Mejoras en la Clasificación de Tumores Cerebrales con CNNs Utilizando AGCWD con Doble Corrección Gamma por Bloques

1st Raúl Pérez Núñez
Microse
Centro de Investigación en Computación
Ciudad de México, México
rperezn2023@cic.ipn.mx

***Abstract*—La detección temprana de tumores cerebrales en imágenes de resonancia magnética (IRM) puede apoyar al especialista médico, especialmente en contextos donde el acceso a diagnóstico es limitado. En este trabajo se analiza el impacto del preprocesamiento avanzado de imágenes mediante técnicas basadas en ecualización adaptativa (AGCWD), su variante con doble corrección gamma y una modificación propuesta que emplea particiones por bloques. Se evalúa el desempeño de dichas técnicas como etapa previa a una arquitectura convolucional tipo ResNet para la clasificación de imágenes de cerebros con y sin presencia de tumor. Para ello, se utilizaron 7,194 imágenes distribuidas mediante validación cruzada de cinco particiones. Los resultados muestran que el preprocesamiento influye significativamente en la precisión del clasificador, pudiendo mejorar la detección de características relevantes sin incrementar el tamaño del modelo, lo que lo convierte en una alternativa útil para sistemas de diagnóstico asistido por computadora.**

***Index Terms*—component, formatting, style, styling, insert**

I. Introduction

La identificación de tumores cerebrales a partir de imágenes de resonancia magnética (IRM) es una tarea crítica que normalmente requiere la intervención de especialistas. Sin embargo, factores como la disponibilidad de personal médico, tiempos de atención y carga hospitalaria pueden retrasar el diagnóstico. En este contexto, los sistemas de diagnóstico asistido mediante aprendizaje profundo ofrecen una alternativa para complementar la labor del especialista y reducir los tiempos de análisis. Las Redes Neuronales Convolucionales (CNNs) han demostrado ser una de las herramientas más eficaces en la detección y clasificación de anomalías en imágenes médicas. No obstante, su desempeño puede verse afectado por el sobreajuste o por variaciones en la calidad de las imágenes de entrada. Por ello, el preprocesamiento juega un papel importante para resaltar características relevantes y estabilizar la distribución de intensidades antes del entrenamiento. En este trabajo se estudia el efecto de distintas técnicas de preprocesamiento —principalmente el método AGCWD, su variante con doble corrección gamma y una modificación propuesta basada en la partición por bloques— sobre la precisión de una arquitectura ResNet aplicada a la clasificación de IRMs. El objetivo es determinar si estas técnicas pueden mejorar la identificación temprana de tumores cerebrales sin incrementar la complejidad del modelo. Todo el proyecto se puede observar en https://github.com/RaulPerez298/Deep_learning_mri_cancer_cerebro

II. Estado de arte

Planteamiento del problema

La detección de tumores en imágenes médicas requiere un especialista en la interpretación de estas. Frecuentemente, el diagnóstico está sujeto a la disponibilidad de los servicios médicos (equipo, personal y procesos administrativos), por lo que el diagnóstico asistido por computadora contribuye a agilizar el proceso de revisión de algunos estudios médicos. Las Redes Neuronales Convolucionales (CNNs) han demostrado ser útiles en la detección y segmentación de tumores cerebrales a partir de Imágenes de Resonancia Magnética (IRMs).

Motivación

Los tumores cerebrales se clasifican en dos principales grupos: 1) tumores primarios, los cuales se originan en el mismo cerebro y 2) tumores secundarios, cuyo origen se encuentra en otro órgano del cuerpo y posteriormente migró al cerebro, implantándose por el proceso de metástasis. Los tumores primarios más frecuentes son el meningioma y el glioblastoma; en el caso de los secundarios, las metástasis más frecuentes son de cáncer pulmonar, mama y piel [1].

