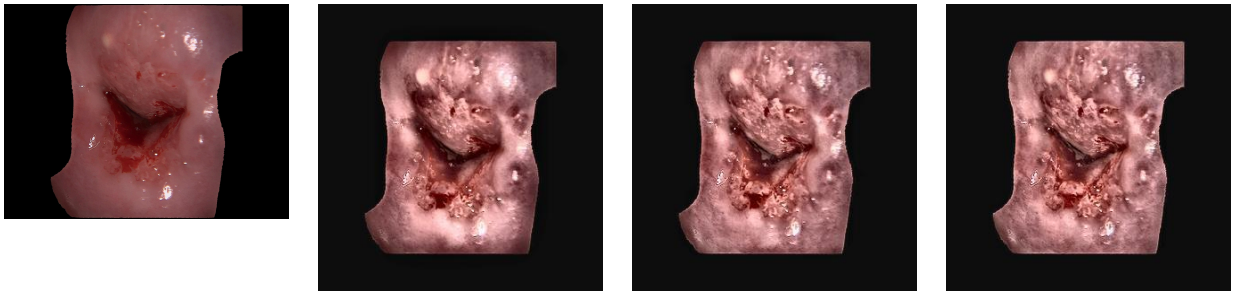
Esto permite adaptar dinámicamente el grado de mejora en función de la calidad original de la imagen.

```python
# Función para calcular un clipLimit adaptativo
def adaptive_cliplimit(img):
    img_np = np.array(img.convert('L'))  # Convertir a escala de grises
    hist = cv2.calcHist([img_np], [0], None, [256], [0, 256])
    hist = hist.ravel() / hist.sum()

    # Medir la entropía de la imagen
    image_entropy = entropy(hist)

    # Medir el coeficiente de variación (std / mean)
    std_dev = np.std(img_np)
    mean_val = np.mean(img_np)
    cv_ratio = std_dev / mean_val if mean_val != 0 else 0

    # Definir clipLimit en función de los valores calculados
    if image_entropy < 4.5 or cv_ratio < 0.5:
        clipLimit = 3  # Imagen de bajo contraste
    elif image_entropy > 5.5 or cv_ratio > 0.8:
        clipLimit = 1  # Imagen con mucho detalle, no necesita mucha mejora
    else:
        clipLimit = 2  # Rango intermedio

    return clipLimit
```

```python
# Función para aplicar CLAHE adaptativo
def apply_clahe_ycrcb(img):
    img_resized = resize_with_padding(img)
    img_np = np.array(img_resized)
    ycrcb = cv2.cvtColor(img_np, cv2.COLOR_RGB2YCrCb)
    y, cr, cb = cv2.split(ycrcb)

    # Aplicar CLAHE solo en el canal Y (luminancia)
    clipLimit = 2.0  # valor fijo o adaptativo como vimos antes
    #clipLimit = adaptive_cliplimit(img_resized)
    clahe = cv2.createCLAHE(clipLimit=clipLimit, tileGridSize=(8, 8))
    y_clahe = clahe.apply(y)

    ycrcb_clahe = cv2.merge((y_clahe, cr, cb))
    img_clahe = cv2.cvtColor(ycrcb_clahe, cv2.COLOR_YCrCb2RGB)

    return Image.fromarray(img_clahe)
```

A continuación se presentan las versiones de la misma imágen con los filtros aplicados y con la variación de gridSize de 8x8, 12x12 y 16x16.

Imagen original | Clahe gridSize 8x8 | Clahe gridSize 12x12 | Clahe gridSize 16x16



Se puede apreciar el cambio en el contraste y como efecto secundario se ven cambios en la coloración de las imágenes, esto es un inconveniente ya que los tonos rojizos y anaranjados son importantes para la identificación de características relevantes para el caso de estudio.

A.2.3 Normalización de color

El objetivo de esta función es normalizar el color de las imágenes de manera que se preserve la información relevante (como tonalidades rojizas, amarillas o blanquecinas), minimizando las variaciones que generan las condiciones de captura de la imagen (tipo de luz, enfoque, distancia, balance de blancos)

- En imágenes médicas, los colores tienen valor diagnóstico, por lo cual se debe evitar alterarlos de forma agresiva, ya que puede ocultar texturas importantes como zonas acetoblancas, cambios vasculares, etc.
- Al tener varias fuentes de datos, las imágenes provienen de diferentes dispositivos y escenarios, esto genera variabilidad en exposición y color.

Descripción del método

1. **Conversión al espacio LAB.**

La imagen se transforma de RGB a LAB, donde se separan los canales de luminancia (L) y los componentes de color (A: verde-rojo, B: azul-amarillo).

2. **Análisis del rango dinámico**

Para cada canal se calcula el rango entre los percentiles 5 y 95, para evitar que los outliers extremos distorsionen la normalización.

3. **Normalización adaptativa por canal**

Se aplica una transformación lineal por canal que:

- comprime o expande el rango dinámico original a un rango reducido
- centra el canal en un valor objetivo (128 por defecto)
- preserva parte del rango dinámico con un factor (preserve_factor)

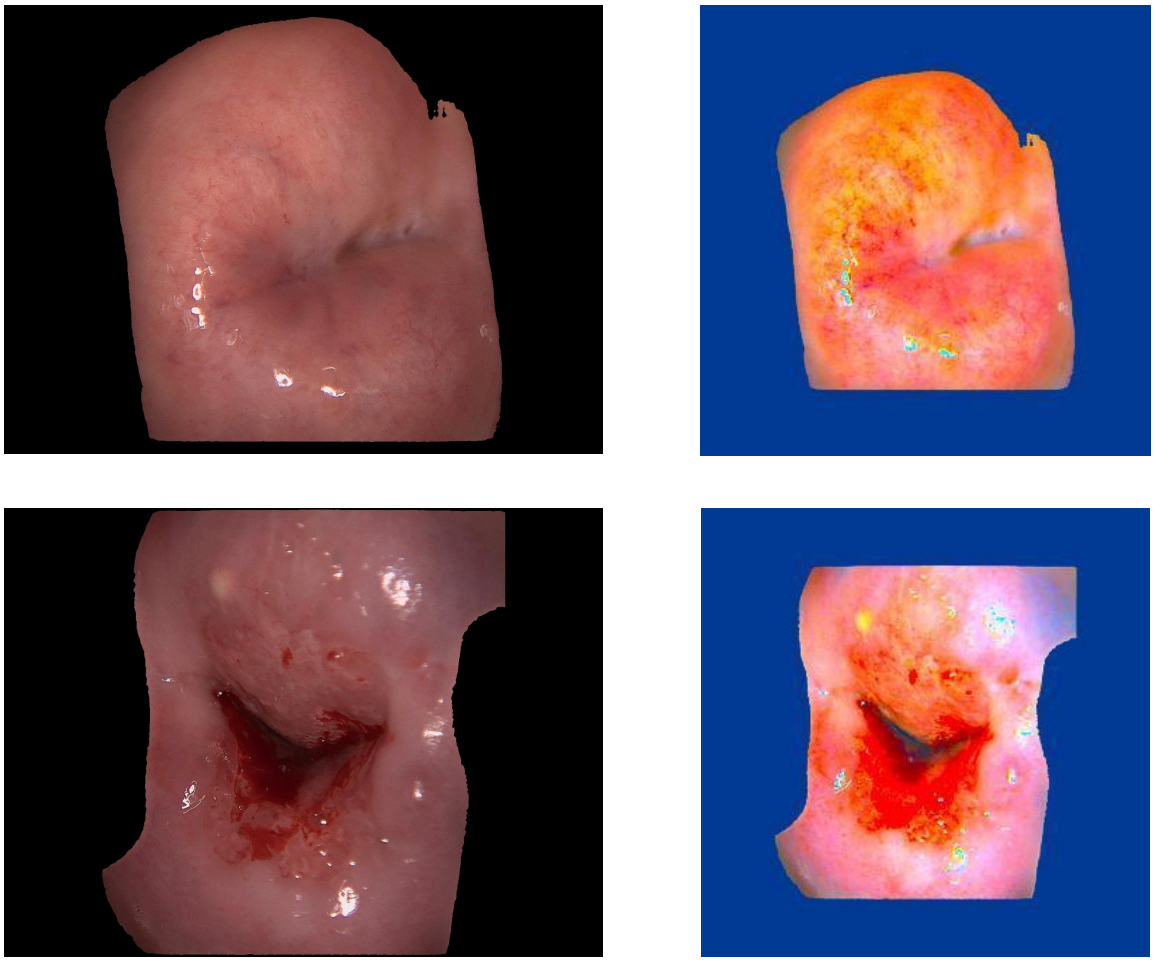
4. Reconstrucción de la imagen RGB

Los canales normalizados se combinan nuevamente y se convierten a RGB, obteniendo una imagen más consistente visualmente.

