Modelo de clasificación de imágenes para la detección de adenocarcinoma de colon

Jesús David López Sáez

Asesor

Kevin Rafael Palomino Pacheco

Universidad Nacional Abierta y a Distancia UNAD

Escuela de Ciencias Básicas, Tecnología e Ingeniería ECBTI

Tecnología en Desarrollo de Software

2025

Dedicatoria

A mi mamá que siempre me ha apoyado en todo momento y me ha animado a seguir adelante y a mi tía Soledad, que me despertó y motivó el amor por el conocimiento y la academia, desde donde esté estoy seguro que sigue animándome a seguir en este hermoso camino.

Agradecimientos

A Dios por darme la sabiduría para realizar este proyecto, a mi mamá que me ha motivado siempre a dar lo mejor de mí y esforzarme cada día por mis metas, al profesor Mario Ávila por creer en mis ideas y orientarme y al profesor Kevin Palomino por asesorarme y ayudarme en este camino complejo pero muy enriquecedor.

Resumen

El cáncer colorrectal es una enfermedad común y mortal a nivel mundial. El uso de tecnologías como la visión por computadora ayuda a reducir el tiempo de diagnóstico y los costos de tratamiento, mejorando la probabilidad de supervivencia del paciente, Este proyecto tiene como objetivo desarrollar un modelo de inteligencia artificial basado en redes neuronales para la detección de adenocarcinoma de colon mediante el análisis de imágenes médicas. Para tal finalidad, Se utilizarán herramientas y tecnologías de visión por computadora y machine learning, trabajando con datos médicos reales. Los resultados de este proyecto muestran, después de haber utilizado y probado los modelos DenseNet-121, EfficientNet_B0, GoogleNet, RegNet-y_002, Resnet-34, ResNet-50 y VGG-16, que el mejor en términos de tiempo de entrenamiento fue ResNet-50 y en términos de Accuracy y F1 fue DenseNet-121.

Palabras clave: Modelo, visión artificial, google colab, clasificación, adenocarcinoma

Abstract

Colorectal cancer is a common and deadly disease worldwide. The use of technologies such as computer vision helps to reduce diagnosis time and treatment costs, improving the probability of patient survival. This project aims to develop an artificial intelligence model based on neural networks for the detection of colon adenocarcinoma by analyzing medical images. For this purpose, computer vision and machine learning tools and technologies will be used, working with real medical data. The results of this project show, after having used and tested the models AlexNet, DenseNet-121, EfficientNet_B0, GoogleNet, RegNet-y_002, Resnet-34, ResNet-50 and VGG-16, that the best in terms of training time was ResNet-50 and in terms of Accuracy and F1 was DenseNet-121.

***Keywords*:** Model, computer vision, google colab, classification, adenocarcinoma

Tabla de Contenido

Lista de Tablas

Tabla de Figuras

Lista de Apéndices

Introducción

La inteligencia artificial (IA), que se define como la disciplina de la informática que se enfoca en la creación de sistemas capaces de realizar tareas que requieren inteligencia humana, como el aprendizaje, el razonamiento y la toma de decisiones (Russell & Norvig, 2021), se convierte cada día en una herramienta más usada en diferentes campos del saber, entre estos, en el servicio sanitario, siendo la visión artificial, rama de esta, donde se aplican procesos digitales para clasificación y detección de objetos, ampliamente utilizada, ya desde hace un tiempo se trabaja con ella en diagnóstico de diversas enfermedades, entre esas, el cancer colorrectal.

A nivel mundial, el cáncer colorrectal es una de las principales causas de mortalidad por cáncer. Según datos de la Organización Mundial de la Salud (OMS, 2020), las tasas de mortalidad varían significativamente entre países. Por ejemplo, en algunos países, las tasas de mortalidad por cáncer colorrectal son notablemente altas, mientras que en otros son más bajas. Estas diferencias pueden atribuirse a factores como el acceso a servicios de salud y programas de detección temprana.

En Colombia, según el instituto nacional de cancerología (2022), el cancer colorrectal fue el cuarto tipo de cáncer con mayor prevalencia en hombres y quinto en mujeres, con 195 y 224 nuevos casos respectivamente, y en tasa de defunciones, segundo en hombres con 56 personas y cuarto en mujeres con 52 personas, toda esta información nos muestra que esta, es una problemática de gran impacto a nivel tanto mundial como nacional, que con un diagnóstico temprano, usando las herramientas tecnológicas más recientes, como la anteriormente mencionada visión artificial, tendría unas cifras muy diferentes a las actuales, para esto ya se han hecho anteriores investigaciones sobre el tema, por dar algunos ejemplos, En 2014, (Rathore et al., 2014) desarrollaron un enfoque que combina descriptores como Histogram of Oriented

Gradients (HOG) y Color Coherence Vector (CCV) con máquinas de soporte vectorial (SVM) en un esquema de ensamblaje para clasificar imágenes de biopsias de colon. Este enfoque mejoró considerablemente la precisión diagnóstica al integrar características híbridas en el proceso de clasificación, luego, En 2020, (Iizuka et al., 2020) implementaron redes neuronales convolucionales (CNNs) y recurrentes (RNNs) para clasificar adenocarcinomas, adenomas y tejidos no neoplásicos en imágenes histopatológicas del tracto gastrointestinal. Este modelo demostró una alta capacidad para diferenciar tipos tumorales en pruebas independientes Ese mismo año, (Rodríguez-Díaz et al., 2020) desarrollaron un sistema basado en segmentación semántica profunda para diferenciar entre pólipos neoplásicos y no neoplásicos durante procedimientos endoscópicos en tiempo real, mejorando la precisión diagnóstica y la toma de decisiones clínicas.

A pesar de estas y otras herramientas desarrolladas, que aplican el uso de la inteligencia artificial a solucionar problema sanitarios, siempre se busca la disminución de la complejidad de estas tecnologías, haciendolas más simples, pequeñas y menos costosas, para así hacerlas más accesibles a un mayor número de personas y organizaciones, para lo cual mediante este trabajo se ha desarrollado, usando varios modelos preentrenados con ciertas configuraciones y el uso de la novedosa librería AISee, que nos provee un desarrollo muchísimo más sencillo, además de otras librerías convencionales como pytorch y scikit-learn, un sistema de modelos o modelo ensamblado, que permite realizar la clasificación de las imágenes médicas de colon, mediante el uso de los resultados de sus clasificaciones en conjunto para obtener un resultado final, determinando si existe o no adenocarcinoma de colon en estas.

Planteamiento del Problema

El cáncer de colon representa un problema de salud pública a nivel mundial, con una incidencia y mortalidad en aumento, especialmente en países en desarrollo. En 2019, se registraron aproximadamente 2,17 millones de casos de cáncer colorrectal en el mundo, lo que representa un aumento del 157 % desde 1990. A pesar de que en las regiones con alto índice socio-demográfico (SDI) la incidencia tiende a disminuir, en países de ingresos medianos y bajos el número de casos sigue aumentando debido a factores como el envejecimiento de la población y la adopción de estilos de vida occidentalizados. Entre los principales factores de riesgo identificados se encuentran una dieta baja en calcio y leche, el consumo de alcohol y un alto índice de masa corporal (Liu et al., 2023).

En Colombia, el cáncer colorrectal es una de las principales causas de morbilidad y mortalidad. En 2022, se registraron 11.163 casos nuevos, representando el 9,5% de todos los cánceres diagnosticados en el país. De estos, 5.455 casos correspondieron a hombres (9,7% de los cánceres masculinos) y 5.708 a mujeres (9,3% de los cánceres femeninos) (Globocan, 2022). En términos de mortalidad, el cáncer colorrectal fue responsable de 3.037 muertes en 2022, situándose como una de las principales causas de muerte por cáncer en el país (Consultorsalud, 2023). Además, según datos de la Cuenta de Alto Costo, hasta el 28 de febrero de 2023, se habían registrado 34.644 casos de cáncer de colon y recto, ocupando el tercer lugar entre los 11 tipos de cáncer priorizados en Colombia (Consultorsalud, 2023).

En Cartagena de Indias, el cáncer colorrectal representa aproximadamente el 10% del total de fallecimientos por cáncer en la ciudad (Universidad de Cartagena, 2021). Durante la pandemia de COVID-19, el diagnóstico de esta enfermedad se vio afectado significativamente debido a las diversas medidas de bioseguridad implementadas, lo que impactó en la detección

temprana y el tratamiento oportuno (Universidad de Cartagena, 2021). Toda la información anterior muestra el impacto que tiene el cáncer de colon en la salud pública y la importancia de encontrar métodos más rápidos y eficientes para su detección y tratamiento.

A lo largo del tiempo se han creado varios métodos de diagnóstico: Rayos X con Enema de Bario (1960s), Colonoscopia (1970s), Sigmoidoscopia (1980s), Prueba de Sangre Oculta en Heces(PSOH) (1990s), Colonografía por Tomografía Computarizada (Colonoscopia Virtual) (2000s), Pruebas Genéticas y Biomarcadores (2010s) y, en los últimos años, Inteligencia Artificial y Aprendizaje Automático (2020s), esta última tecnología puede hacer más rápida la detección de esta y muchas otras enfermedades, mediante el análisis de imágenes médicas de todo tipo. Este proyecto estará enfocado en responder a la pregunta problema: ¿Cómo crear un modelo de inteligencia artificial para detectar adenocarcinoma de colon?, lo cual será desarrollado utilizando algunas de las tecnologías más recientes en el campo de la inteligencia artificial.

Justificación

La importancia de este proyecto radica en los siguientes motivos:

Médico: Utilizando herramientas tecnológicas (Visión por Computadora) se disminuye el tiempo de diagnóstico de adenocarcinoma de colon, logrando así mayores probabilidades de éxito en el tratamiento del paciente, incremento en la eficiencia en el tiempo de atención a grandes grupos de pacientes con sospecha de esta enfermedad, y, en consecuencia, permite la atención a más pacientes potenciales.

Académico: La realización del proyecto me permitirá aprender, aplicar e incrementar mis conocimientos en Inteligencia Artificial, en la subrama de Visión por Computadora, utilizando datos y contexto real en el ámbito médico, lo cual será de gran beneficio para mis habilidades en el campo tecnológico.

Personal: El interés principal para desarrollar este proyecto es investigar y trabajar en un proyecto que tenga un impacto positivo importante en la sociedad, por esta razón elegí el área médica, parte fundamental para el desarrollo y calidad de vida de la sociedad.

Además de los anteriores, también estará articulado con dos de los objetivos de desarrollo sostenible: el tercero que se relaciona con salud y bienestar, ya que este proyecto impactará este sector y el noveno, relacionado con industria, innovación e infraestructura, debido a que se generará un producto nuevo a partir de la investigación realizada, teniendo relación con las políticas del actual gobierno, referentes a potenciar el uso de la inteligencia artificial a nivel nacional.

Objetivos

Objetivo General

Desarrollar un modelo de inteligencia artificial basado en redes neuronales para la detección de adenocarcinoma de colon mediante el análisis de imágenes médicas.

Objetivos Específicos

Seleccionar las herramientas y tecnologías óptimas para la creación de un modelo de visión por computadora, centrado en la detección de adenocarcinoma de colon mediante el análisis de imágenes médicas.

Comparar diferentes arquitecturas de modelos preentrenados, para desarrollar un modelo ensamblado con los modelos que obtengan los mejores resultados.

Desarrollar un modelo de inteligencia artificial basado en redes neuronales que permita el análisis de imágenes de colon, utilizando datos médicos reales para ayudar al diagnóstico del adenocarcinoma.

Marco de Referencia

Estado del Arte

Los modelos a continuación tienen características similares y han sido creados en como apoyo en el área médica para el diagnóstico de cáncer de colon, es importante recalcar que en la última década (y más aún a partir de la pandemia) ha habido un "Boom" en la aparición de sistemas de inteligencia artificial en diversos campos, más aún los creados en el campo médico, lo que ha generado competencia por los gigantes tecnológicos y un crecimiento exponencial y rápido de sus capacidades y herramientas.