A nivel mundial, los tumores cerebrales son la segunda causa de muerte por cáncer en la población pediátrica. En la población adulta, el glioblastoma es el tumor que ocasiona mayor mortalidad, en cuyo caso el 39.3% sobrevive un año a partir del diagnóstico y el 5.5% sobrevive cinco años. Los tumores primarios del sistema nervioso central tienen una incidencia de 21.42 por cada 10,000 habitantes y los tumores secundarios de 10 por cada 10,000 habitantes. En México, durante el año 2019, el cáncer de encéfalo y otras

partes del sistema nervioso central representó la segunda causa de muerte por tumor maligno en menores de 15 años [2].

La tasa de supervivencia es el porcentaje de personas (que tienen en común el mismo tipo de tumor) que permanecen con vida durante cierto tiempo posterior al diagnóstico. La tasa de supervivencia es útil para obtener una aproximación del tiempo de vida de un paciente que presente algún tipo de tumor. De acuerdo con el Central Brain Tumor Registry of the United States (CBTRUS), entre los años 2001 y 2015 la tasa relativa de supervivencia a 5 años en pacientes con glioblastoma es menor del 22% para todas las edades y en meningioma de 84% para pacientes de 20 a 44 años y de 74% para personas mayores de 44 años [3]. Los tumores cerebrales ocupan el lugar 19 entre todas las neoplasias, y el décimo entre las más letales. Anualmente se detectan 300 mil casos nuevos de tumores cerebrales en el mundo, que corresponden al 2.5% de la mortalidad por cáncer [2].

La detección automática planteada ayudará al especialista, quien, utilizando estos algoritmos y otros estudios, puede generar el diagnóstico en etapas tempranas. También puede apoyar a los estudiantes de medicina mediante la verificación de sus resultados cuando se está aprendiendo a interpretar IRMs del cerebro. Por lo tanto, este trabajo busca determinar características que indiquen variaciones o anomalías en las imágenes, que pueden ser precursores de tumores cerebrales, sin olvidar que el diagnóstico generado sirve como apoyo al personal médico, por lo que será necesario en todos los casos la ratificación del médico especialista.

Resumen de literatura, métodos o puntos de referencia relevantes

Diversos estudios e investigaciones han empleado IRMs para la detección y segmentación de tumores. A continuación, se describen los métodos que algunos investigadores han seguido para desarrollar ambas tareas. Los algoritmos de aprendizaje supervisado como las CNNs y las máquinas de soporte vectorial han sido los más utilizados.

Chattopadhyay y Maitra implementaron una CNN para la segmentación de tumores cerebrales en IRM y reportaron una eficiencia del 99.74%. También mencionaron en sus conclusiones que el uso de CNN en el procesamiento de imágenes es mucho mejor que otros algoritmos de aprendizaje supervisado como máquinas de soporte vectorial, con el cual obtuvieron una precisión de sólo 20.83%. Aunque el objetivo de su investigación no es precisamente el mismo, sus resultados sugieren que las CNNs superan por mucho a otros algoritmos cuando se trata de analizar IRMs, por lo tanto y en relación con el trabajo aquí presentado, comparten el estudio de redes CNNs, pero con la diferencia de las arquitecturas y esquemas de redes neuronales propuestos [4].

Wahlang y otros utilizaron arquitecturas de aprendizaje profundo para clasificar IRM del cerebro en dos clases: normales y anormales. También propusieron una técnica basada en una CNN de aprendizaje profundo y una Red Neuronal Profunda (DNN). Asimismo, se implementaron otras arquitecturas de aprendizaje profundo como LeNet, AlexNet, ResNet y enfoques tradicionales como SVM para analizar y comparar los resultados. Las redes más profundas, como AlexNet y ResNet, no alcanzaron los resultados deseados, aunque los mismos autores atribuyen tales desempeños a que las cantidades de datos usadas para el entrenamiento estaban desbalanceadas y a que las redes profundas son precisas cuando se manipulan grandes cantidades de datos, que en su caso eran limitadas. La comparación de diferentes clasificadores coincidió con la investigación anterior, concluyendo que el desempeño de las CNNs y DNN supera por mucho a las técnicas tradicionales como las máquinas de soporte vectorial o la red AlexNet, ya que las precisiones alcanzadas fueron 88%, 80%, 64% y 82% respectivamente [5].