Este método buscaba reducir las diferencias de color entre imágenes heterogéneas, lo que permite mejorar la estabilidad del entrenamiento, no afecta significativamente los tonos rojizos y blancos que podrían indicar lesiones, sin embargo, el resultado no cumplió con las expectativas a la hora de obtener un mejor rendimiento del modelo, en la siguiente figura se muestran ejemplos de imágenes con y sin la normalización de color.

Uno de los principales efectos nocivos que se encontraron en este procesamiento, fue la reducción de contraste y el balance de blancos.

Imagen original **Imagen con normalización de color**



A.2.4 Reducción de reflejo especular

Uno de los fenómenos más recurrentes en las tres fuentes de datos de las imágenes de colposcopia es el de los reflejos especulares, que como se mencionó antes, se producen por la incidencia directa de la luz en superficies hùmedas o brillantes del cuello uterino y se manifiestan como regiones de un blanco intenso que no aporta información clínica relevante, pero que sí puede confundir el modelo, llevándolo a falsas activaciones o sobreajuste.
Para mitigar su efecto se implementó un algoritmo basado en el espacio de color HSV y técnicas de inpainting(Nie et al., 2023).

1. **Conversión al espacio HSV**

La imagen se convierte del espacio **RGB** al espacio **HSV** (**Hue**, **Saturation**, **Value**) donde:

- El canal **V** (valor) representa la luminosidad
- El canal **S** (saturación) permite detectar zonas con poca información cromática (e.g. reflejos)

2. **Detección del reflejo**

Se genera una **máscara binaria** que identifica como reflejo aquellas regiones que cumplen dos condiciones:

- Alta luminosidad (valor V > 230)
- Baja saturación (S < 50)

Esto busca distinguir los reflejos de otras áreas blancas naturales o de la mucosa cervical

3. **Refinamiento morfológico**

La máscara se refina mediante operaciones morfológicas

- **Apertura** (erosión + dilatación) para eliminar ruido
- **Dilatación** para asegurar cobertura completa del reflejo

4. **Relleno por inpainting**

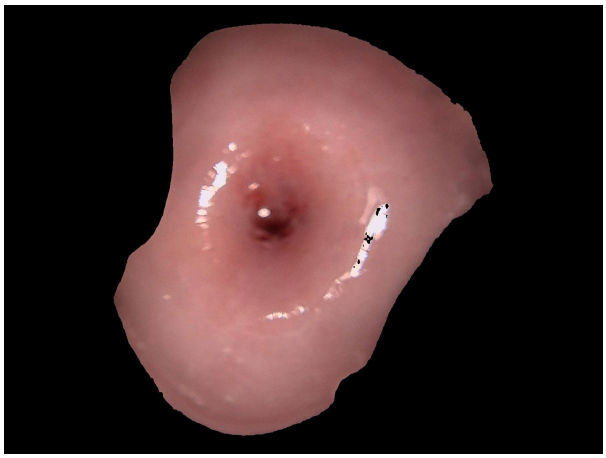
Las regiones detectadas se corrigen usando el método **TELEA** de OpenCV, que estima el contenido perdido basándose en los bordes y el contexto de la imagen (Telea, 2004).

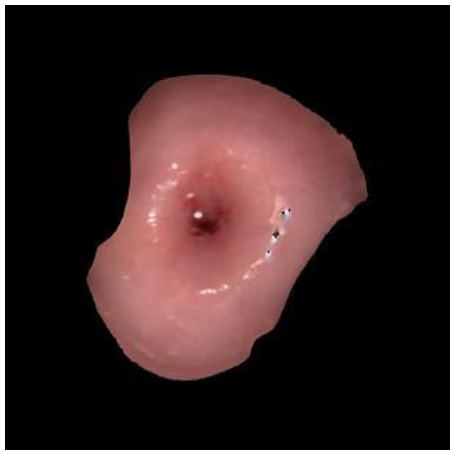
5. **Restauración del color**

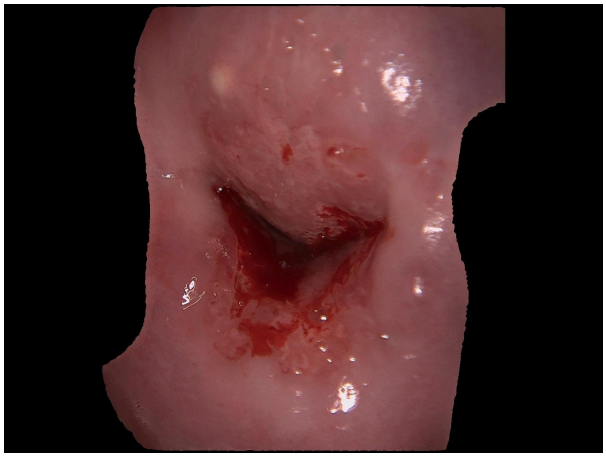
Tras el inpainting, se observó una pérdida de saturación. Para compensarlo:
Se incrementa levemente el canal Saturación (S) en el espacio HSV.
Esto mejora la visualización y mantiene un color más cercano al original sin sobresaturar la imagen.

Si bien la reducción del reflejo no fue total, si se logra reducir el brillo de aquellas áreas afectadas por este fenómeno, este procesamiento ayudó a mejorar la atención del modelo en las zona de interés, correspondiente a la zona de transformación.

Imagen original **Imagen con reducción de reflejo especular**







Anexo B - Apéndice del capítulo 5: Entrenamiento de modelos de clasificación

B.1 Caracterización de experimentos de clasificación y extracción de características

B.1.1 Extracción de características

Para representar cada imagen colposcópica como un vector numérico útil para los clasificadores, se implementó un proceso de extracción de características basado en modelos pre entrenados en ImageNet. Este proceso se compone de los siguientes pasos:

- **Carga del modelo base:** Se utiliza un modelo convolucional preentrenado (por ejemplo, DenseNet121, EfficientNetV2S, entre otros), descartando su capa de clasificación final (include_top=False). En su lugar, se emplea un esquema de *global average pooling* (pooling='avg') que resume las activaciones de la última capa convolucional en un vector de características de longitud fija.
- **Preparación del dataset:** Las imágenes son convertidas en tensores y organizadas como un tf.data.Dataset, lo que permite una carga eficiente en memoria y procesamiento por lotes (*batching*) en GPU.
- **Extracción en lotes:** El modelo procesa las imágenes por lotes para extraer las características, las cuales son vectores representativos del contenido visual de cada imagen.
- **Persistencia en disco:** Para evitar repetir la extracción en ejecuciones posteriores, los vectores de características (X_features) y sus respectivas etiquetas (y_labels) se guardan en formato .npy. Esto permite cargar los features del archivo al cambiar el clasificador y así poder comparar el desempeño de cada modelo.

La cantidad de características extraídas depende de cada modelo, el detalle se relaciona en la tabla 9.