En 2014, (Rathore et al., 2014) desarrollaron un enfoque que combina descriptores como Histogram of Oriented Gradients (HOG) y Color Coherence Vector (CCV) con máquinas de soporte vectorial (SVM) en un esquema de ensamblaje para clasificar imágenes de biopsias de colon. Este enfoque mejoró considerablemente la precisión diagnóstica al integrar características híbridas en el proceso de clasificación, luego, En 2020, (Iizuka et al., 2020) implementaron redes neuronales convolucionales (CNNs) y recurrentes (RNNs) para clasificar adenocarcinomas, adenomas y tejidos no neoplásicos en imágenes histopatológicas del tracto gastrointestinal. Este modelo demostró una alta capacidad para diferenciar tipos tumorales en pruebas independientes Ese mismo año, (Rodríguez-Díaz et al., 2020) desarrollaron un sistema basado en segmentación semántica profunda para diferenciar entre pólipos neoplásicos y no neoplásicos durante procedimientos endoscópicos en tiempo real, mejorando la precisión diagnóstica y la toma de decisiones clínicas

En 2021, (Schiele et al., 2021) aplicaron la arquitectura InceptionResNetV2 para analizar imágenes histológicas binarias y predecir el riesgo de metástasis en pacientes con cáncer de colon localmente avanzado. Este modelo ofrece una herramienta potencial para planificar

tratamientos y personalizar estrategias terapéuticas. Además, (Jheng et al., 2021).diseñaron el modelo GUTAID, basado en la arquitectura VGG16, para identificar y clasificar automáticamente lesiones como pólipos y cáncer en imágenes de colonoscopia, logrando una alta precisión diagnóstica.

En 2023, (Nizam et al., 2023). utilizaron redes Generative Adversarial Networks (GANs) para generar imágenes de tomografía computarizada sintéticas, que fueron utilizadas para entrenar modelos de clasificación. Este enfoque permitió mejorar el rendimiento de clasificación en el diagnóstico de cáncer colorrectal. (Ba Alawi et al., 2023) emplearon redes Fully Convolutional Network (FCN) basadas en EfficientNetB2 para reconocer y clasificar imágenes de cáncer de colon, logrando una precisión del 99.99%. Por otro lado, (Obayya et al., 2023) combinaron técnicas como Gabor filtering, GhostNet para extracción de características y Tuna Swarm Algorithm (TSA) para lograr una precisión del 99.33% en la detección de cáncer de colon y pulmón.

Finalmente, muy recientemente en 2024, (El-Aziz et al., 2024) utilizaron la arquitectura InceptionV3 optimizada con descenso de gradiente estocástico (SGD) para clasificar imágenes de cáncer de colon, alcanzando una precisión del 99.74% (Jaware et al., 2024). Asimismo, desarrollaron un modelo que fusiona arquitecturas como ResNet-101V2 y EfficientNet-B0 para el diagnóstico multi-clase de cáncer de colon y pulmón, logrando una precisión del 99.94% .

Marco Teórico

Inteligencia Artificial

La inteligencia artificial (IA) ha sido un campo de estudio fundamental en la informática, abordando la simulación de procesos de pensamiento humano en sistemas computacionales. Según Minsky (1967), "la inteligencia artificial es el estudio de lo que hacen los humanos cuando piensan". Esta definición muestra el objetivo de la IA de replicar las capacidades cognitivas humanas y aplicar esos procesos en la resolución de problemas complejos, teniendo un sin número de posibles usos en múltiples campos diferentes.

Además, Andrew Ng (2016) enfatiza la relevancia de la inteligencia artificial en el mundo actual al afirmar que "la inteligencia artificial es la nueva electricidad". Esta afirmación subraya la potencial transformación que la IA puede tener en todos los sectores de la economía y la sociedad, similar al impacto que tuvo la electricidad en la industrialización, cambiando para siempre la forma en como nos comunicamos, trabajamos, estudiamos e, incluso, pensamos, ya que genera nuevas dinámicas sociales nunca antes vistas, como sucede con la aparición de cada periodo que marca un cambio importante en la historia.

Vision Artificial

La visión artificial es un campo interdisciplinario que tiene como objetivo dotar a las máquinas de la capacidad de percibir, interpretar y actuar sobre el entorno visual de manera autónoma. Este proceso abarca desde técnicas básicas de procesamiento de imágenes hasta complejos algoritmos de inteligencia artificial que permiten a las máquinas reconocer patrones, segmentar objetos y tomar decisiones informadas. La interpretación visual no es una tarea sencilla, ya que las imágenes suelen ser ambiguas y la máquina debe ser capaz de manejar esta incertidumbre para realizar análisis precisos (Forsyth, 2018).

Además de imitar la capacidad humana de ver, la visión artificial también se enfoca en resolver problemas prácticos en diversas aplicaciones, como la robótica, la realidad aumentada y el reconocimiento facial. En estos campos, la máquina no solo necesita identificar objetos, sino también comprender y adaptarse a su entorno de manera dinámica. Esta adaptación es clave para lograr interacciones realistas y efectivas entre los sistemas automatizados y su entorno (Szeliski, 2020).

Por otro lado, la visión artificial también busca ir más allá de la simple identificación visual de objetos, aspirando a una comprensión más profunda del mundo visual. Este enfoque se basa en la creación de sistemas que no solo perciban imágenes, sino que también razonen sobre lo que ven y aprendan de sus experiencias, lo que les permite adaptarse a nuevas situaciones y contextos. Este avance hacia una visión más cognitiva e interactiva es esencial para construir máquinas que puedan tener una comprensión contextualizada y emocional del entorno visual (Li, 2017). En conjunto, la visión artificial se presenta como un campo en constante evolución que va desde la simple interpretación de imágenes hasta el desarrollo de sistemas inteligentes capaces de interactuar y adaptarse de manera autónoma al mundo visual que los rodea.

Modelo Ensamblado

Un modelo ensamblado, también conocido como ensemble model, es una técnica de aprendizaje automático que combina múltiples modelos base para mejorar el rendimiento general y la robustez de las predicciones. Esta estrategia aprovecha la diversidad de los modelos individuales para minimizar errores y reducir el riesgo de sobreajuste. Existen varias técnicas populares de ensamblado, como bagging, boosting y stacking. El bagging, representado por algoritmos como el Random Forest, mejora la precisión mediante la combinación de predicciones de varios árboles de decisión entrenados en subconjuntos diferentes de datos

(Breiman, 2001). Por otro lado, el boosting, utilizado en algoritmos como Gradient Boosting y XGBoost, entrena secuencialmente modelos débiles, corrigiendo errores en cada iteración para crear un modelo final más fuerte (Friedman, 2001).

Los modelos ensamblados se han convertido en herramientas fundamentales en competencias de ciencia de datos y en aplicaciones del mundo real debido a su capacidad para manejar datos desbalanceados y complejos. Según Zhou (2012), estas técnicas no solo mejoran la precisión, sino que también ofrecen mayor estabilidad frente a las variaciones en los datos. Además, el stacking combina las predicciones de diferentes modelos mediante un meta-modelo, maximizando la capacidad predictiva al aprovechar las fortalezas de cada modelo base (Wolpert, 1992). En áreas críticas como la medicina y las finanzas, los modelos ensamblados son valorados por su alta fiabilidad y capacidad para capturar relaciones no lineales complejas, haciendo que sean una opción preferida en tareas de clasificación y regresión.

Esta estrategia para aumentar la fiabilidad de las predicciones tiene una serie de variantes, de las cuales, para efectos de la realización de este trabajo, se eligió usar la llamada voting o voteo, como se conoce en español, la cual tiene una explicación con mayor detalle a continuación.

Voting en Modelos Ensamblados

El voting es una técnica común en modelos ensamblados que combina las predicciones de múltiples modelos base para generar una predicción final más robusta y precisa. En esencia, el voting utiliza una estrategia democrática en la que cada modelo base emite un "voto" sobre la predicción de una instancia, y la decisión final se toma en función de un criterio de mayoría (voto duro) o de promedios ponderados (voto suave). En el voto duro, la clase que recibe la mayoría de los votos es seleccionada como la predicción final, mientras que en el voto suave se

calcula la probabilidad promedio de cada clase y se elige la clase con la mayor probabilidad (Kuncheva, 2014).

El voting es especialmente útil cuando se combinan modelos base que tienen diferentes arquitecturas o perspectivas sobre los datos, ya que aprovecha su diversidad para mejorar el rendimiento general y reducir errores de generalización (Zhou, 2012). Por ejemplo, un conjunto que combina árboles de decisión, redes neuronales y máquinas de soporte vectorial puede beneficiarse de la complementariedad de estas técnicas. Además, estudios muestran que el voting mejora la estabilidad de los modelos, haciéndolos menos sensibles a cambios en los datos de entrada (Dietterich, 2000). Esta técnica es ampliamente utilizada en tareas de clasificación en áreas como reconocimiento de patrones y análisis de datos médicos, donde la precisión y la confiabilidad son críticas.

Adenocarcinoma

Para el desarrollo de este trabajo se escogió como enfermedad a diagnósticar, usando la clasificación de imágenes histopatológicas, el adenocarcinoma de colon, una enfermedad que cobra cada año muchas vidas a nivel mundial, a continuación se expone, mediante conceptos detallados, en que consiste y el por qué de la selección de esta para la creación de este trabajo.

El adenocarcinoma es un tipo de cáncer que se desarrolla a partir de las células glandulares del tejido epitelial, caracterizadas por su capacidad para secretar sustancias como moco, enzimas o fluidos. Este tipo de tumor es considerado uno de los cánceres más comunes en humanos y puede manifestarse en diversos órganos del cuerpo, donde están presentes estas células especializadas. Entre las localizaciones más frecuentes se encuentran el pulmón, colon, recto, mama, próstata, esófago y páncreas. El comportamiento clínico varía según la ubicación, la etapa en la que se diagnostique y factores individuales del paciente, como predisposiciones

genéticas y ambientales. La identificación temprana de un adenocarcinoma es fundamental para mejorar las opciones de tratamiento, que pueden incluir cirugía, quimioterapia, radioterapia o terapias dirigidas, dependiendo de la localización y la extensión del tumor (National Cancer Institute, 2023; World Health Organization, 2022).

CPU VS GPU VS TPU: Comparativa de Rendimiento

Para el desarrollo de modelos de inteligencia artificial hay un factor muy importante que incide en el rendimiento: la unidad de procesamiento. A continuación se hace un contraste con cada una de las existentes:

CPU(Unidad de Procesamiento Central)

Son más versátiles y eficientes para tareas secuenciales y procesos que requieren una lógica compleja o toma de decisiones. Su rendimiento varía en función de la cantidad de núcleos y hilos.

GPU(Unidad de Procesamiento Gráfico)

para operaciones en paralelo, como las que se encuentran en gráficos, simulaciones o entrenamiento de redes neuronales. Son muy potentes en términos de operaciones por segundo en contextos de IA.

TPU(Unidad de Procesamiento Tensorial)

Especializadas para cálculos de redes neuronales y aprendizaje profundo, las TPUs sobresalen en operaciones tensoriales, siendo más eficientes que las GPUs en ciertas aplicaciones de IA.

Indicadores de Rendimiento

En la siguiente tabla se muestran los parámetros que sirven como referencia para determinar el nivel de rendimiento de un modelo de inteligencia artificial, con la respectiva importancia de cada uno respecto al proceso evaluativo del mismo:

Tabla 1

Parámetros de Rendimiento de Modelos de Inteligencia Artificial

Indicador	Descripción	Importancia
Exactitud (Accuracy)	Proporción de predicciones correctas respecto al total de predicciones realizadas.	Es una medida general para evaluar el rendimiento de un modelo cuando las clases están equilibradas.
Precisión (Precision)	Proporción de verdaderos positivos entre todas las instancias que fueron predichas como positivas.	Es importante cuando las consecuencias de las predicciones falsas positivas son altas (e.g., detección de enfermedades).
Exhaustividad (Recall)	Proporción de verdaderos positivos entre todas las instancias realmente positivas.	Importante cuando es crucial identificar la mayoría de las instancias positivas (e.g., detección de fraude o enfermedades graves).