Rao y Goswami realizaron un estudio comparativo del desempeño en segmentación de tumores cerebrales en IRM entre técnicas de aprendizaje supervisado, específicamente el algoritmo de los k-vecinos más cercanos, y técnicas no supervisadas como k-mínimos y operadores morfológicos. A diferencia de las investigaciones anteriores, no se utilizaron CNNs. Otra diferencia relevante es la cantidad de imágenes utilizadas: se emplearon 100 IRMs para ejecutar cada técnica, mientras que en las investigaciones anteriores se utilizaron alrededor de 2,500 imágenes. En cuanto a la precisión, mientras los autores reportan valores mayores al 90%, las técnicas de k-vecinos más cercanos, k-mínimos y operadores morfológicos aplicados a las imágenes no superaron el 80% (79%, 68% y 60% respectivamente) [6].

Waghmare y Kolekar propusieron una CNN de arquitectura VGG-16 y obtuvieron una precisión de 95.71% segmentando tumores cerebrales en IRMs. Su metodología menciona el uso de una red neuronal con 13 capas convolucionales, aunque ninguna de ellas residual. También reportaron haber utilizado un preprocesamiento de imágenes que consistía en normalizar sus dimensiones y utilizar técnicas de realce de bordes; sin embargo, atribuyeron su alta precisión a la variedad de imágenes de entrenamiento y no al preprocesamiento [7].

Hemanth et al. centraron su investigación en la segmentación de tumores y no en la clasificación de IRMs en normales o anormales. Es el único artículo citado que reconoce como parte relevante el preprocesamiento de imágenes para la extracción de bordes en un tumor. De acuerdo con los autores, esto contribuyó a la eliminación de datos innecesarios y atípicos, al suavizado de datos ruidosos y a la normalización de las imágenes [8].

La Universidad de Washington también ha explorado las CNNs. En su trabajo utilizaron una arquitectura conocida como ESPNet, cuya particularidad radica en el tipo de

convoluciones: a diferencia de las operaciones convencionales que reducen la cantidad de píxeles a procesar, esta arquitectura expande la imagen colocando píxeles con valor cero entre los píxeles internos antes de realizar la convolución [9].

El estado del arte sugiere que las arquitecturas de CNNs más estudiadas corresponden a AlexNet, LeNet y VGG-16. En este documento se propone explorar el comportamiento de la arquitectura ResNet. Además, los efectos del preprocesamiento de imágenes en la detección y segmentación no han sido objeto de estudio en la literatura reciente; en muchos casos se menciona como un paso previo al entrenamiento del modelo, pero no se analiza su impacto en la precisión de la clasificación. En este trabajo se revisan los cambios que el preprocesamiento de imágenes puede causar en el desempeño de la arquitectura en cuestión.

Descripción del conjunto de datos El uso de redes neuronales tiene como finalidad analizar el impacto que cada técnica de procesamiento aporta en la clasificación de imágenes de resonancia magnética. Para ello, se emplearon 7,194 imágenes de resonancia magnética de cerebros humanos: 3,594 correspondientes a pacientes con diagnóstico de tumor y 3,594 etiquetadas como cerebros sanos. Además, se reservaron 358 imágenes exclusivamente para pruebas (empleadas para calcular las matrices de confusión). En total, se utilizaron 6,836 imágenes. Estas imágenes se distribuyeron en 5 grupos mediante validación cruzada de 5 particiones, manteniendo el 20% de los datos para prueba y el 80% para entrenamiento en cada iteración. Esta técnica permite evaluar de manera robusta el desempeño de cada modelo, generando en total 5 conjuntos de pesos neuronales por arquitectura probada. Las imágenes provienen de bases de datos públicas disponibles en el repositorio *Kaggle*, referenciadas en [10] y [11]. La base original también fue procesada mediante la técnica propuesta, obteniéndose una nueva versión adicional del conjunto de datos. Se optó por estas bases públicas debido a la amplia cantidad de imágenes disponibles; aunque existen bases con mayor detalle diagnóstico, no fue posible acceder a datos hospitalarios reales debido a restricciones de privacidad de pacientes.