B.1.2 División y validación del dataset

Se implementó un procedimiento sistemático para dividir el conjunto de datos en tres subconjuntos: **entrenamiento**, **validación** y **prueba,** asegurando un balance entre clases (patología y sin patología). El procedimiento se desarrolló de la siguiente manera:

1. Carga de características y etiquetas
2. Balanceo previo a la división
   a. Se aplicó *undersampling* a la clase mayoritaria para igualar el número de muestras por clase.
3. División estratificada
   a. Se usó *train_test_split* de forma estratificada.
   b. Se asignó un 20% del total al conjunto de prueba y el 80% restante fue subdividido en 80% entrenamiento y 20% validación.
   c. Esta división fue guardada en disco para asegurar su reproducibilidad.
4. Validación de distribución de clases
   a. Se imprimió la distribución porcentual de clases en cada conjunto, confirmando el balance. Esto garantiza que

los modelo no estén sesgados por desequilibrio en la clase objetivo.

5. Validación cruzada estratificada
   a. Se usó *StratifiedKFold* para aplicar validación cruzada estratificada sobre el conjunto de entrenamiento.
   b. Para cada uno de los 3 folds, se verificó que las proporciones de clases se mantuvieran similares en cada partición, validando así la estabilidad de la división y la confiabilidad del entrenamiento.

B.1.3 Procesamiento de características

Con el objetivo de asegurar que los datos de entrada estén correctamente escalados y listos para ser utilizados por los clasificadores, se implementó un sistema de preprocesamiento adaptado a cada arquitectura de modelo. Esta etapa es clave para mejorar la estabilidad numérica durante el entrenamiento y acelerar la convergencia de los algoritmos.

El proceso se resume de la siguiente manera:

1. **Identificación del modelo de extracción:** Según el modelo preentrenado utilizado para extraer las características (DenseNet121, ResNet50V2, etc.), se aplica una estrategia específica de transformación de datos.
2. **Selección de estrategia de escalado:**
   - **Normalización [0,1] (MinMaxScaler):** Utilizada para modelos que operan naturalmente en el rango [0, 1], como EfficientNetV2S, EfficientNetB3 y DenseNet121. Si las características no están ya normalizadas, se aplica MinMaxScaler para ajustar los valores entre 0 y 1.
   - **Estandarización (StandardScaler):** Aplicada para modelos como ResNet50V2, cuyas características requieren una distribución con media cero y desviación estándar uno. Se usa StandardScaler si las estadísticas de entrada no se encuentran ya cerca del estándar.
   - **Sin transformación:** Algunos modelos, como InceptionV3, ya entregan características en un rango adecuado o preprocesadas, por lo que no se aplica transformación adicional.
3. **Aplicación consistente del escalado:** Todos los escaladores son ajustados únicamente con el conjunto de entrenamiento (X_train), y luego se aplican a los conjuntos de validación (X_val) y prueba (X_test), garantizando así que no haya filtración de información entre conjuntos.
4. **Verificación de rango de valores:** Antes y después de aplicar la transformación, se imprimen los valores mínimos y máximos del conjunto de entrenamiento para verificar que la transformación fue aplicada correctamente o que ya estaba en el formato esperado.

Esta estrategia permite adaptar el flujo de trabajo a las particularidades de cada modelo de extracción y asegura la homogeneidad de las entradas al clasificador, lo cual es fundamental para obtener resultados estables y comparables.

En la siguiente figura se muestra el resultado de este procesamiento de características:

```
Aplicando preprocesamiento para el modelo: DenseNet121
Rango de valores inicial en X_train:
Min: 0.0000, Max: 17.8040
🔹 Estrategia: Normalización [0,1] (MinMaxScaler)
Rango de valores después del preprocesamiento:
Min: 0.0000, Max: 1.0000
Preprocesamiento completado.
Nuevo rango en X_train -> Min: 0.0000, Max: 1.0000
Nuevo rango en X_val -> Min: -0.0725, Max: 1.9817
Nuevo rango en X_test -> Min: -0.1612, Max: 1.9453
```

Fig. Procesamiento de características

Como se puede apreciar en la figura, existe un porcentaje fuera del rango de normalización para los conjuntos de *Val* y *Test.* Esta variación se calcula a través de la media de valores por encima de 1. Por ejemplo para el conjunto de validación sería:

print(f"El porcentaje de VAL fuera del rango de normalización : {(np.mean(X_val > 1.0) *100):.4f}%")

y el resultado es:

El porcentaje de VAL fuera del rango de normalizacion : 0.0433%

B.1.4 Búsqueda de hiperparámetros para SVM

Con el objetivo de encontrar la configuración óptima del clasificador **SVM (Support Vector Machine)**, se utilizó la herramienta GridSearchCV de scikit-learn para realizar una búsqueda sistemática de los mejores hiperparámetros.

Pasos destacados:

- **Definición del espacio de búsqueda:**

  Incluye variaciones en los parámetros C, gamma, kernel, degree y coef0, además de configuraciones para class_weight ajustadas al desbalance de clases, o en este caso para penalizar más la clase negativa.

  - C - Parámetro de Regularización
    - Controla el compromiso entre complejidad del modelo y el margen de separación.
    - Un valor bajo de C crea un margen más amplio, tolerando más errores de clasificación (modelo más general)
    - Un valor alto de C penaliza más los errores, produciendo márgenes más estrechos y posibles sobreajustes (overfitting).

    - Se evaluaron valores típicos desde 0.5 hasta 1000.
  - Gamma - Coeficiente del Kernel RBF y polinomial
    - Define la influencia de un solo punto en el entrenamiento
    - Un valor alto, significa que cada punto tiene una influencia más cercana, generando curvas de decisión más complejas (mayor riesgo de overfitting).
    - Un valor bajo, produce fronteras más suaves y generalizadas.
    - Se probaron valores como 0.05, 0.07, 0.1, 0.12, entre otros.
  - kernel - Función del núcleo
    - Define el tipo de transformación que se aplica para proyectar los datos a un espacio de mayor dimensión.
    - Se evaluaron principalmente:
      - **rbf** (Radial Basis Function): útil para fronteras no lineales complejas.
      - **poly** (polinomial): crea fronteras más flexibles, se combina con los hiperparámetros ***degree*** (grado del polinomio) y ***coef0*** (término independiente).

En las pruebas el kernel que mostró mejor desempeño siempre fue **rbf**.

- **Validación cruzada y métricas:**
  Se usa roc_auc como métrica principal para la evaluación con validación cruzada (cv=3), permitiendo seleccionar el modelo con mejor balance entre sensibilidad y especificidad.
- **Visualización y análisis de resultados:**
  Se generan gráficos de distribución del Recall según C y gamma, y se realiza un análisis del posible overfitting a partir de los resultados del entrenamiento (return_train_score=True) que permiten visualizar el rendimiento mediante:
  - Gráficos de líneas (desempeño promedio por valor de C)
  - Heatmaps (matrices AUC para combinaciones kernel vs C)
  - Boxplots por combinación de hiperparámetros

A continuación se muestra uno de los resultados de esta fase con **DenseNet121** y **SVM**

Mejor configuración encontrada:
C: 8
class_weight: {0: 1.0, 1: 2.0}
gamma: 0.1
kernel: rbf