Indicador	Descripción	Importancia
Puntuación F1 (F1-Score)	Media armónica entre la precisión y la exhaustividad. Se utiliza para equilibrar ambas métricas en escenarios donde hay clases desbalanceadas.	Es valiosa cuando se necesita un balance entre precisión y recall, especialmente en casos de desbalance de clases.
AUC-ROC (Área Bajo la Curva ROC)	Mide la capacidad de un modelo para distinguir entre clases, representando la tasa de verdaderos positivos frente a la tasa de falsos positivos.	Útil para evaluar la capacidad discriminativa de un modelo, especialmente con clases desbalanceadas.
Log Loss (Pérdida Logarítmica)	Mide la incertidumbre de las predicciones de un modelo clasificatorio, penalizando predicciones incorrectas que están muy lejos de la clase verdadera.	Refleja la confianza de un modelo en sus predicciones, lo que es crítico en problemas de clasificación probabilística.

Indicador	Descripción	Importancia
MAE (Error Absoluto Medio)	Promedio de la magnitud de los errores en las predicciones, calculado como la media de las diferencias absolutas entre predicciones y valores reales.	Ideal para problemas de regresión donde se requiere una interpretación directa de los errores en términos absolutos.
MSE (Error Cuadrático Medio)	Promedio de los errores al cuadrado entre las predicciones y los valores reales, penalizando errores más grandes.	Importante en problemas de regresión cuando se desean penalizar los grandes errores más severamente.
RMSE (Raíz del Error Cuadrático Medio)	Raíz cuadrada del MSE, lo que lo hace más interpretable al devolver el error en la misma escala que los valores de las predicciones.	Es una métrica estándar para modelos de regresión, especialmente en escenarios donde grandes errores son costosos.
R^2 (Coeficiente de Determinación)	Mide qué proporción de la varianza en la variable dependiente es explicada por el modelo.	Indica qué tan bien se ajustan las predicciones a los datos reales en problemas de regresión.

Indicador	Descripción	Importancia
Matriz de Confusión	Tabla que muestra las verdaderas etiquetas frente a las predicciones, dividiendo los casos en verdaderos positivos, verdaderos negativos, etc.	Proporciona una visión detallada del rendimiento del modelo, permitiendo identificar dónde se equivocan las predicciones.

Nota. tabla de los parámetros de rendimiento de inteligencia artificial. Obtenido de autoría Propia.

Inteligencia Artificial Cuántica: El Siguiente paso

Los modelos de inteligencia artificial cuántica han mostrado un crecimiento notable en su aplicación al campo médico, permitiendo avances significativos en diagnóstico y tratamiento de enfermedades como también el desarrollo de fármacos. En 2017, se inició la integración de algoritmos de inteligencia artificial clásica en la práctica médica, con modelos como máquinas de soporte vectorial y redes neuronales aplicados a la detección de patrones complejos en datos médicos (Miller & Brown, 2017).

Para 2022, la combinación de computación cuántica y machine learning permitió la creación de modelos como el Quantum Support Vector Machine (QSVM) y el Quantum Random Forest (QRF), mejorando la precisión y eficiencia en el análisis de grandes conjuntos de datos clínicos, incluidos aquellos relacionados con COVID-19 (Ullah et al., 2022).

En 2023, los modelos híbridos cuántico-clásicos, como Quantum Neural Networks (QNN), se utilizaron para personalizar tratamientos y optimizar terapias basadas en datos genéticos (Flöther, 2023). Más recientemente, en 2024, la integración de sensores cuánticos con inteligencia artificial avanzó en el campo de la imagenología médica, permitiendo diagnósticos más precisos mediante la detección de anomalías sutiles (Banerjee & Chatterjee, 2024). Estos avances reflejan el potencial transformador de la inteligencia artificial cuántica en la medicina, impulsando un futuro donde los tratamientos sean más personalizados y efectivos.

Marco Conceptual

Para el desarrollo de un modelo de inteligencia artificial para la detección de adenocarcinoma de colon, es indispensable el uso de diferentes plataformas, herramientas y estrategias y términos que nos sirvan de referencia y soporte para realizar de manera óptima cada paso hasta llegar hasta nuestro producto final, el cual es el software de clasificación de imágenes sanas y enfermas, a continuación se presenta una lista de los recursos utilizados para tal fin.

Inteligencia Artificial

Área de la informática que busca desarrollar sistemas capaces de realizar tareas que requieren inteligencia humana, como reconocimiento de voz, toma de decisiones y traducción automática.

Entrenamiento

Fase en la que un modelo de inteligencia artificial aprende a partir de un conjunto de datos para mejorar su desempeño en una tarea específica.

Validación

Proceso de evaluación de un modelo utilizando un conjunto de datos diferente al de entrenamiento para ajustar sus parámetros y evitar el sobreajuste.

Prueba

Fase final en la que se mide el desempeño de un modelo utilizando datos no vistos durante el entrenamiento o la validación.

Modelo

Conjunto de algoritmos y parámetros entrenados que permiten hacer predicciones o tomar decisiones basadas en datos de entrada.

Red

Estructura compuesta por nodos (o neuronas) interconectados que transmite información y procesa datos, comúnmente referida a redes neuronales en inteligencia artificial.

Recurrente

Tipo de red neuronal donde las conexiones entre las unidades forman un ciclo, lo que permite que la información se mantenga en "memoria" durante varias iteraciones.

Convolucional

Tipo de red neuronal especializada en procesamiento de datos que tienen una estructura en forma de cuadrícula, como imágenes. Utiliza filtros para detectar características locales.

Capa

Componente de una red neuronal que agrupa nodos y donde se realizan operaciones de procesamiento de datos en un modelo.

Visión por Computadora

Rama de la inteligencia artificial que permite a los sistemas automatizados interpretar y tomar decisiones basadas en imágenes y videos.

Clasificación de Imágenes

Técnica de visión por computadora que categoriza imágenes en diferentes clases o categorías a partir de sus características visuales.

Machine Learning

Subcampo de la inteligencia artificial que permite a los sistemas aprender de los datos sin ser programados explícitamente para realizar una tarea.

Deep Learning

Subcampo del machine learning que utiliza redes neuronales profundas con múltiples capas para aprender representaciones complejas de los datos.

Python

Lenguaje de programación versátil y ampliamente utilizado en inteligencia artificial, machine learning y deep learning debido a su simplicidad y gran cantidad de bibliotecas especializadas.

Input

Datos que se proporcionan a un modelo o sistema para que realice una operación o predicción.

Output

Resultado generado por un modelo o sistema después de procesar los datos de entrada.

Patch

Fragmento o porción pequeña de una imagen que se utiliza como entrada en tareas de visión por computadora o procesamiento de imágenes.

Librería

Conjunto de funciones y herramientas predefinidas que simplifican el desarrollo de software, especialmente en machine learning y deep learning (e.g., TensorFlow, PyTorch).

AISee

Entorno de desarrollo de inteligencia artificial que permite la visualización y análisis de modelos, comúnmente usado en tareas de deep learning.

Freezing

Técnica en la que se bloquean los parámetros de ciertas capas de un modelo preentrenado para que no se actualicen durante el proceso de entrenamiento adicional.

Dataset

Conjunto de datos utilizado para entrenar, validar y probar modelos de inteligencia artificial.

Google Colab

Entorno de desarrollo en la nube que permite ejecutar código Python de forma gratuita, con soporte para GPUs y TPUs, ideal para proyectos de machine learning y deep learning.

Azure

Plataforma en la nube de Microsoft que ofrece servicios para el desarrollo de aplicaciones, almacenamiento y ejecución de modelos de inteligencia artificial y machine learning.

TPU

(Tensor Processing Unit) Unidad de procesamiento desarrollada por Google específicamente para acelerar tareas de machine learning y deep learning.

CPU

(Central Processing Unit) Unidad de procesamiento central que ejecuta las operaciones principales de un sistema informático.

GPU

(Graphics Processing Unit) Unidad de procesamiento diseñada para realizar cálculos paralelos masivos, ampliamente utilizada en tareas gráficas y de deep learning.

Metodología

El desarrollo del proyecto se estructurará siguiendo una metodología ágil, basada en Scrum, que permita un desarrollo iterativo, adaptable y colaborativo. Esto facilitará el cumplimiento de los objetivos, asegurando una mejora continua y ajustes rápidos conforme surjan nuevos requerimientos o descubrimientos. La metodología se divide en las siguientes fases:

Fase de Investigación y Análisis de Requerimientos

Revisión de literatura científica sobre la detección de adenocarcinoma de colon mediante el análisis de imágenes médicas, incluyendo estudios previos sobre redes neuronales y machine learning aplicados en el ámbito de la medicina.

Análisis de requerimientos técnicos y funcionales, definiendo los objetivos del modelo de inteligencia artificial. Esto incluirá:

Identificación de las características específicas de las imágenes de colon que se utilizarán.

Estudio de los datasets disponibles, como bases de datos de imágenes médicas públicas o privadas.

Revisión de la normativa vigente para la manipulación de datos médicos, garantizando el cumplimiento de regulaciones como la HIPAA o GDPR.

Selección de Herramientas y Tecnologías

Evaluación de tecnologías y frameworks para visión por computadora y machine learning, como TensorFlow, Keras, PyTorch, OpenCV, entre otros.

Selección de hardware y software adecuados para el entrenamiento de redes neuronales. Esto incluirá:

Plataformas de computación de alto rendimiento (CPUs, GPUs).

Herramientas de visualización de datos y análisis, como Python, R o MATLAB.

Plataformas en la nube (Google Cloud, AWS) si se requiere capacidad adicional para procesamiento masivo de datos.

Validación de herramientas de almacenamiento de datos para el manejo de grandes volúmenes de imágenes, como bases de datos no relacionales o sistemas de almacenamiento en la nube.

Desarrollo del Modelo de Visión por Computadora

Recolección y preparación de datos:

Recolección de imágenes médicas de colon, asegurando su adecuada clasificación y anotación. Esto incluye la eliminación de ruidos y preprocesamiento de las imágenes (escalado, normalización).

Aplicación de técnicas de Data Augmentation para aumentar el número de muestras a partir de las imágenes disponibles, lo cual mejorará la capacidad de generalización del modelo.

Desarrollo del modelo de inteligencia artificial:

Diseño e implementación de una red neuronal profunda (CNN) específicamente configurada para la detección de adenocarcinoma en imágenes de colon.

Entrenamiento del modelo utilizando los datasets disponibles, aplicando técnicas de backpropagation y ajuste de hiperparámetros (tasa de aprendizaje, regularización, optimizadores).

Implementación de métricas de evaluación como la precisión, recall, F1-score y curvas ROC-AUC para medir el rendimiento del modelo en la identificación del adenocarcinoma.

Validación y Optimización del Modelo

Evaluación del modelo:

Aplicación de técnicas de validación cruzada y prueba del modelo en conjuntos de datos de testeo independientes.

Identificación de posibles problemas de sobreajuste o subajuste en el modelo, ajustando los parámetros o la arquitectura de la red.

Optimización del modelo:

Refinamiento del modelo utilizando algoritmos de optimización como Adam o RMSprop, y ajuste de la arquitectura de la red neuronal en función de los resultados obtenidos en las pruebas iniciales.

Implementación de técnicas de reducción de tiempo de procesamiento, como cuantización de modelos o pruning de capas no esenciales de la red.

Recolección de Datos

Para el desarrollo de este trabajo se utilizaron dos datasets: una parte del dataset "*Lung and Colon Cancer Histopathological Image Dataset (LC25000)*" , correspondiente a las muestras de colon sano y enfermo y otra parte de "*100,000 histological images of human colorectal cancer and healthy tissue (v0.1)*", correspondiente a colon sano y enfermo, se utilizo un formato tipo .jpeg para el primero y el segundo utiliza formato tipo .tif en sus imágenes.