A. AGCWD

En [12] se introduce un método de procesamiento de imágenes basado en la segmentación por bloques, donde la modificación de la tonalidad de cada píxel se realiza mediante la función de reasignación definida en la Ecu. 1. Esta ecuación es la responsable de ajustar los niveles de intensidad de la imagen.

Para su formulación, se consideran las siguientes variables: el término $cdf_w(l)$ se define en la Ecu. 2, donde l_{max} representa el valor máximo de intensidad en la imagen (comúnmente 255). Por su parte, $pdf_w(l)$ está especificado en la Ecu. 3, mientras que pdf corresponde a la función de densidad de probabilidad mostrada en la Ecu. 4. Esta función equivale al histograma de la imagen y se calcula a partir del total de píxeles $M \times N$.

A partir de dicha distribución, se determinan los valores mínimo y máximo del histograma, denotados como pdf_{min} y pdf_{max}, respectivamente. La suma acumulada $\sum pdf_w$ aparece en la Ecu. 5, y finalmente, el parámetro γ se define mediante la Ecu. 6.

$$T(l) = l_{max}\left(\frac{l}{l_{max}}\right)^{\gamma} = l_{max}\left(\frac{l}{l_{max}}\right)^{1-cdf_w(l)} \tag{1}$$

$$cdf_w(l) = \sum_{l=0}^{l_{max}} \frac{pdf_w(l)}{\sum pdf_w} \tag{2}$$

$$pdf_w(l) = pdf_{max}\left(\frac{pdf(l) - pdf_{min}}{pdf_{max} - pdf_{min}}\right) \tag{3}$$

$$pdf(l) = \frac{n_l}{MN} \tag{4}$$

$$\sum pdf_w = \sum_{l=0}^{l_{max}} pdf_w(l) \tag{5}$$

$$\gamma = 1 - cdf_w(l) \tag{6}$$

B. AGCWD con doble corrección gamma

En el procesamiento descrito por *AGCWD*, el parámetro gamma es una función decreciente, dado que se define como $\gamma = 1 - cdf_w(l)$. Esta característica puede provocar que las zonas oscuras de la imagen reciban un realce limitado. En [13] se plantea una alternativa en la que gamma se comporta como una función creciente, para lo cual se introducen los parámetros γ_1 y γ_2, definidos en las Ecs. 8 y 9, respectivamente. En particular, γ_2 se aproxima gradualmente a 1, lo que permite evitar un realce excesivo en las regiones con alta iluminación.

$$T(l) = l_{max}\left(\frac{l}{l_{max}}\right)^{\gamma} \tag{7}$$

$$\gamma_1 = \frac{\ln\left(e + cdf(l)\right)}{8} \tag{8}$$

$$\gamma_2 = \frac{1 + cdf_w(l)}{2} \tag{9}$$

C. Modificación al método AGCWD dual gamma

Tras analizar los efectos de dividir una imagen de diversas maneras y aplicar ecualización de histograma a cada segmento, se planteó una técnica alternativa. En este enfoque se decidió emplear AGCWD, aprovechando la estructura cuadriculada que surge durante su cálculo. Para ello se utilizó el parámetro γ_2 definido en la Ecu. 9; al sustituirlo en la Ecu. 7 se obtiene la expresión mostrada en la Ecu. 10. En esta formulación, $T(l)$ representa una sección de la imagen, considerando que la imagen completa se divide en una cuadrícula de $m \times n$ bloques. Cada bloque tiene dimensiones $\frac{w}{m} \times \frac{z}{n}$, donde w corresponde

al número de píxeles en filas de la imagen original y z al número de píxeles en columnas.

$$T(l) = l_{max}\left(\frac{l}{l_{max}}\right)^{\gamma_2} = l_{max}\left(\frac{l}{l_{max}}\right)^{\frac{1+cdf_w(l)}{2}} \quad (10)$$

Como ejemplo, en la Fig. 1 se aprecia cómo una imagen puede dividirse en una cuadrícula de 8×8, generando así un total de 64 bloques.