📊 Desempeño en validación:
AUC: 0.8761 | Recall: 0.7841 | Specificity: 0.8019

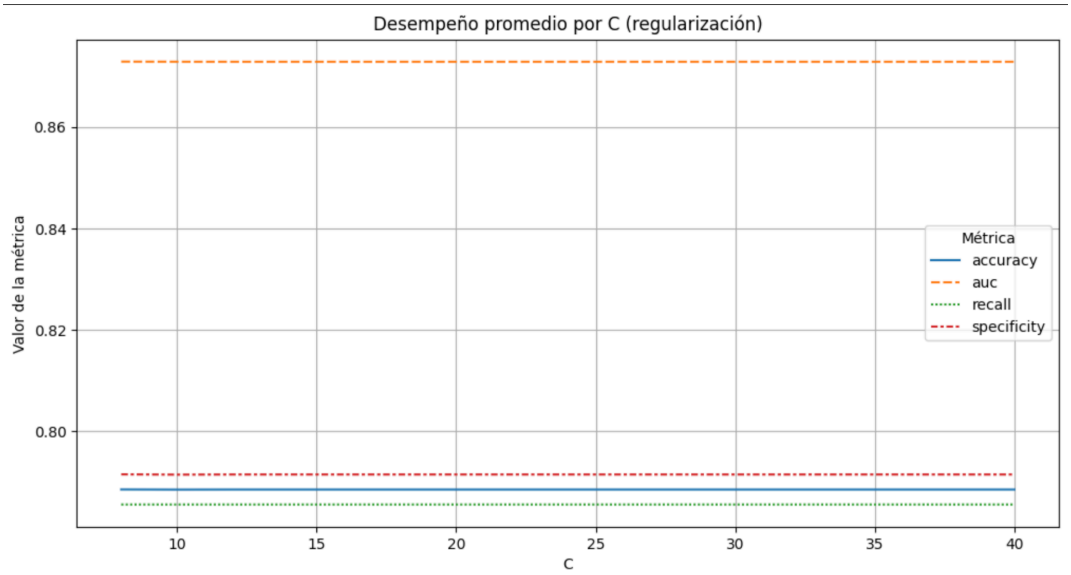


Fig. Desempeño promedio por valor de C

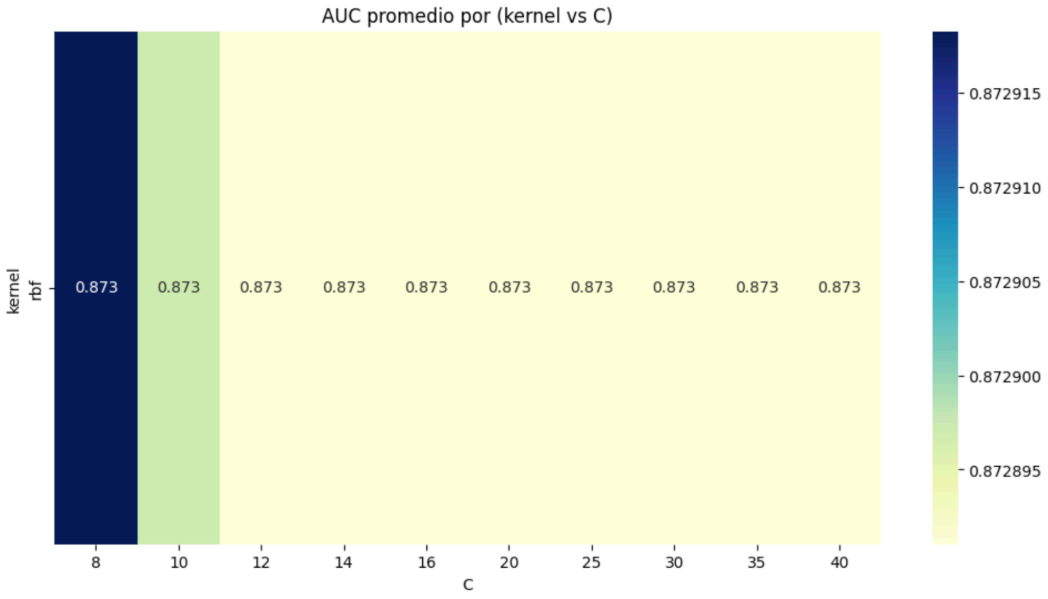


Fig. Heatmap, matrices AUC para combinaciones kernel vs C

B.1.5 Búsqueda de hiperparámetros para Random Forest

Para identificar la mejor configuración del clasificador **Random Forest**, se implementó un proceso de **búsqueda exhaustiva de hiperparámetros (Grid Search)** acompañado de **validación cruzada estratificada con 3**

particiones. El objetivo fue maximizar el área bajo la curva ROC (AUC).

Hiperparámetros evaluados:

Se exploraron múltiples combinaciones de parámetros clave del modelo:

- **n_estimators**: número de árboles en el bosque (50 a 300).
- **max_depth**: profundidad máxima de cada árbol (None, 8, 16, 32).
- **min_samples_split**: número mínimo de muestras para dividir un nodo interno (2, 4, 6).
- **min_samples_leaf**: número mínimo de muestras en una hoja (1, 2, 4).
- **max_features**: número de características consideradas para dividir un nodo ("sqrt", "log2", None).
- **class_weight**: ajuste de pesos por clase para mitigar desbalance ("balanced", "balanced_subsample", None).
- **bootstrap**: si se utilizan muestras con reemplazo en el entrenamiento de cada árbol.
- **criterion**: función para medir la calidad de una división ("gini", "entropy").

Proceso:

1. **Inicialización del modelo base** con RandomForestClassifier.
2. **Exploración combinatoria** de hiperparámetros mediante GridSearchCV, paralelizado con todos los núcleos disponibles (n_jobs=-1).
3. **Entrenamiento del modelo** para cada combinación, utilizando como métrica de selección el roc_auc.
4. **Selección del mejor modelo** basado en su rendimiento promedio en validación cruzada.
5. **Evaluación final en el conjunto de validación**, calculando métricas clave:
    - Accuracy, AUC, Recall, Specificidad

Visualizaciones:

- Tabla resumen con las 10 combinaciones con mejor desempeño.
- **Boxplot** para analizar cómo varía el Recall en función del número de árboles (n_estimators).
- **Heatmap** cruzando max_depth y n_estimators para visualizar el impacto en AUC.