Resultados

Primer Resultado

Para el desarrollo del modelo de clasificación se utilizaron principalmente tres tecnologías con gran demanda y trayectoria en la actualidad: Google Colab, y las librerías Pytorch y AISee, a continuación se muestran los parte conceptual y la importancia de cada uno en el proyecto desarrollado .

Google Colab

Google Colab, abreviatura de Google Colaboratory, es una plataforma de computación en la nube diseñada para facilitar la creación y ejecución de notebooks de Jupyter en un entorno accesible desde cualquier navegador web. Colab es especialmente popular en la comunidad de aprendizaje automático y ciencias de datos debido a su capacidad para ofrecer acceso gratuito a hardware de alto rendimiento, como GPUs y TPUs.

Esta herramienta permite a los usuarios ejecutar código Python, realizar análisis de datos, crear gráficos interactivos y documentar experimentos científicos de manera integrada. Además, Colab elimina la necesidad de instalar dependencias localmente, ya que permite importar bibliotecas y manejar archivos directamente en su entorno virtual. Su integración con Google Drive facilita el almacenamiento y la colaboración en tiempo real, haciendo que sea una solución eficiente para proyectos grupales o individuales (Bisong, 2019).

Google Colab se ha convertido en un recurso invaluable para estudiantes, investigadores y profesionales debido a su enfoque en la accesibilidad y la reducción de barreras para el acceso a tecnologías avanzadas.

PyTorch

PyTorch es una biblioteca de aprendizaje profundo de código abierto ampliamente utilizada en investigación y desarrollo de inteligencia artificial. Diseñada inicialmente por Facebook AI Research (FAIR), PyTorch ha ganado una amplia adopción gracias a su enfoque dinámico para la construcción y el entrenamiento de redes neuronales, lo que permite realizar ajustes en tiempo de ejecución. Esto se logra a través de su motor de gráficos computacionales dinámicos, que contrasta con los gráficos estáticos de otras bibliotecas como TensorFlow.

PyTorch proporciona una API intuitiva y fácil de usar, ideal tanto para principiantes como para expertos, permitiendo realizar operaciones tensoriales, definir modelos personalizados y entrenar redes neuronales complejas. Además, ofrece soporte nativo para operaciones en GPU, lo que mejora significativamente el rendimiento en tareas computacionalmente intensivas. Gracias a su ecosistema robusto y en constante evolución, PyTorch se ha convertido en una herramienta esencial para el desarrollo de aplicaciones de visión por computador, procesamiento del lenguaje natural y aprendizaje por refuerzo (Paszke et al., 2019).

AISee

AISee es una librería de código abierto para visión por computadora, desarrollada sobre las librerías PyTorch y Timm. Está diseñada para proporcionar una interfaz fácil de usar para el entrenamiento y la predicción con redes neuronales de última generación.

AISee está basada en:

PyTorch. Una librería de aprendizaje profundo que proporciona flexibilidad y velocidad en el desarrollo de modelos.

Timm. Una colección de modelos de visión por computadora preentrenados y herramientas para su uso.

Cómo funciona?

AISee simplifica el proceso de entrenamiento y predicción mediante las siguientes características:

Interfaz simple. Permite entrenar y predecir usando modelos de la librería Timm con pocas líneas de código.

Carga de imágenes. Facilita la carga de imágenes desde una carpeta, un DataFrame de pandas o una ruta de imagen única.

Entrenamiento de redes neuronales. Permite entrenar redes neuronales desde pesos preentrenados o desde cero.

Soporte para clasificación. Admite tareas de clasificación de imágenes multicategoría y multietiqueta.

Gestión de DataLoaders y transformaciones. Se encarga de los DataLoaders, transformaciones de imágenes y bucles de entrenamiento e inferencia.

Al tener una gran personalización de los parámetros para desarrollar el modelo, AISee nos brinda múltiples combinaciones de clases, learning method, tipo de etiqueado (multiclase o multietiqueta), # de épocas, etc , lo cual hace mucho más flexible nuestras creaciones y mejores rendimientos para cada caso particular que se requiera su uso.

Estas tres tecnologías son fundamentales para la realización del proyecto, proporcionándonos gran practicidad, flexibilidad, disponibilidad y un enorme potencial para el desarrollo de diversos tipos de proyectos.

Google Colab proporciona un ambiente más amigable y fácil de utilizar a diferencia de otras plataformas para iguales fines como lo son Azure, mediante ella podemos crear, modificar, personalizar y correr los modelos seleccionados utilizando librerías bastante conocidas como

Pytorch y Scikit-learn, apoyados además en AISee, una novedosa librería que nos permite hacer configuraciones de los modelos de manera rápida y más sencilla, generando así un modelo de clasificación óptimo para nuestra tarea de clasificación de imágenes médicas, en este caso particular, pero de igual manera, al integrar las tecnologías antes mencionadas, hay múltiples e innmumerables posibilidades para desarrollar.

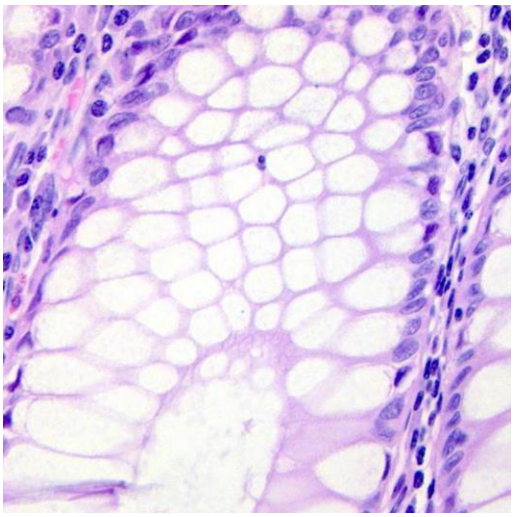
Segundo Resultado

Para el desarrollo de este proyecto se utilizaron algunas de las arquitecturas más populares en el mercado (ResNet-34, ResNet-50,DenseNet-121,EfficientNet B0, RegNet y_002, VGG-16, Inception(GoogleNet), los cuales, después de ser personalizados, utilizados y analizados, sirvieron de base para el modelo ensamblado final, a continuación se presenta el tipo de imágenes utilizadas, el concepto de cada arquitectura seleccionada para este proyecto, después se muestran los pasos de los procesos de parámetrización, entrenamiento y prueba de un modelo de clasificación de imágenes histopatológicas sanas y enfermas de colon, luego de lo cual de los pasos de dichos procesos, están presentadas de forma esquemática, con todos sus componentes, las arquitecturas mencionadas, posteriormente se podrá visualizar la configuración de los parámetros y por último, la información de entrenamiento de cada modelo.

Para el entrenamiento de los modelos estudiados se utilizó un dataset de imágenes histopatológicas de colon, 50% de ellas con adenocarcinoma y 50% sanas, a continuación se muestra un ejemplo de cada tipo de colon.

Figura 1

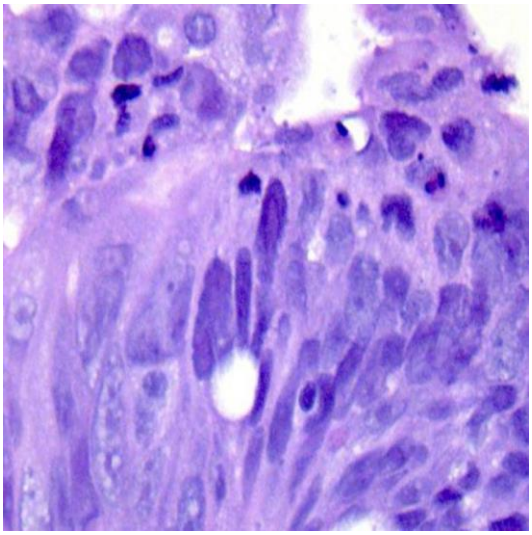
Imagen Histopatológica de Colon



Nota. la imagen muestra un colon sano (Borkowski AA et al,. 2019).

Figura 2

Imagen Histopatológica de Colon Enfermo



Nota. la imagen muestra un colon con adenocarcinoma (Borkowski AA et al,. 2019).

Siguiendo la metodología establecida, a continuación, se presentará una breve introducción a cada una de las arquitecturas de redes neuronales convolucionales (CNN) utilizadas en este estudio. Estas arquitecturas han sido seleccionadas por su relevancia y desempeño en tareas de visión por computadora, permitiendo extraer características de las imágenes de manera eficiente. En los siguientes apartados, se describirán sus principales características, ventajas y aplicaciones, proporcionando así una visión general de su funcionamiento y utilidad en el contexto de esta investigación.

ResNet-34 (Residual Network con 34 capas)

Concepto. ResNet-34 es una red neuronal profunda diseñada para abordar el problema de la degradación del gradiente en arquitecturas profundas. Fue introducida en el artículo "Deep Residual Learning for Image Recognition" (He et al., 2015).

Características Clave. Arquitectura residual con conexiones de salto para mejorar la propagación del gradiente, compuesta por 34 capas, con convoluciones de tamaño 3×3 y max pooling y utiliza bloques residuales simples con dos capas convolucionales, seguidas de normalización por lotes y activación ReLU.

Aplicaciones. Clasificación de imágenes en conjuntos de datos como ImageNet y CIFAR-10, detección y segmentación de objetos cuando se usa como base en modelos como Faster R-CNN.

ResNet-50 (Residual Network con 50 capas)

Concepto. ResNet-50 es una versión más profunda de ResNet-34 con 50 capas, optimizada para mejorar la precisión sin aumentar significativamente la complejidad computacional.

Diferencias con ResNet-34. Utiliza bloques residuales tipo "bottleneck" con tres capas convolucionales (1×1, 3×3, 1×1) en lugar de dos, reduce la dimensionalidad con convoluciones 1×1 antes y después de la convolución 3×3, tiene mayor cantidad de parámetros, lo que le permite capturar características más complejas.

Aplicaciones. Clasificación de imágenes de alta resolución, detección y segmentación de objetos en modelos avanzados como Mask R-CNN.

DenseNet-121 (Densely Connected Convolutional Networks)

Concepto. DenseNet-121 introduce una arquitectura en la que cada capa está conectada con todas las capas posteriores, maximizando la reutilización de características y reduciendo el número de parámetros.

Características Clave. Cada capa recibe la salida de todas las capas anteriores y la transmite a todas las siguientes, utiliza menos parámetros que ResNet, ya que evita la redundancia de información y controla la cantidad de nueva información añadida en cada capa mediante el parámetro growth rate.

Aplicaciones. Clasificación y segmentación de imágenes en tiempo real, diagnóstico médico mediante análisis de imágenes de resonancia magnética y tomografías.

EfficientNet-B0

Concepto. EfficientNet-B0 es parte de la familia EfficientNet, diseñada para lograr un balance óptimo entre eficiencia computacional y precisión mediante la técnica de escalamiento compuesto.

Características Clave. Utiliza compound scaling, una estrategia que ajusta simultáneamente la profundidad, el ancho y la resolución de la red de manera equilibrada,

emplea la activación Swish en lugar de ReLU para mejorar la propagación del gradiente y utiliza convoluciones separables en profundidad para reducir los cálculos sin perder precisión.

Aplicaciones. Clasificación de imágenes en dispositivos con restricciones computacionales e Implementaciones en sistemas embebidos y dispositivos móviles.

RegNetY_002 (Regularized Network - Y series)

Concepto. RegNet es una familia de redes neuronales diseñadas para encontrar automáticamente la mejor estructura de red para una tarea dada, maximizando eficiencia y rendimiento.

Características clave. Optimiza el diseño de la red ajustando automáticamente la distribución de canales y profundidad, emplea bloques Squeeze-and-Excitation (SE) para mejorar la capacidad del modelo de enfocarse en características relevantes, utiliza menos parámetros y cálculos en comparación con redes profundas tradicionales.

Aplicaciones. Clasificación y segmentación de imágenes en tiempo real y optimización de modelos en hardware de última generación.

VGG-16 (Visual Geometry Group - 16 capas)

Concepto. VGG-16 es una arquitectura de red neuronal profunda caracterizada por el uso de múltiples capas convolucionales de pequeño tamaño y max pooling.