El método propuesto genera un conjunto de k imágenes a las cuales se aplica el procesamiento AGCWD utilizando γ_2. Posteriormente, todas estas imágenes serán promediadas. Cada una de ellas se obtiene aplicando una cantidad distinta de divisiones en filas y columnas. Para determinar estas divisiones se introduce un parámetro adicional denominado offset, cuya finalidad es establecer un límite inferior para el número de particiones tanto horizontales como verticales en la imagen original (se recomienda que este valor no sea menor a 10).

La relación empleada para calcular estas divisiones se muestra en la Ecu. 11, donde k_i indica el número de secciones correspondientes a la imagen i, la cual posteriormente se utilizará en el promedio. Este número determina cuántas particiones se aplicarán a cada una de las k imágenes sobre las que se ejecutará AGCWD.

$$k_i = i - 1 + offset: \quad i = 1, 2, 3, \ldots, k \quad (11)$$

Para ilustrar el procedimiento, considérese el siguiente caso: se desea aplicar el método propuesto a una imagen bajo los parámetros siguientes. Se define un offset de 15 (valor mínimo recomendado para filas y columnas) y se generan 7 imágenes procesadas con AGCWD (es decir, $k = 7$). La Tabla I muestra las particiones resultantes para cada imagen i derivada de la imagen original.

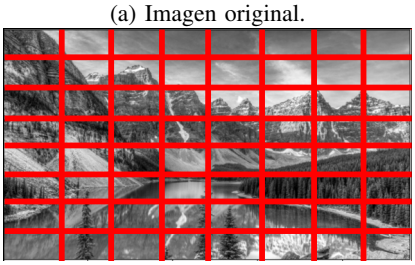
TABLE I: Divisiones para renglones y columnas, para 7 imágenes con $offset$ de 15

i (imágen)	divisiones en renglón columna	secciones totales obtenidas
1	15	225
2	16	256
3	17	289
4	18	324
5	19	361
6	20	400
7	21	441

III. Desarrollo del proyecto

En la Fig. 2 se muestran tanto las imágenes originales como las obtenidas tras aplicar la técnica propuesta. En cada caso, la imagen experimenta modificaciones distintas, principalmente en la iluminación de ciertas regiones, lo que puede resultar en un incremento del contraste o de la saturación. Aunque estos ajustes pueden mejorar la percepción visual, es necesario evaluar si realmente contribuyen a incrementar la precisión de los modelos de clasificación, que es el propósito central de esta investigación.

(a) Imagen original.



(b) Modificación con 8 divisiones en renglones y columnas.

Fig. 1: Aplicando a la imagen 8 divisiones en renglones y columnas, teniendo 64 secciones a calcular.

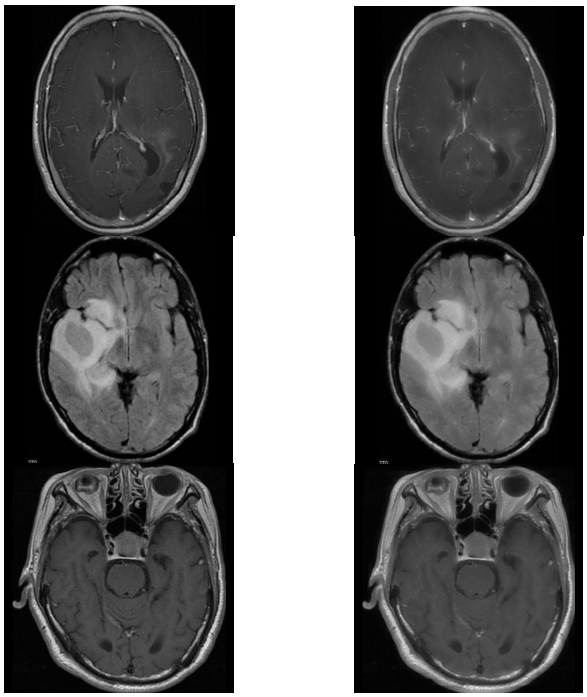


Fig. 2: Aplicación del procesamiento de imagen *gamma modificado* a 3 imágenes de resonancia magnética del cerebro diagnosticadas con tumores, la primera columna son las imágenes originales, segunda con el procesamiento propuesto.