La siguiente gráfica muestra ejemplo de DenseNet121 con RandomForest

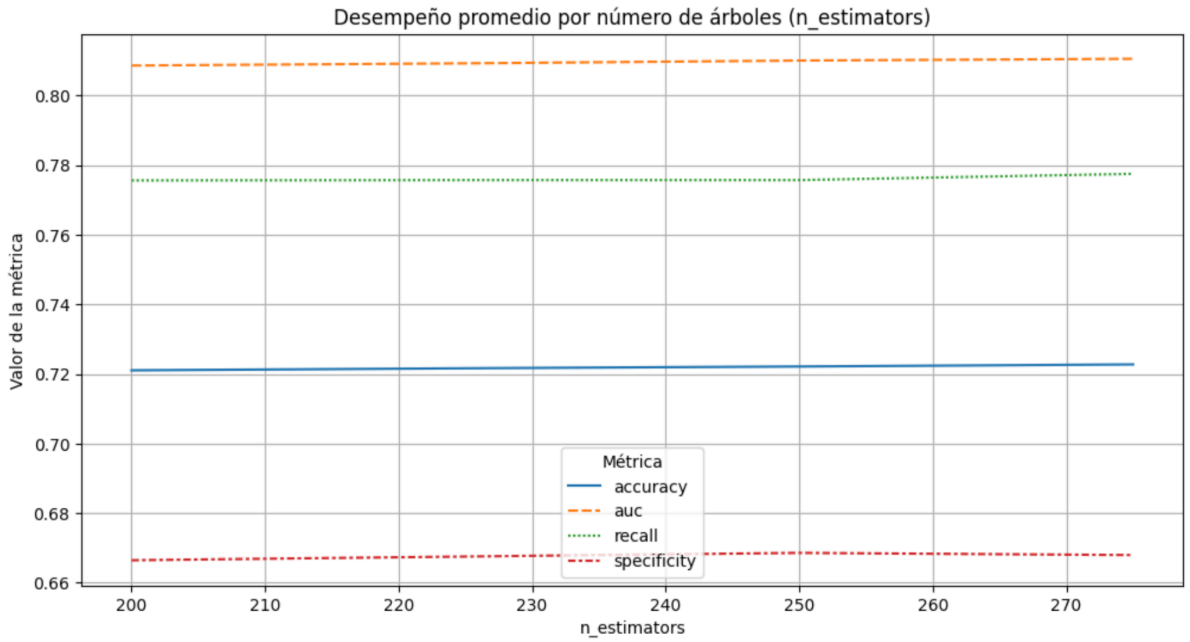


fig. Desempeño AUC por n_estimators

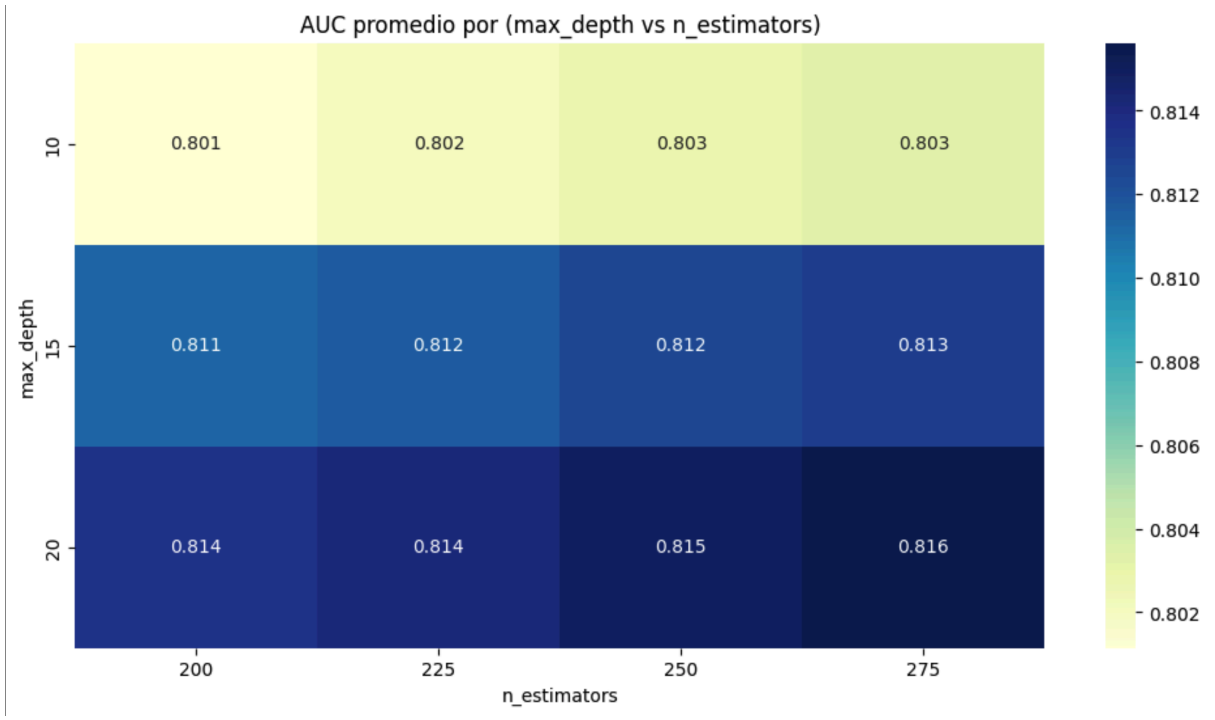


Fig.. Heatmap, valor AUC promedio por n_estimators y max_depth.

B.1.6 Búsqueda hiperparámetros KNN

Para determinar la configuración óptima del clasificador **K-Nearest Neighbors (KNN)**, se implementó una búsqueda de hiperparámetros utilizando **GridSearchCV**, enfocada en maximizar el AUC (Área Bajo la Curva ROC)

Hiperparámetros evaluados:

- **n_neighbors (k)**: número de vecinos a considerar para la predicción. Se exploraron valores entre 3 y 13.
- **metric**: función de distancia utilizada para calcular la proximidad entre instancias. Se evaluaron:
  - Euclidean, Manhattan, Chebyshev, Minkowski
- **p**: parámetro del orden de la métrica Minkowski (p=1 equivale a Manhattan, p=2 a Euclidean, etc.).
- **weights**: estrategia de ponderación de los vecinos:
  - "uniform": todos los vecinos tienen igual peso.
  - "distance": vecinos más cercanos tienen mayor peso.
- **algorithm**: estructura de datos usada para acelerar la búsqueda de vecinos (Ball Tree, KD Tree).
- **leaf_size**: tamaño de los nodos hoja para los algoritmos Ball Tree y KD Tree.

Proceso:

1. **Definición del espacio de búsqueda** con múltiples combinaciones de hiperparámetros.
2. **Búsqueda sistemática** utilizando GridSearchCV, con validación cruzada estratificada de 3 particiones.
3. **Evaluación y selección** del mejor modelo según la métrica roc_auc.
4. **Cálculo adicional de métricas clínicas** sobre el conjunto de validación:
   - Accuracy, AUC, Recall, Specificidad (calculada manualmente a partir de la matriz de confusión).

Análisis y visualización:

- Tabla comparativa por valor de k, mostrando el comportamiento del modelo en diferentes métricas.
- Gráfico de líneas para visualizar el impacto de k en el desempeño general.
- Curvas de sobreajuste (train_score vs test_score) para analizar estabilidad y capacidad de generalización.

Este análisis permitió identificar el número óptimo de vecinos y la combinación de distancia y ponderación más efectiva, adaptando el comportamiento del modelo a las características específicas del problema clínico de clasificación de imágenes colposcópicas.

A continuación se presenta el ejemplo de los resultados con el entrenamiento del modelo DenseNet121 y el clasificador KNN.

Mejor configuracion encontrada:

algorithm: ball_tree

leaf_size: 10

metric: euclidean

n_neighbors: 3

p: 1

weights: distance

Desempeno en validacion:

AUC: 0.8349 | Recall: 0.7189 | Specificity: 0.7807

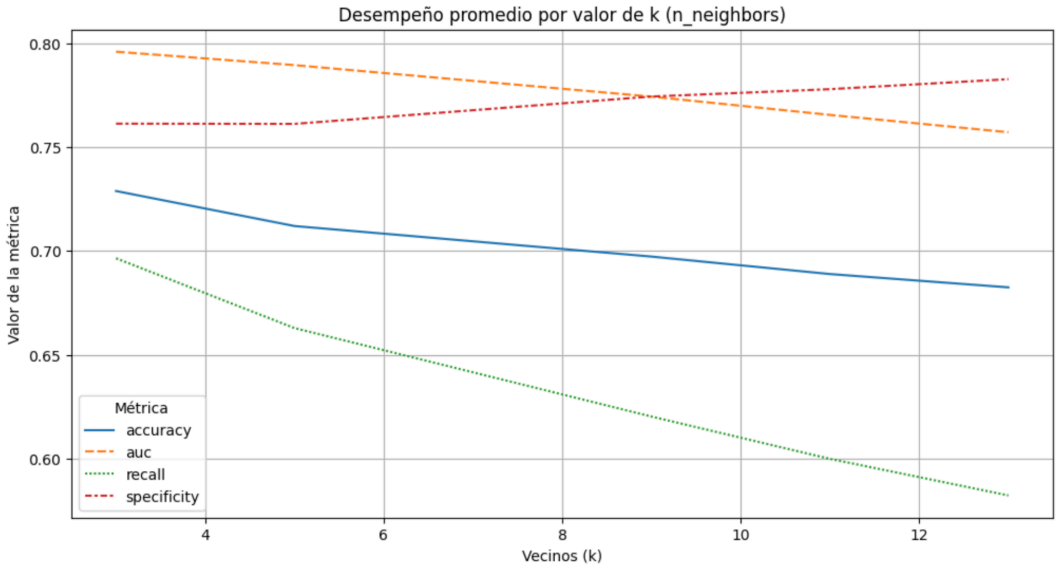


Fig. Desempeño por valor de K (vecinos)