Características Clave. Posee 16 capas con convoluciones de tamaño 3×3 y max pooling, su diseño prioriza la simplicidad estructural, lo que facilita su implementación., presenta un elevado número de parámetros, aproximadamente 138 millones.

Aplicaciones. Clasificación de imágenes en conjuntos de datos como ImageNet y transfer learning en diversas tareas de visión por computador.

GoogleNet (Inception v1)

Concepto.GoogleNet, también conocida como Inception v1, introduce el uso de módulos Inception, que permiten aplicar convoluciones de diferentes tamaños en paralelo dentro de una misma capa.

Características Clave. Emplea módulos Inception, que combinan convoluciones de tamaños 1×1, 3×3 y 5×5 en paralelo para capturar características a diferentes escalas, reduce el número de parámetros mediante convoluciones 1×1, utilizadas para disminuir la dimensionalidad antes de convoluciones más grandes y es más eficiente en comparación con redes profundas tradicionales como VGG.

Aplicaciones. Clasificación y segmentación de imágenes, detección de objetos en tiempo real.

Pasos de los Procesos de Parámetrización, Entrenamiento y Prueba de un Modelo de Clasificación de Imágenes

El proceso de construcción de un modelo de clasificación de imágenes involucra varias etapas clave que garantizan su correcto funcionamiento y desempeño. En primer lugar, la parámetrización define la configuración del modelo, estableciendo aspectos como la arquitectura de la red, las funciones de activación y los hiperparámetros. Luego, en la fase de entrenamiento, el modelo aprende a reconocer patrones en las imágenes a partir de un conjunto de datos etiquetado, ajustando sus pesos mediante algoritmos de optimización. Finalmente, en la etapa de prueba, se evalúa el rendimiento del modelo con datos no vistos, permitiendo medir su capacidad de generalización y ajuste. Estos pasos son fundamentales para garantizar un modelo preciso y eficiente en tareas de clasificación de imágenes. A continuación, se encuentra una breve descripción de cada paso.

Paso 1: Importar Librerías Necesarias. Importa e instala las librerías necesarias para realizar todos los pasos del código y los procesos correspondientes al modelo que se selecciona para utilizar, entre las que se destaca la novedosa AISee, enfocada en clasificación de objetos, nos permite una parametrización fácil y más rápida.

Paso 2: Configuración de dispositivos. Se le indica al sistema que use “cuda” como dispositivo para correr el código, y si no está disponible, que utilice “cpu”, de la disponibilidad de “cuda” dependerá la rapidez y facilidad con la cual se desarrollará este código en Google Colab.

Paso 3: Descargar el dataset. Descarga el dataset que vamos a usar, directamente de Kaggle, una forma más práctica de usar el dataset del proyecto, también se puede descargar en Google drive el dataset y utilizarlo pero es más fácil, rápida y eficiente la primera opción mencionada.

Paso 4: Extraer y cargar el dataset. Extrae los datos del dataset en formato zip que proviene de Kaggle: de esta manera se optimiza espacio disponible y luego se le asigna un nombre a la carpeta producto de la extracción.

Paso 5: Dividir datos del dataset. Divide los datos del dataset, según el porcentaje asignado al conjunto de entrenamiento, validación y prueba, en el caso de la figura, 70%, 15% y 15%, respectivamente. Hay que tener mucho cuidado con la estructura de las rutas de directorio en “input_folder” y “output_folder”, ya que un un mínimo detalle puede generar un error al generar la carpeta con los datos de entrenamiento, validación y prueba.

Paso 6: Crear el dataset de entrenamiento. Se crea el dataset de entrenamiento del proyecto, este será muy importante para el rendimiento del modelo a la hora de la clasificación final en la etapa de evaluación o testing.

Paso 7: Parametrización del modelo e instaciación de la clase trainer. Configura los detalles del modelo (número de clases, índices de las clases, método de aprendizaje, dispositivo, etc). Por otro lado, instancia la clase trainer, la cual entrenará el modelo para la posterior utilización en clasificación de objetos, es importante configurar todos los parámetros de la manera correcta para obtener un buen desempeño al utilizar nuestro modelo.

Paso 8: Entrenamiento del modelo. Entrena el modelo, en este caso, se usaron 10 épocas para su entrenamiento para obtener resultados óptimos. En cada Época se va incrementando el rendimiento del modelo que se está entrenando, logrando su máximo nivel en ocasiones antes de la última época, por esto es importante analizar el entrenamiento por cada época para optimizar la configuración del modelo y, de ser necesario, quitar, agregar , disminuir o aumentar ciertos componentes o parámetros, para mejorar los niveles de desempeño del modelo.

Paso 9: Evaluación del modelo. Evalua mediante dos parámetros de rendimiento (F1 Score y Accuracy), el nivel de fiabilidad del modelo. Estos son puntos de referencia para saber qué tan bueno es nuestro modelo y sus resultados.

Paso 10: Prueba del modelo. Clasificación de una imagen de colon: el primer número decimal muestra la probabilidad que sea maligna y en el segundo que sea benigna, obteniendo en este caso, una clasificación correcta (célula maligna), con un 99,30% de probabilidad, señalada en el primer número decimal.

Para acceder al código correspondiente a los pasos descritos anteriormente, puede consultar el apéndice J de este documento.

En este repositorio, encontrará la implementación detallada de la parámetrización, el entrenamiento y la prueba del modelo de clasificación de imágenes. Esto le permitirá comprender mejor cada etapa del proceso y replicar los experimentos realizados en este estudio.

Si desea explorar en mayor profundidad las arquitecturas de redes neuronales convolucionales utilizadas, puede consultar el apéndice K de este documento.

A continuación, se presenta la configuración de parámetros utilizada para cada uno de los modelos empleados en este estudio. Es importante destacar que se mantuvo la misma configuración en todos los casos para garantizar una comparación equitativa. A modo de ejemplo, se muestra la configuración aplicada al modelo **ResNet-34**:

Configuración del Modelo

Parámetros del modelo:

Nombre del modelo: resnet34

Número de clases: 2

Mapeo de clases: "colon_aca": 0, "colon_n": 1

Método de aprendizaje: "freezed"

Tipo de tarea: "single_label"

Dispositivo de procesamiento: "cuda"

Instanciación de la clase Trainer

Para el entrenamiento del modelo, se utilizó la siguiente configuración al instanciar la clase Trainer**:**

Parámetros de entrenamiento:

Fuente de datos: "data"

Directorio de salida: "/content/data/test_trainer.pt"

Tamaño del lote: 8

Número de épocas: 10

Métrica para guardar puntos de control: "loss"

Aleatorización del conjunto de datos**:** True

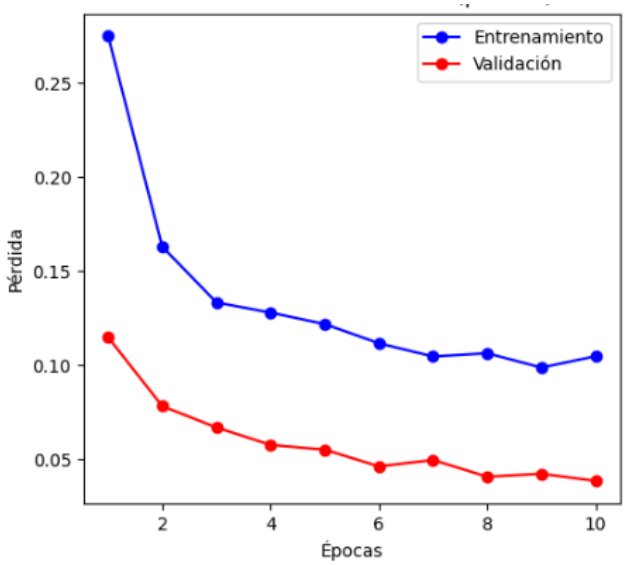
Cabe resaltar que en cada epoch o época se muestran los parámetros de rendimiento de F1 y accuracy, como también el tiempo de entrenamiento total y la epoch con mejor valor de perdida, todo esto es posible gracias a la librería AISee, con la cual podemos entrenar validar y evaluar los modelos disponibles, de manera más práctica que otras librerías de uso común para inteligencia artificial

Se probó el uso de menos epochs o épocas en el entrenamiento, luego de lo cual se llegó a la conclusión que 10 era el número ideal para obtener buenos resultados en la clasificación de imágenes, en cada epoch fue disminuyendo la pérdida, lo cual indica incremento de la precisión del modelo con cada nueva época, como es la meta, si quiere observar el proceso de entrenamiento de cada modelo, puede consultar el apéndice L de este documento.

En la siguiente imagen se muestra el desempeño en cuando a pérdida o loss, en este caso particular, para el modelo ResNet-34, donde se compara la pérdida a través del entrenamiento versus la validación, es un ejemplo que refleja como mejora el rendimiento del modelo a medida que pasa cada época:

Figura 3

Imagen Comparativa Entrenamiento vs Validación



Nota. Imagen comparativa entrenamiento vs validación del modelo ResNet-34 en términos de pérdida. Obtenido. Autoría Propia.

Tercer Resultado

En esta etapa, se presenta la aplicación detallada de diversos modelos de redes neuronales convolucionales, los cuales fueron utilizados para construir un modelo ensamblado final. Además, se incluye una Tabla 2 comparativa que evalúa el rendimiento de cada modelo mediante distintas métricas. Se llevaron a cabo múltiples ejecuciones con los siguientes modelos previamente configurados**:** ResNet-50, DenseNet-121, EfficientNet-121, RegNet Y_002, ResNet-34, Inception (GoogleNet) e Inception v3. Estos modelos fueron empleados para la clasificación de imágenes histopatológicas, diferenciando entre muestras benignas y malignas. Es importante destacar que todos los modelos se entrenaron bajo la misma configuración de parámetros, previamente descrita en el segundo resultado, con el fin de garantizar una

comparación equitativa. La siguiente tabla presenta los tiempos de entrenamiento, la precisión (Accuracy) y la puntuación F1-score obtenida por cada modelo.

Tabla 2

Tiempos de Entrenamiento de los Modelos Seleccionados

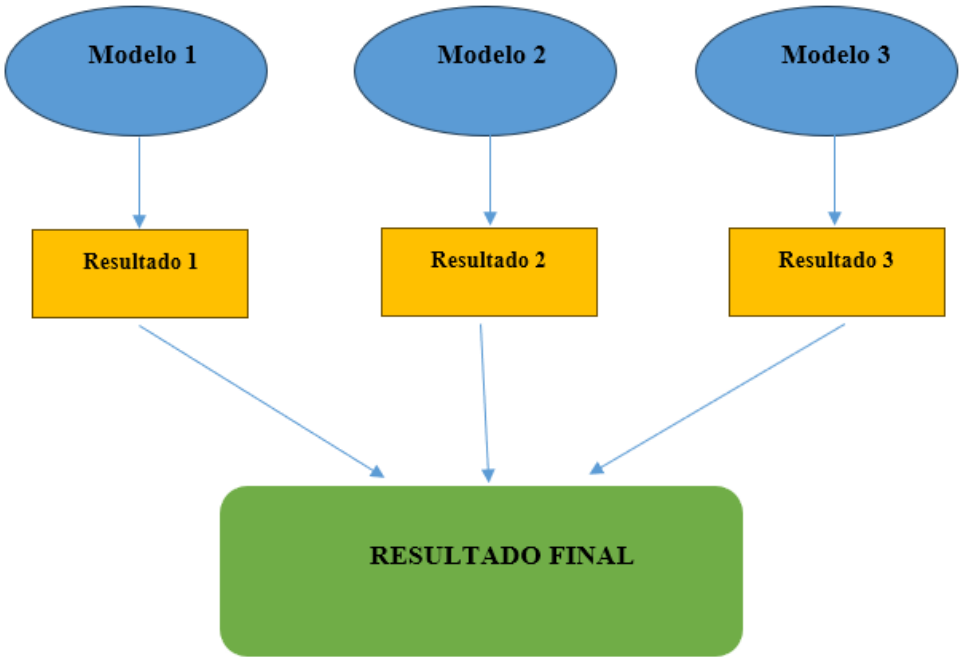
Modelo	Tiempo de entrenamiento	F1	Accuracy
ResNet-50	13 min y 13 s	0.9919	0.992
DenseNet-121	15 min y 44 s	1.0000	1.000
EfficientNet B0	12 min y 60 s	0.9973	0.997
RegNet y_002	13 min y 19 s	0.9986	0.9986
ResNet-34	14 min y 1 s	0.9893	0.9893
Inception v3	15 min y 56 s	0.9966	0.9966
VGG16	15 min y 44 s	0.9979	0.998

Nota. tabla de los tiempos de entrenamiento de los modelos de inteligencia artificial. Obtenido. Autoría Propia.