A. Arquitectura de las redes neuronales

En este trabajo se optó por implementar dos arquitecturas ampliamente utilizadas en tareas de clasificación de imágenes: ResNet-50 e Inception-ResNet V2. Ambas redes aprovechan conexiones residuales y, en el caso de la segunda, también conexiones paralelas. Dado que el objetivo es ex-

clusivamente la clasificación, se emplearon únicamente redes convolucionales (*CNN*), prescindiendo de modelos RNN o Transformers (estos últimos más orientados a procesamiento de texto o, en casos particulares, de video). Adicionalmente, se realizó un entrenamiento alternativo de ResNet-50 utilizando funciones de activación *ELU*, debido a sus ventajas frente a ReLU, así como una versión que incorpora bloques tomados de otras arquitecturas como parte de un modelo propuesto. A continuación se describe brevemente cada arquitectura, su estructura y la justificación del uso de *ELU*.

1) ResNet (Residual Networks): La arquitectura ResNet, presentada en 2015 por investigadores de Microsoft [14], está inspirada en el patrón de conexión existente entre neuronas biológicas, donde una misma neurona puede enviar su salida a dos neuronas distintas. En particular, una de estas conexiones dirige la salida hacia la neurona adyacente, idea que dio origen a los dos bloques característicos de ResNet: Residual Identity y Residual Convolutional. El comportamiento del bloque *Residual Identity* puede describirse mediante la ecuación:

$$y = F(x, W_i) + x \tag{12}$$

donde:

- x: entrada del bloque convolucional.
- W_i: pesos de las capas convolucionales (puede haber varias, conectadas de forma secuencial).
- F: función de activación aplicada después de las convoluciones.
- y: salida del bloque convolucional.

Este tipo de conexión también se denomina paso por identidad, ya que la entrada se suma directamente a la salida producida por la secuencia de capas convolucionales. El segundo bloque, denominado Residual Convolutional, se caracteriza por generar dos rutas de procesamiento: la primera aplica varias capas convolucionales encadenadas, mientras que la segunda emplea una capa convolucional adicional destinada principalmente a ajustar las dimensiones (tanto en filas como en columnas) antes de realizar la suma. Su formulación matemática es:

$$y = F(x, W_i) + W_s x \tag{13}$$

donde:

- x: entrada del bloque convolucional.
- W_i: pesos correspondientes a las capas convolucionales principales, típicamente con n filtros cada una.
- W_s: pesos de la convolución utilizada para ajustar dimensiones (generalmente una única capa con n filtros).
- F: función de activación de las capas convolucionales.
- y: salida final del bloque.

2) Inception-ResNet V2: El modelo Inception-ResNet surge de la combinación de dos enfoques arquitectónicos distintos: la familia Inception y las redes residuales ResNet. Cada una aporta mecanismos clave que potencian el desempeño de la red final.

Las arquitecturas Inception se caracterizan principalmente por su diseño de aprendizaje en paralelo, donde una misma entrada se procesa simultáneamente mediante varias capas convolucionales con configuraciones distintas. Las salidas de estas ramas paralelas se concatenan, permitiendo capturar diferentes tipos de características de una misma imagen. Gracias a ello, el modelo obtiene una representación más rica antes de alimentar las capas posteriores, incrementando su capacidad de aprendizaje.

Inception-ResNet V2 integra esta lógica de procesamiento paralelo con las conexiones residuales propias de ResNet. De esta forma, dentro de cada bloque residual, las capas que transforman la entrada ya no se estructuran únicamente como una secuencia de convoluciones, sino como un conjunto de convoluciones organizadas en paralelo. Esto incrementa la diversidad de características extraídas dentro del propio bloque residual, garantizando que la suma con la ruta de identidad incorpore representaciones más completas y robustas [15].