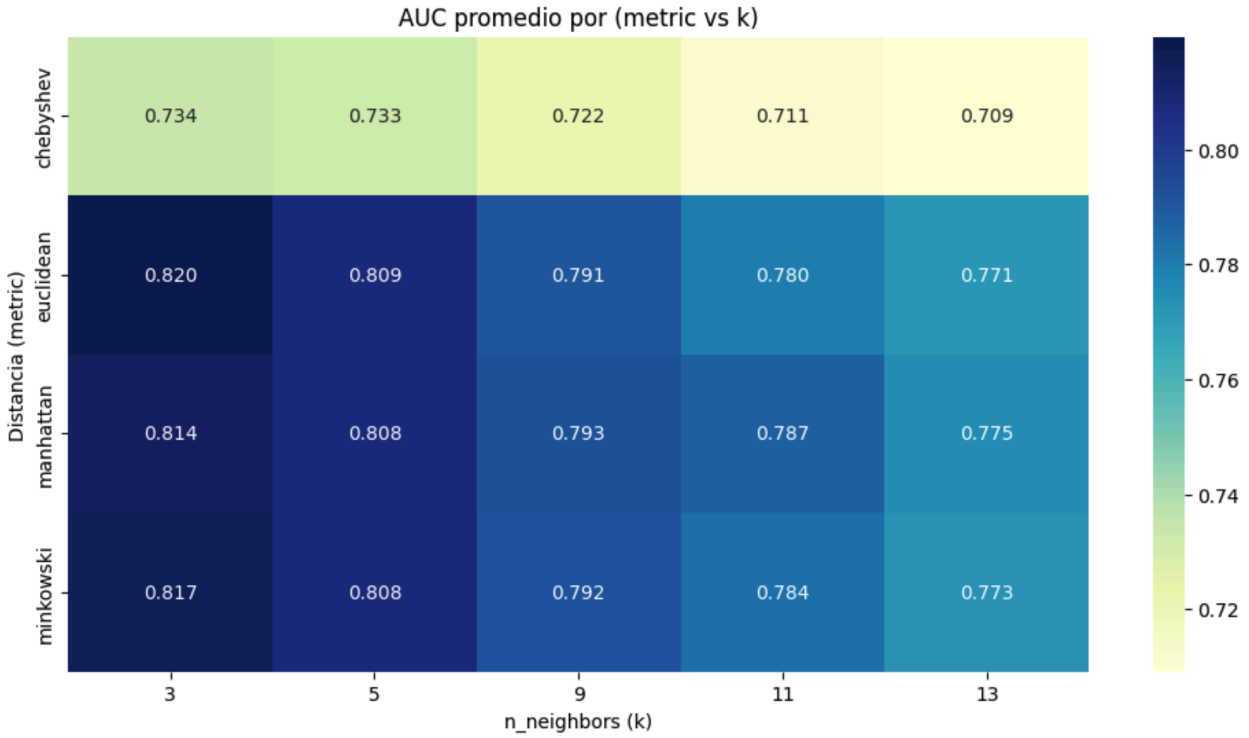


fig . Heatmap que relacion distanci y número de vecinos vs AUC

En esta gráfica se puede apreciar que se logran los mejores resultados con k=3 y distancia euclidiana.

B.1.7 Búsqueda hiperparámetros XGBoost

Se diseñó un proceso sistemático de búsqueda de hiperparámetros mediante **GridSearchCV** con validación cruzada estratificada. Este algoritmo, basado en gradiente boosting, es conocido por su alta precisión, capacidad de manejar datos tabulares y tolerancia al desbalance de clases.

Hiperparámetros evaluados:

- **n_estimators**: número de árboles en el modelo (evaluados 50, 75 y 100).
- **max_depth**: profundidad máxima de cada árbol, que controla la complejidad del modelo.
- **learning_rate**: tasa de aprendizaje, que regula cuánto se ajusta el modelo en cada iteración.
- **subsample**: proporción de datos usados por árbol para reducir sobreajuste.
- **colsample_bytree**: fracción de características usadas por cada árbol.
- **scale_pos_weight**: peso asignado a la clase minoritaria, útil para datos desbalanceados.

Proceso de búsqueda:

1. **Definición del espacio de búsqueda**: combinaciones discretas de hiperparámetros que equilibran complejidad y eficiencia.
2. **Búsqueda exhaustiva** con GridSearchCV (validación cruzada de 3 particiones).
3. **Selección del mejor modelo** basado en la métrica **AUC**, priorizando sensibilidad y balance frente a la clase positiva.
4. **Evaluación clínica** adicional sobre el conjunto de validación:
    - Accuracy, AUC, Recall, Especificidad (calculada desde la matriz de confusión).

Visualización y análisis:

- **Heatmaps** que comparan el AUC promedio entre combinaciones de n_estimators, max_depth y learning_rate.
- **Curvas de aprendizaje** adicionales para observar el comportamiento de overfitting.
- Gráficos de líneas por max_depth para entender cómo varían las métricas con la profundidad del árbol.

A continuación los resultados de uno de los experimentos, DenseNet121 y XGBoost

Mejor configuración encontrada:
colsample_bytree: 0.7
gamma: 0.1
learning_rate: 0.07
max_depth: 11
min_child_weight: 3
n_estimators: 375
reg_alpha: 0.001
reg_lambda: 1
scale_pos_weight: 2
subsample: 0.8