Realizado esto, de acuerdo a los mejores tiempos, se seleccionaron los modelos ResNet-50, EfficientNet B0 y RegNet y_002, para la realización del modelo ensamblado, en la modalidad de voteo o *voting*, la cual consiste en realizar la clasificación de la misma imagen por cada uno de los modelos seleccionados, escogiendo como resultado el que seleccione la mayoría de los modelos implicados (para este caso particular, sería el que nos arrojaran dos de los tres modelos), para mayor claridad se muestra la dinámica del sistema en un esquema a continuación:

Figura 4

Esquema Ejemplo de Modelo Ensamblado



Nota. Esquema Modelo ensamblado, usando la variante de voteo o *voting.* Obtenido. Autoria Propia.

Ahora se dará una ampliación de la conceptualización de los tres modelos seleccionados para el modelo ensamblado para conocer en mayor detalle acerca de sus características principales:

ResNet-50

La arquitectura ResNet-50 es una red neuronal convolucional profunda que introduce conexiones residuales para mitigar el problema de la desaparición del gradiente en redes muy profundas. Compuesta por 50 capas, ResNet-50 emplea bloques residuales que permiten que la información fluya sin interrupciones a través de las capas mediante atajos o "skip connections",

mejorando la eficiencia y precisión del entrenamiento en tareas de clasificación de imágenes (Sarwinda et al., 2021).

EfficientNet B0

Es una arquitectura de red neuronal optimizada mediante "compound scaling", donde la profundidad, el ancho y la resolución de las imágenes se balancean sistemáticamente para maximizar la eficiencia del modelo. Utiliza bloques invertidos de cuellos de botella y módulos squeeze-and-excitation para mejorar la capacidad de representación con un bajo costo computacional. Este enfoque permite que EfficientNet B0 sobresalga en precisión comparada con arquitecturas más pesadas (Hoang & Jo, 2021).

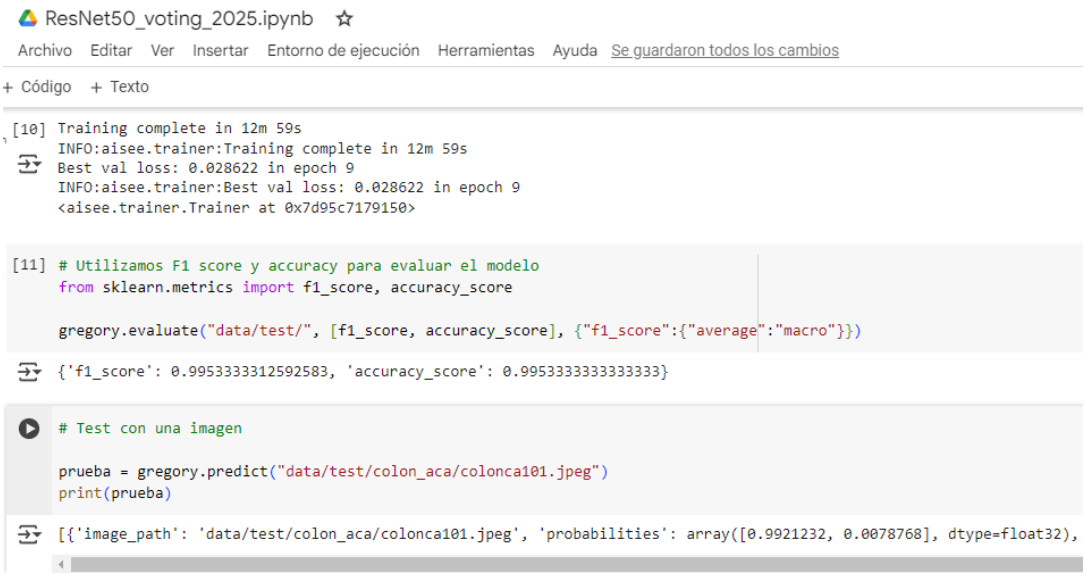
RegNet Y_002

Pertenece a la familia RegNet, que se diferencia por su diseño paramétrico simple pero eficaz para la búsqueda de arquitectura de redes neuronales. Se construye mediante el ajuste de parámetros estructurales como la cantidad de canales y la longitud de los bloques para alcanzar un balance óptimo entre eficiencia computacional y precisión en clasificación de imágenes. Su arquitectura incorpora mecanismos de atención dividida que mejoran la representación multi-escala (Zhang et al., 2020).

Hecho esto, se mostrará el proceso de voting o voteo, con los modelos seleccionados, utilizando para esto la misma imagen histopatológica como referencia para realizar el proceso de clasificación, con cada modelo y luego, teniendo en cuenta el resultado o voto de cada uno, dar el resultado final sobre la imagen de referencia, para esto usaremos la imagen “colonca101.jpeg”. Se realiza el proceso de clasificación de la imagen con ResNet-50, obteniendo el siguiente resultado mostrado en las siguientes imágenes.

Figura 5

Resultado de Clasificación Usando el Modelo ResNet-50



Nota. Clasificación de una imagen de colon. Obtenido. Autoria Propia.

Podemos observar que usando el modelo ResNet-50 tenemos con un 99,21% de probabilidad que la imagen pertenece a una colon enfermo (adenocarcinoma de colon).

Figura 6

Resultado de Clasificación Usando el Modelo EfficientNet B0

Nota. Clasificación de una imagen de colon. Obtenido. Autoria Propia.

Podemos observar que usando el modelo EfficientNet B0, tenemos como resultado con un 99,75% de probabilidad, que la imagen pertenece a una colon enfermo (adenocarcinoma de colon).

Figura 7

Resultado de Clasificación Usando el Modelo RegNet y_002

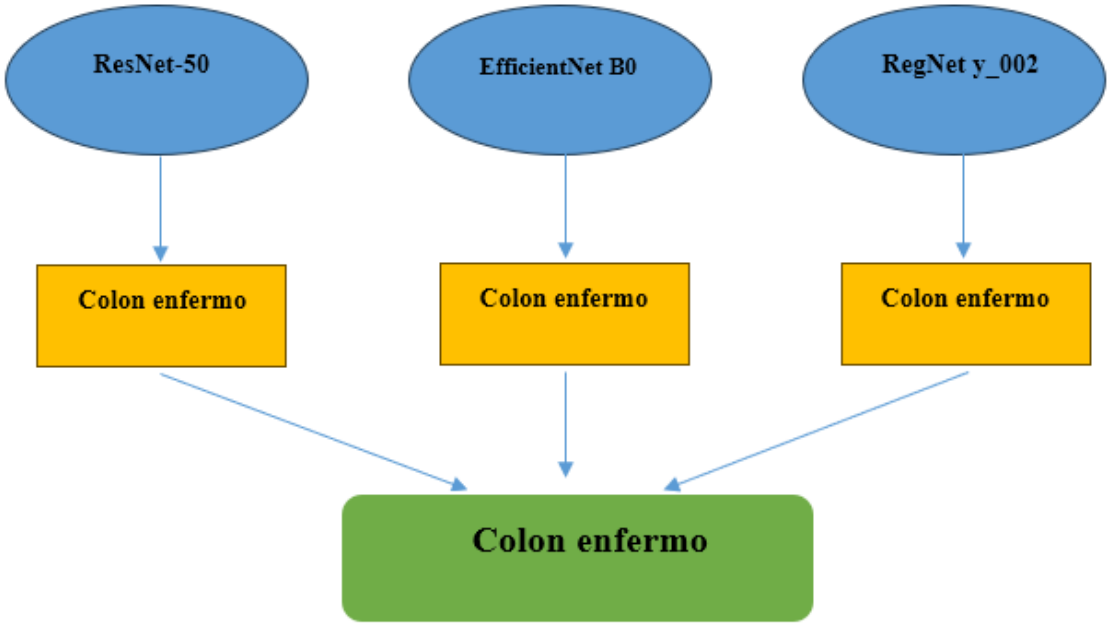
Nota. Clasificación de una imagen de colon. Obtenido. Autoria Propia.

De este modo, se observa que usando el modelo RegNet y_002, tenemos como resultado con un 99,21% de probabilidad, que la imagen pertenece a una colon enfermo (adenocarcinoma de colon).

Finalmente, se realiza la recopilación de los tres resultados o votos, obteniendo como resultado contundente, ya que los tres modelos nos arrojaron como voto colon enfermo, se refleja este proceso en el siguiente esquema:

Figura 8

Modelo Ensamblado Final



Nota. Esquema de Modelo ensamblado para clasificar adenocarcinoma de colon, usando la variante de voteo o *voting*, en este caso particular, con resultado positivo. Obtenido. Autoria Propia.

Después de entrenar el sistema de Modelo ensamblado, al realizar la validación, se obtuvieron los siguientes resultados:

Métricas del modelo ensamblado:

Precisión: 1.0000

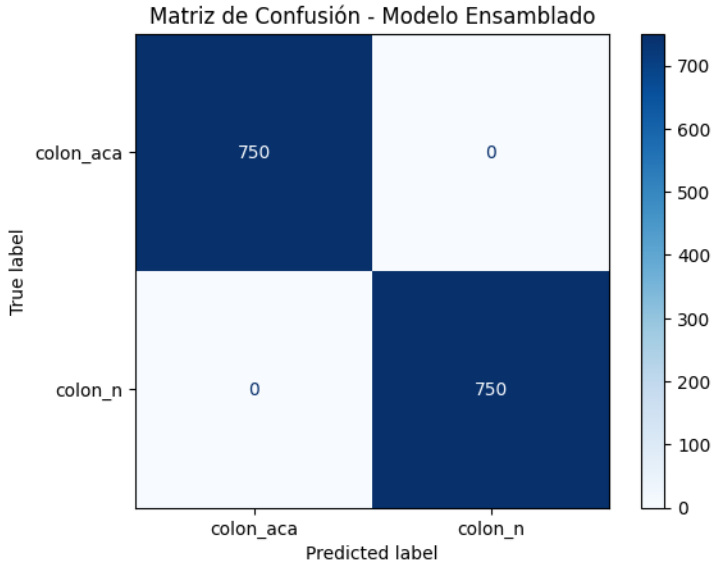
Precisión Macro: 1.0000

Recall Macro: 1.0000

F1-Score Macro: 1.0000

Figura 9

Matriz de Confusión del Modelo Ensamblado



Nota. Matriz de confusión del Modelo ensamblado luego de realizar la validación. Obtenido. Autoria Propia.

Si quiere ver el cuaderno con el código a detalle, puede consultar el apéndice M de este documento.

Luego, al usar un dataset con imágenes diferentes del dataset inicial, para testing o prueba, se obtuvieron la siguientes métricas:

Métricas del modelo ensamblado (testing):

Precisión: 0.8248

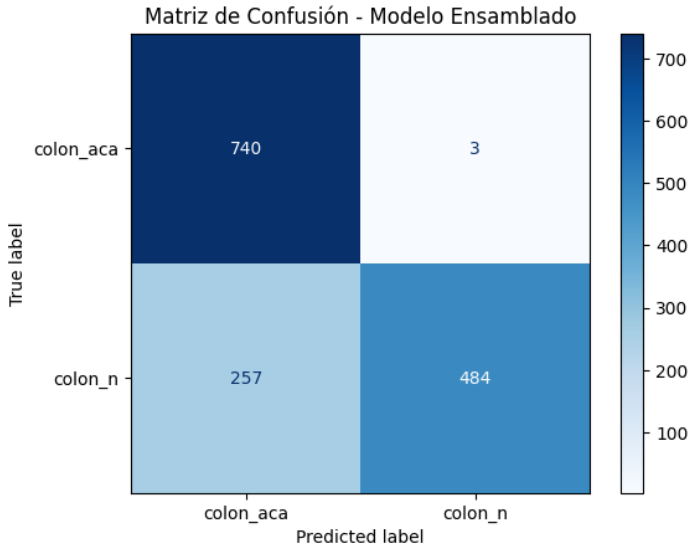
Precisión Macro: 0.8680

Recall Macro: 0.8246

F1-Score Macro: 0.8194

Figura 10

Matriz de Confusión del Modelo Ensamblado (Testing)



Nota. Matriz de confusión del Modelo ensamblado luego de realizar el testing o prueba. Obtenido. Autoria Propia.

Si quiere ver el cuadernillo con el código a detalle, puede consultar el apéndice N de este documento.

Lo anterior refleja un desempeño de 82,48% de precisión en las clasificaciones de las imágenes, que aunque no es excelente, si un buen índice de partida, teniendo en cuenta los estándares de este tipo de software, el cual puede ser optimizado a futuro si se requiere para usos reales en el campo médico.

Análisis complementario aplicado a otros tipos de cáncer

Después de realizar el análisis y clasificación de imágenes de cáncer de colon con el dataset nuevo, se decidió realizar, teniendo en cuenta el modelo ensamblado base (con el mismo

entrenamiento basado en el dataset inicial de imágenes de células sanas y con cáncer de colon), dos nuevas pruebas a un dataset de imágenes de células sanas y con cáncer de pulmón y otro dataset con células sanas y con cáncer de seno para contrastar sus resultados con los obtenidos inicialmente, al hacerlo se obtuvieron los siguientes resultados:

Prueba con dataset de imágenes sanas y enfermas de pulmón

Métricas del modelo ensamblado (testing):

Precisión: 0.5000

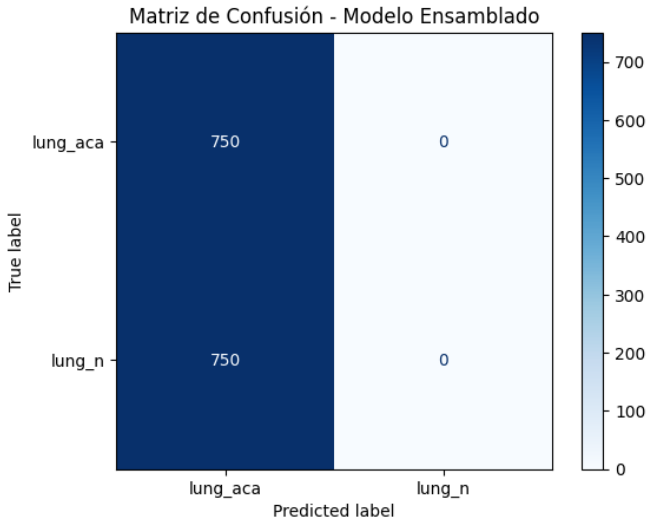
Precisión Macro: 0.2500

Recall Macro: 0.5000

F1-Score Macro: 0.3333

Figura 11

Matriz de Confusión del Modelo Ensamblado - Cáncer de Pulmón



Nota. Matriz de confusión del Modelo ensamblado luego de realizar el testing con dataset de imágenes sanas y enfermas de pulmón . Obtenido. Autoria Propia.

Si quiere ver el cuadernillo con el código a detalle, consultar el apéndice O.

Luego de estos resultados, podemos observar que el modelo ensamblado tiene un fuerte sesgo, tendiendo a clasificar todas las imágenes como enfermas, obteniendo el 50% de precisión sus resultados(como también se puede ver en las métricas anteriores a la matriz de confusión).

Prueba con dataset de imágenes sanas y enfermas de seno

Métricas del modelo ensamblado:

Precisión: 0.5000

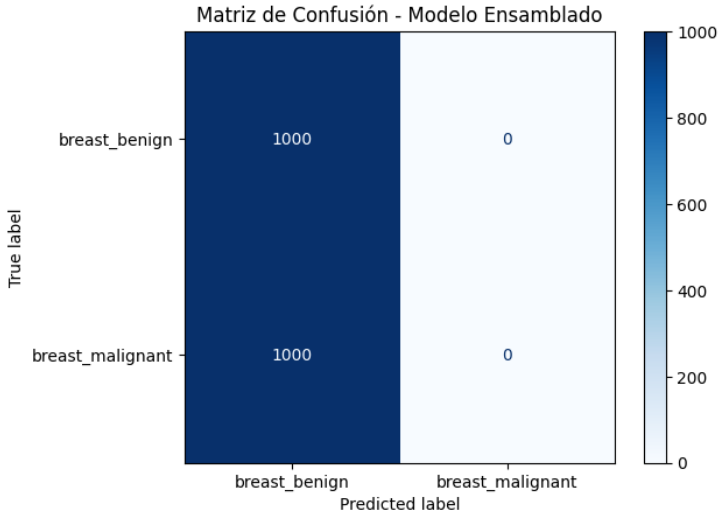
Precisión Macro: 0.2500

Recall Macro: 0.5000

F1-Score Macro: 0.3333

Figura 12

Matriz de Confusión del Modelo Ensamblado - Cáncer de Seno



Nota. Matriz de confusión del Modelo ensamblado luego de realizar el testing con dataset de imágenes sanas y enfermas de seno. Obtenido. Autoria Propia.

Sorprendentemente, el modelo ensamblado obtiene los mismos valores en las métricas de rendimiento en el caso de este nuevo dataset, con sesgo igualmente marcado, pero en este casi hacia la clase “célula sana” , obteniendo una precisión del 50% en sus resultados, este interesante cambio de sesgo en las clasificaciones puede deberse a que las distribuciones de color, textura y morfología cambian entre órganos, los filtros que antes separaban “cáncer” de “sano”, dejan de ser informativos, y posiblemente, si en pulmón las imágenes cancerosas se parecen más a las de cáncer de colon, la red se inclina a predecir *cáncer*; si en seno las imágenes sanas se parecen más a las células sanas de colon, la red se inclina a predecir *sano,* en pocas palabras, al no haber sido entrenado en la clasificación de los dos nuevos datasets de pulmón y seno, el modelo ensamblado puede inclinarse fuertemente al tipo de imágenes de células con las que tenga la mayor semejanza, ya sea sanas o enfermas.

Conclusiones

Este proyecto fue creado debido al gran impacto del cáncer de colon a nivel mundial, teniendo altas tasas de mortalidad anualmente, las cuales podrían disminuir en gran medida si se incorporara en estos procesos herramientas tecnológicas como lo es la visión artificial, mediante la clasificación de imágenes médicas, este proyecto responde a esa necesidad en el sector médico y contribuye por consiguiente a mejorar la calidad de vida de las personas.

Para el desarrollo de este se probaron múltiples opciones disponibles de inteligencia artificial en el sector tecnológico para observar su comportamiento en momentos claves como el entrenamiento hasta el resultado final en la clasificación de las imágenes médicas, al obtener resultados muy similares en los parámetros de rendimiento (accuracy y F1 score) se decidió tomar como referencia principal de rendimiento el tiempo de entrenamiento, escogiendo los tres mejores modelos según esta referencia (ResNet-50, EfficientNet b0 y RegNet y_002) para usarlos para crear un sistema de modelo emsamblado, en la variante de *voting* o voteo, maximizando así la precisión de nuestros resultados.

Por último, para obtener óptimos resultados es importante usar un dataset con una cantidad importante de imágenes, que sea obtenida de fuentes confiables, esto ayudará a aumentar la precisión de nuestro modelo en el momento del entrenamiento y a que sea más confiable al ser utilizado.

Recomendaciones

A continuación se ofrecen una serie de recomendaciones para tener en cuenta, luego de terminado el proyecto en mención:

Recomendación 1: Se debe identificar claramente el problema a solucionar, esto nos servirá para crear un modelo que resuelva un problema puntual o satisfaga una necesidad del usuario.

Recomendación 2: Es necesario asegurar la calidad del dataset a utilizar, por lo cual debe provenir de una fuente confiable y con una trayectoria importante en el área.

Recomendación 3: Es importante asegurar una gran diversidad de característica de las imágenes en el dataset de entrenamiento de nuestro modelo, ya que esto será crucial en el nivel de precisión que tendrá a la hora de probarlo y posteriormente usarlo.

Recomendación 4: Al crear un dataset para entrenamiento del modelo, debemos asegurarnos de que haya un equilibrio entre las clases a predecir, esto será importante para prevenir posibles sesgos en las posteriores clasificaciones del modelo.

Recomendación 5: Es importante realizar varias pruebas al modelo para así no caer en resultados erróneos en el momento del despliegue final.

Recomendación 6: Definido el problema a resolver, debemos tomarnos el tiempo para seleccionar las herramientas adecuadas para resolver el problema, esto será determinante para hacer más o menos rápido y sencillo la resolución de dicho problema.

Referencias Bibliográficas

Ba Alawi, A. E., & Bozkurt, F. (2023). CNN-based colon cancer recognition model. *2023 3rd International Conference on Emerging Smart Technologies and Applications (eSmarTA),* 1–5.

Banerjee, N., & Chatterjee, K. (2024). Quantum AI in healthcare: Revolutionizing diagnosis, treatment and drug discovery. *International Journal of Scientific Research in Science and Technology.*

Bisong, E. (2019). *Building Machine Learning and Deep Learning Models on Google Cloud Platform: A Comprehensive Guide for Beginners.* Apress.

Breiman, L. (2001). Random forests. *Machine Learning, 45*(1), 5–32. https://doi.org/10.1023/A:1010933404324

Dietterich, T. G. (2000). Ensemble methods in machine learning. *In International Workshop on Multiple Classifier Systems* (pp. 1–15). Springer.

El-Aziz, A. A., Mahmood, M. A., & El-Ghany, S. A. (2024). Advanced deep learning fusion model for early multi-classification of colon cancer using histopathological images. *Diagnostics.*

Flöther, F. F. (2023). The state of quantum computing applications in health and medicine. *Research Directions: Quantum Technologies.*

Forsyth, D. (2018). *Computer vision: A modern approach.* Prentice Hall.

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *Annals of Statistics, 29*(5), 1189–1232. https://doi.org/10.1214/aos/1013203451

Hoang, V.-T., & Jo, K. (2021). Practical Analysis on Architecture of EfficientNet. *2021 14th International Conference on Human System Interaction (HSI),* 1-4.

Jaware, T. H., Kasturiwale, H., Thakur, R., Jakhete, M. D., Chavan, M., & Rane, M. (2024). Deep learning model for colon cancer classification using InceptionV3. *Journal of Electrical Systems.*

Jheng, Y. C., Wang, Y. P., Lin, H. E., Sung, K. Y., Chu, Y. C., Wang, H. S., Jiang, J. K., Hou, M., Lee, F., & Lu, C. L. (2021). A novel machine learning-based algorithm to identify and classify lesions and anatomical landmarks in colonoscopy images. *Surgical Endoscopy, 36*, 640–650.

Kather, J. N., Halama, N., & Marx, A. (2018). 100,000 histological images of human colorectal cancer and healthy tissue (v0.1) [Data set]. Zenodo. https://doi.org/10.5281/zenodo.1214456

Kuncheva, L. I. (2014). *Combining pattern classifiers: Methods and algorithms.* John Wiley & Sons.

Li, F. F. (2017). *The future of vision and artificial intelligence.* MIT Press.

Miller, D., & Brown, E. W. (2017). Artificial intelligence in medical practice: The question to the answer? *The American Journal of Medicine, 131*(2), 129–133.