Desempeño en validación:
AUC: 0.8222 | Recall: 0.7544 | Specificity: 0.7358

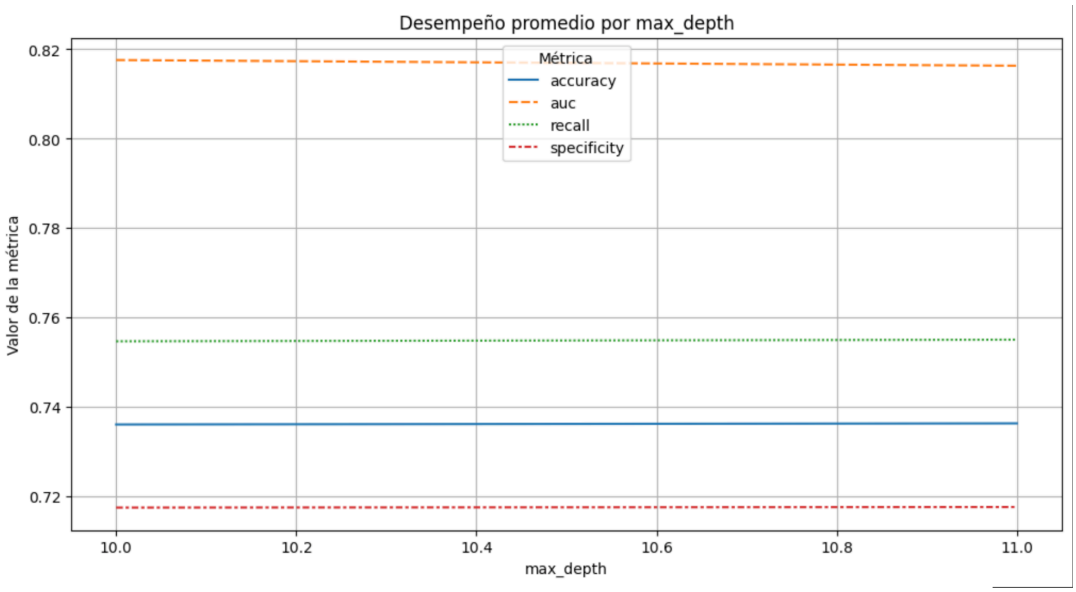


fig. metricas con base en max_depth

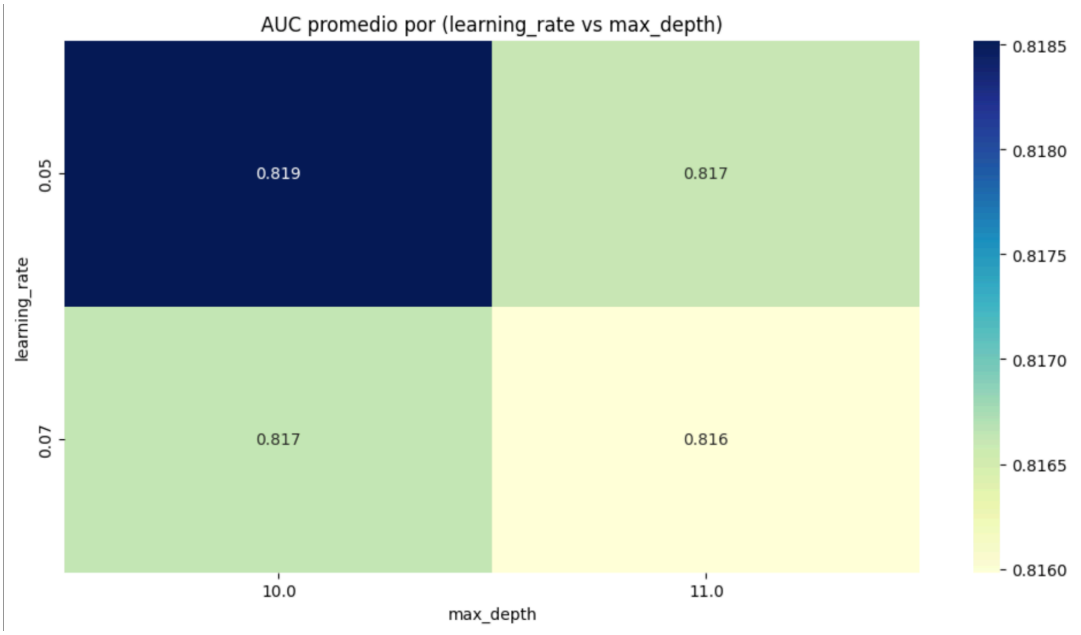


fig. Heatmap que relaciona Learning rate, Max_depth y AUC

B.1.8 Búsqueda hiperparámetros red Fully-connected

Como alternativa basada en redes neuronales tradicionales, se implementó un clasificador **Fully-Connected**

(también conocido como red densa), que recibe como entrada las características extraídas desde modelos preentrenados y aprende una función de decisión óptima a partir de ellas.

Arquitectura de la red

La red fue diseñada con una arquitectura flexible de dos capas ocultas densas, cada una seguida de capas de **Batch Normalization** y **Dropout** para mitigar el sobreajuste. La última capa es una unidad sigmoide para clasificación binaria. Esta estructura permite ajustar la complejidad del modelo y adaptarse a distintas representaciones extraídas.

- **Capa 1**: 256–512 unidades, activación seleccionable (relu, swish, mish)
- **Capa 2**: 128–256 unidades, misma activación
- **Salida**: 1 unidad con activación sigmoid
- **Regularización**: Dropout entre 20–40%
- **Normalización**: Batch Normalization tras cada capa densa
- **Función de pérdida personalizada**:
  - **weighted_binary_crossentropy** permite penalizar con mayor peso los **falsos negativos**, crucial en tareas clínicas donde es prioritario no omitir casos patológicos.

Optimización con Keras Tuner

Para encontrar la mejor combinación de hiperparámetros, se utilizó el algoritmo **Hyperband** de **Keras Tuner**, que explora de forma eficiente múltiples arquitecturas posibles bajo un criterio de validación AUC:

- **Parámetros optimizados**:
  - Cantidad de unidades por capa (units_1, units_2)
  - Función de activación
  - Tasa de dropout
  - Peso de falsos positivos y falsos negativos en la pérdida
  - Tasa de aprendizaje (learning_rate)
- **Criterio de evaluación**: Métrica de validación AUC (val_auc)
- **Detección de sobreajuste**: EarlyStopping con patience=20 y reducción dinámica de la tasa de aprendizaje

Entrenamiento y evaluación

El entrenamiento se realizó durante un máximo de 50 épocas, con evaluación sobre el conjunto de validación para seleccionar el modelo final. Se aplicó balance de clases mediante class_weight={0: 1.0, 1: 2.0} para dar mayor

importancia a la clase positiva.

Además, se implementaron métricas personalizadas como:

- Accuracy
- Precision y Recall
- Área bajo la curva ROC (AUC)
- Specificity at 90% Sensitivity

A continuación los resultado para DenseNet con FCN

Mejores Hiperparámetros Encontrados:
classifier_type: fully_connected
activation_fn: mish
units_1: 448
dropout_1: 0.2
units_2: 224
dropout_2: 0.2
fn_weight: 1.0
fp_weight: 1.0
learning_rate: 0.00015212587400254219
tuner/epochs: 50

últimas dos epocas del entrenamiento
Epoch 49/50
74/74 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.9487 - auc: 0.9887 - loss: 0.2663 - precision: 0.9567 - recall: 0.9390 - specificity_at_90_sensitivity: 0.9765 - val_accuracy: 0.7716 - val_auc: 0.8298 - val_loss: 0.5864 - val_precision: 0.7805 - val_recall: 0.7568 - val_specificity_at_90_sensitivity: 0.4576 - learning_rate: 2.4340e-07
Epoch 50/50
74/74 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.9264 - auc: 0.9832 - loss: 0.2874 - precision: 0.9204 - recall: 0.9292 - specificity_at_90_sensitivity: 0.9572 - val_accuracy: 0.7750 - val_auc: 0.8295 - val_loss: 0.5863 - val_precision: 0.7820 - val_recall: 0.7635 - val_specificity_at_90_sensitivity: 0.4542 - learning_rate: 2.4340e-07

B.1.9 Evaluación del modelo

Para evaluar el rendimiento de los clasificadores, se implementó una función que permite procesar tanto modelos entrenados con Scikit-Learn como redes neuronales definidas en Keras. Esta función realiza una evaluación integral utilizando métricas relevantes para contextos clínicos:

Funcionalidades incluidas:

- **Cálculo de predicciones y probabilidades**:
  - Si el modelo implementa predict_proba (como SVM con probability=True), se utiliza para calcular las probabilidades de clase positiva.
  - En redes neuronales, se obtiene el score directamente con .predict() y se aplica un umbral para clasificar (por defecto 0.5 o 0.52 en algunos casos).
  - Para modelos sin probabilidad, se utiliza predict() directo, con advertencia sobre la limitación en métricas probabilísticas.
- **Matriz de confusión**:
  - Muestra el número de verdaderos positivos, verdaderos negativos, falsos positivos y falsos negativos, permitiendo interpretar el desempeño en términos clínicos (p.ej., identificación errónea de una patología).
- **Informe de clasificación**:
  - Se imprime la precisión, recall, F1-score y soporte de cada clase usando classification_report() de **Scikit-Learn.**
- **Curva ROC y AUC (Área bajo la curva)**:
  - Se grafica la curva ROC para analizar la sensibilidad frente a la especificidad.
  - El valor AUC indica la capacidad discriminativa del modelo.
- **Curva Precision-Recall y Average Precision (AP)**:
  - Particularmente útil cuando las clases están desbalanceadas.
  - El área bajo esta curva ayuda a entender mejor el comportamiento del modelo ante la clase positiva.

Bondades del enfoque:

- Permite **comparar diferentes modelos** con la misma métrica.
- Incluye **métricas probabilísticas**, fundamentales para ajustar umbrales de decisión en contextos donde los falsos negativos deben minimizarse.
- La visualización de curvas ROC y PR facilita la **interpretación clínica y operativa** del modelo.