Lizuka, O., Kanavati, F., Kato, K., Rambeau, M., Arihiro, K., & Tsuneki, M. (2020). Deep learning models for histopathological classification of gastric and colonic epithelial tumours. *Scientific Reports, 10.*

Minsky, M. (1967). Computational models of thought and language. *En Semantic Information Processing* (pp. 217-250). MIT Press.

National Cancer Institute. (2023). Adenocarcinoma. https://www.cancer.gov

Ng, A. (2016). AI is the new electricity. Recuperado de https://www.andrewng.org.

Nizam, S., Ali, M. A. M., & Abidin, M. R. (2023). AI-generated CT scan image for improving colorectal cancer classification. *2023 IEEE 8th International Conference On Software Engineering and Computer Systems (ICSECS),* 1–6.

Obayya, M., Arasi, M. A., Alruwais, N., Alsini, R., Mohamed, A., & Yaseen, I. (2023). Biomedical image analysis for colon and lung cancer detection using Tuna Swarm Algorithm with deep learning. *IEEE Access, 11*, 94705–94712.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., ... & Chintala, S. (2019). PyTorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems, 32,* 8026–8037.

Rathore, S., Hussain, M., Iftikhar, M. A., & Jalil, A. (2014). Ensemble classification of colon biopsy images based on information rich hybrid features. *Computers in Biology and Medicine, 47,* 76–92.

Rodríguez-Díaz, E., Baffy, G., Lo, W. K., Mashimo, H., Vidyarthi, G., Mohapatra, S. S., & Singh, S. K. (2020). Real-time artificial intelligence-based histological classification of colorectal polyps with augmented visualization. *Gastrointestinal Endoscopy.*

Sarwinda, D., Paradisa, R. H., Bustamam, A., & Anggia, P. (2021). Deep Learning in Image Classification using Residual Network (ResNet) Variants for Detection of Colorectal Cancer. *Procedia Computer Science, 179,* 423-431.

Schiele, S., Arndt, T. T., Martin, B., Miller, S., Bauer, S., Banner, B. M., Brendel, E. M., Schenkirsch, G., Anthuber, M., Huss, R., Märkl, B., & Müller, G. (2021). Deep learning prediction of metastasis in locally advanced colon cancer using binary histologic tumor images. *Cancers, 13.*

Szeliski, R. (2020). *Computer vision: Algorithms and applications.* Springer.

Ullah, U., Maheshwari, D., Gloyna, H. H., & Garcia-Zapirain, B. (2022). Severity classification of COVID-19 patients data using quantum machine learning approaches. *2022 International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME)*, 1–6.

Wolpert, D. H. (1992). Stacked generalization. *Neural Networks, 5*(2), 241–259. https://doi.org/10.1016/S0893-6080(05)80023-1

World Health Organization. (2022). Cancer: Fact sheets. https://www.who.int

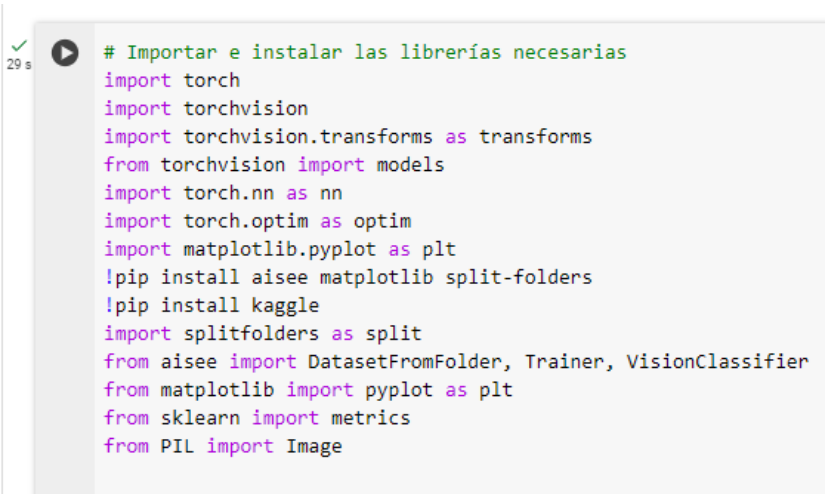
Zhang, H., Wu, C., Zhang, Z., Zhu, Y., Zhang, Z.-L., Lin, H., Sun, Y., He, T., Mueller, J., Manmatha, R., Li, M., & Smola, A. (2020). ResNeSt: Split-Attention Networks. *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2735-2745.

Zhou, Z.-H. (2012). *Ensemble methods: Foundations and algorithms.* CRC Press.

Apéndices

Apéndice A

Importación de Librerías Necesarias



```
# Importar e instalar las librerías necesarias
import torch
import torchvision
import torchvision.transforms as transforms
from torchvision import models
import torch.nn as nn
import torch.optim as optim
import matplotlib.pyplot as plt
!pip install aisee matplotlib split-folders
!pip install kaggle
import splitfolders as split
from aisee import DatasetFromFolder, Trainer, VisionClassifier
from matplotlib import pyplot as plt
from sklearn import metrics
from PIL import Image
```

Nota. importación de librerías necesarias. Obtenido de. Autoría Propia.

Apéndice B

Configuración de Dispositivos

Nota. Configuración de dispositivos. Obtenido de. Autoría Propia.

Apéndice C

Descarga del Dataset

Nota. Descarga del dataset. Obtenido de. Autoría Propia.

Apéndice D

Extracción y Carga del Dataset

Nota. Extracción y carga del dataset. Obtenido de. Autoría Propia.

Apéndice E

División de Datos del Dataset

```
#Dividir carpetas para entrenamiento
input_folder = "/content/data/lung_colon_image_set/colon_image_sets"
output_folder = "/content/data/lung_colon_image_set/colon_image_sets/data"

split.ratio(input = input_folder, output = "data", seed=42, ratio=[0.7, 0.15, 0.15])
```

```
Copying files: 10000 files [00:04, 2154.40 files/s]
```

Nota. División de datos del dataset. Obtenido de. Autoría Propia.

Apéndice F

Creación del Dataset de Entrenamiento

```
# Creamos el dataset de entrenamiento
vtrain_df = DatasetFromFolder("/content/data/train")
```

Nota. Creación del dataset de entrenamiento. Obtenido de. Autoría Propia.

Apéndice G

Parametrización e Instanciación de la Clase Trainer

```python
# Parametrizamos el modelo que utilizaremos
gregory = VisionClassifier(
    model_name="resnet34",
    num_classes=2,
    class_to_idx = {
    "colon_aca": 0,
    "colon_n": 1,
},
    learning_method="freezed",
    task="single_label",
    device="cuda"
)


# Instanciamos la clase Trainer, asignando los parámetros específicos para el entrenamiento del modelo
trainer = Trainer(
    base_model=gregory,
    data= "data",
    output_dir="/content/data/test_trainer.pt",
    batch_size=8,
    num_epochs=10,
    checkpointing_metric="loss",
    shuffle=True,
)
```

Nota. Parametrización e instaciación de la clase trainer. Obtenido de. Autoría Propia.

Apéndice H

Entrenamiento del Modelo

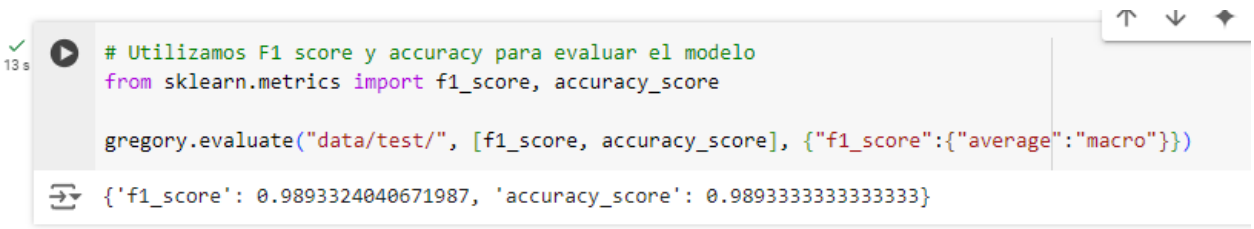
```
# Entrenamos el modelo

trainer.train()
```

```
Initializing Dataloaders...
INFO:aisee.trainer:Initializing Dataloaders...


INFO:aisee.trainer:

Params to learn:
INFO:aisee.trainer:Params to learn:
fc.weight
fc.bias

INFO:aisee.trainer:fc.weight
fc.bias

Epoch 1/10
INFO:aisee.trainer:Epoch 1/10
----------
INFO:aisee.trainer:----------
train batches:  56%|[redacted]     | 491/875 [00:41<00:24, 15.82it/s]
```

Nota. Entrenamiento del modelo. Obtenido de. Autoría Propia.

Apéndice I

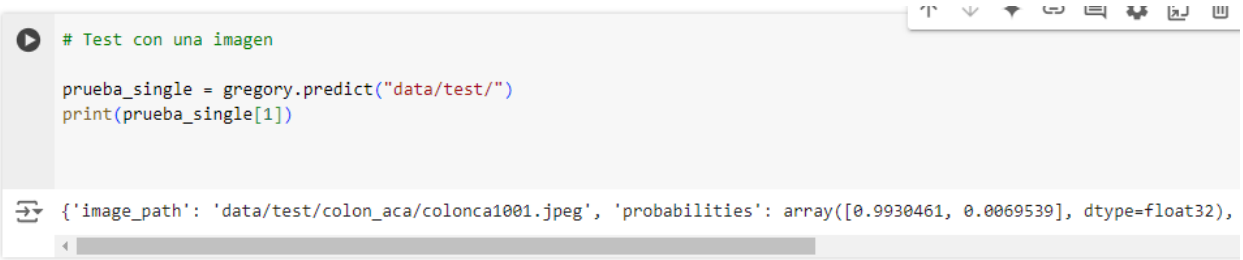
Evaluación del Modelo



Nota. Evaluación del modelo. Obtenido de. Autoría Propia.

Apéndice J

Prueba del Modelo



Nota. Prueba del modelo. Obtenido de. Autoría Propia.

Apéndice K

Enlace de Github a Pasos para la Creación de un Modelo de Clasificación

https://github.com/jedalosa/proyectos/tree/main/proy-grado-tec/pasos

Nota. Enlace de Github a pasos para la creación de un modelo de clasificación. Obtenido de.

Autoría Propia.

Apéndice L

Enlace de Github de Arquitecturas de Redes Neuronales

https://github.com/jedalosa/proyectos/tree/main/proy-grado-tec/arquitecturas

Nota. Enlace de Github de Arquitecturas de redes neuronales. Obtenido de. Autoría Propia.

Apéndice M

Enlace de Github al Entrenamiento de cada Modelo

https://github.com/jedalosa/proyectos/tree/main/proy-grado-tec/entrenamiento

Nota. Enlace de Github al entrenamiento de cada modelo. Obtenido de. Autoría Propia.

Apéndice N

Enlace de GitHub a Validación del Modelo Ensamblado

https://github.com/jedalosa/proyectos/blob/main/proy-grado-tec/Modelo%20Ensamblado%20y%20prueba/Modelo_Emsamblado.ipynb

Nota. Enlace de Github a la validación del modelo ensamblado. Obtenido de. Autoría Propia.

Apéndice O

Enlace de GitHub a Testing del Modelo Ensamblado

https://github.com/jedalosa/proyectos/blob/main/proy-grado-tec/Modelo%20Ensamblado%20y%20prueba/Prueba_modelo_ensamblado.ipynb

Nota. Enlace de Github a la testing del modelo ensamblado. *Obtenido*. Autoría Propia.

Apéndice P

Enlace de GitHub a Testing con Imágenes Sanas y Enfermas de Pulmón y Seno

https://github.com/jedalosa/proyectos/blob/main/proy-grado-tec/Modelo%20Ensamblado%20y%20prueba/Pruebas_modelo_ensamblado_c%C3%A1ncer_de_pulm%C3%B3n_y_c%C3%A1ncer_de_seno.ipynb

Nota. Enlace de GitHub a testing con imágenes sanas y enfermas de pulmón y seno. *Obtenido*. Autoría Propia.