A continuación ejemplos de la visualización de estas métricas:

Tabla. Informe de clasificación

Informex de Clasificación:	precision	recall	f1-score	support
Sin Patología	0.80	0.78	0.79	370
Con Patología	0.79	0.80	0.79	369
accuracy			0.79	739
macro avg	0.79	0.79	0.79	739
weighted avg	0.79	0.79	0.79	739

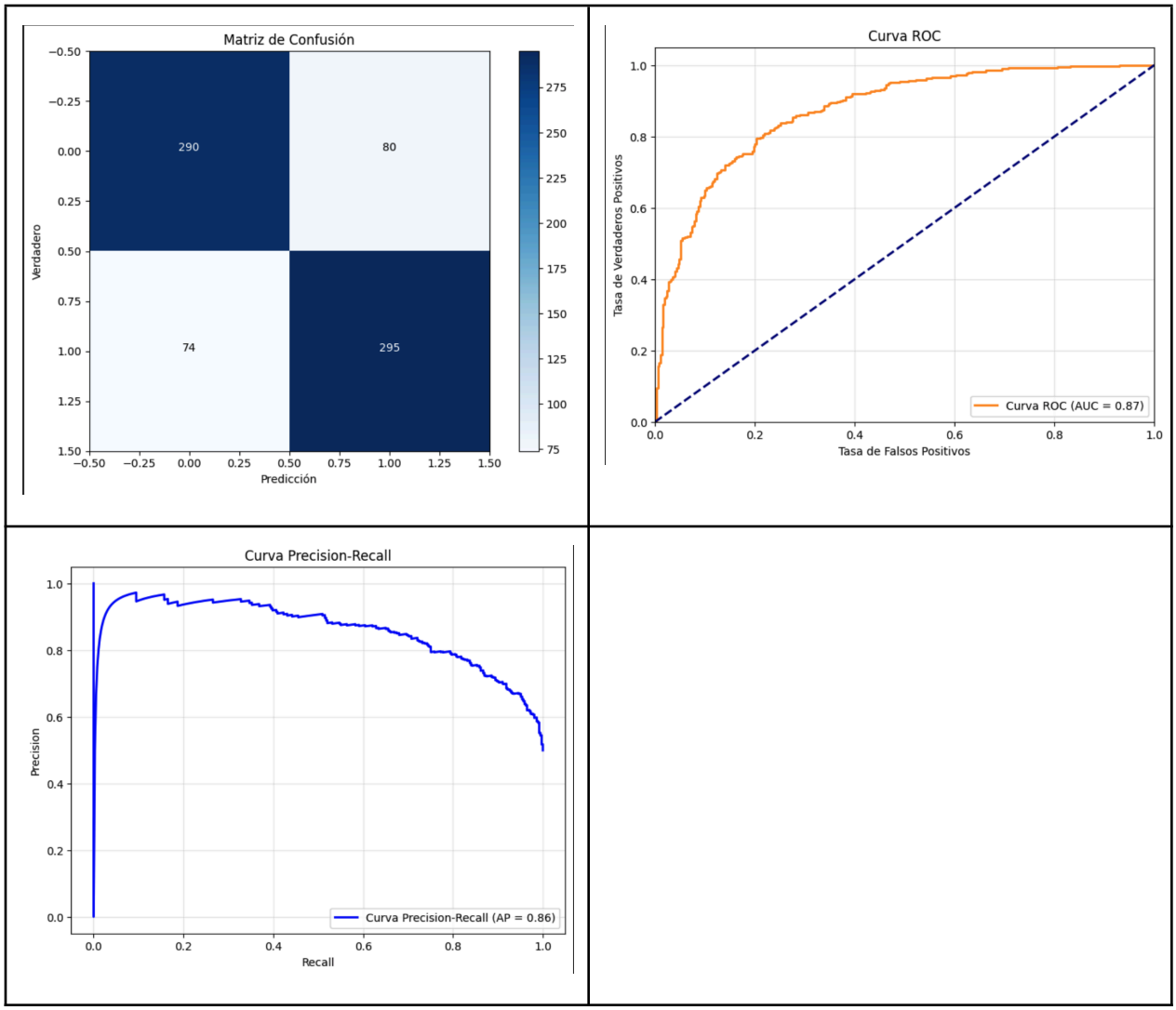


Fig. Matriz de confusión, Curva ROC, Curva PR (precision-recall)

Para los clasificadores Fully-connected se agregan además las curvas de entrenamiento:

La siguiente figura muestra la curvas para un modelo DenseNet121 y un clasificador Fully-connected:

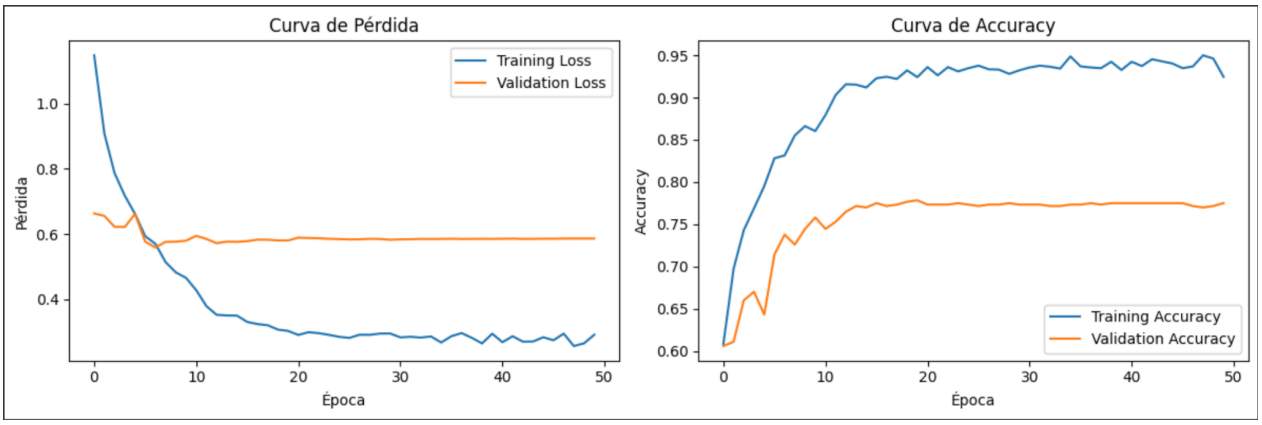


Fig. curva de pérdida y curva de accuracy

Particularmente en este ejemplo, se pueden apreciar las curvas de pérdida y la curva de Accuracy.

- **La curva de pérdida**, miden que tan bien está ajustándose el modelo a los datos de entrenamiento y validación, en este caso se puede apreciar un overfitting.
- **La curva de Accuracy**, da un indicador del porcentaje de aciertos sobre los datos de entrenamiento y validación. Nuevamente se aprecia un sobreajuste, aunque se puede ver un resultado óptimo en entrenamiento, lo que indica que el modelo está aprendiendo bien.

B.2 Data Augmentation con imágenes sintéticas

DCGANs son una clase de redes generativas adversarias (GANs) diseñadas específicamente para generar imágenes. La idea principal detrás de usar una DCGAN para data augmentation sería entrenar la DCGAN con tu conjunto de datos de imágenes de colposcopia (imágenes normales y patológicas). Una vez entrenada, la DCGAN aprendería la distribución subyacente de tus datos y podría generar nuevas imágenes sintéticas que se asemejen a las imágenes reales de tu conjunto de datos.

El funcionamiento básico de una DCGAN involucra dos redes neuronales compitiendo entre sí [No referenciado en las fuentes]:

- **Generador (Generator):** Esta red toma un vector de ruido aleatorio como entrada y trata de generar una imagen que parezca real, intentando engañar al discriminador.
- **Discriminador (Discriminator):** Esta red toma una imagen (que puede ser real del conjunto de datos o generada por el generador) como entrada y trata de distinguir si la imagen es real o falsa (generada).

Estos fueron los resultados obtenidos al usas la estrategia de data augmentation con 8000 imágenes generadas con la estrategia DCGAN.