B. Implementación de los modelos

1) Procedimiento para la implementación de la arquitectura ResNet en su versión 50:

1) **Implementación** Para la clasificación de imágenes de resonancia magnética (IRM), se empleó la arquitectura ResNet-50, la cual incorpora múltiples bloques residuales. El propósito de estos bloques es facilitar la extracción de características relevantes de las imágenes, aumentando así la capacidad de aprendizaje de la red.

   En la Figura 4 se presenta la estructura general del modelo. En la Figura 3 se ilustran los dos tipos de bloques residuales utilizados: *Residual Identity* y *Residual Convolutional*. El bloque *Residual Identity* está formado por tres capas convolucionales dispuestas en serie y una conexión de atajo que omite dichas capas. En contraste, el bloque *Residual Convolutional* también contiene tres capas convolucionales, pero incorpora además una convolución paralela dedicada a ajustar las dimensiones anteriores a la suma residual.
   Tras cada bloque, la información procesada por las capas convolucionales se combina mediante suma con la información que transitó por la ruta de identidad (que no atravesó dichas capas). Esta fusión de representaciones incrementa la estabilidad del entrenamiento y mitiga la degradación del gradiente, fortaleciendo el desempeño del clasificador.

2) **Entrenamiento de la ResNet-50**
   El entrenamiento se llevó a cabo en tres etapas. Primero se realizó un entrenamiento inicial para obtener los pesos sinápticos base. Posteriormente, se ejecutaron dos fases adicionales retomando esos pesos, con el propósito de refinar la precisión del modelo y estabilizar su convergencia. Los parámetros empleados en cada

etapa se muestran en la Tabla II.

TABLE II: Parámetros de entrenamiento de la ResNet-50

Entrenamiento	α	Épocas
1	1×10^{-4}	10
2	1×10^{-5}	15
3	1×10^{-6}	15

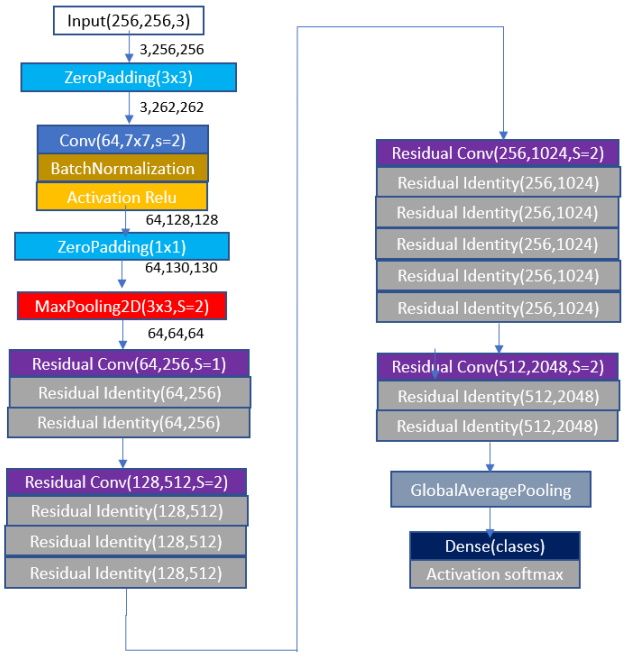


Fig. 3: ARQUITECTURA DE LA RESNET-50

Luego se propuso otro modelo de ResNet-50, pero con una variación utilizando otra función de activación en las capas de convolución, donde se sustituyó $ReLu$ por ELu, y donde la grafica del comportamiento de dicha operación se observa en la figura 5.

2) Procedimiento para la implementación de la arquitectura Inception-ResNet v2:

1) **Implementación**

La arquitectura Inception-ResNet v2 combina dos principios fundamentales: el procesamiento en paralelo y el uso de bloques residuales. El paralelismo se manifiesta cuando varias convoluciones operan simultáneamente sobre la misma entrada, mientras que la parte residual consiste en una ruta de identidad o una convolución que se suma al resultado de dichas ramas paralelas.

La estructura general de la Inception-ResNet v2 (Figura 6) se compone de seis grupos principales de capas convolucionales: Stem, Inception-ResNet-A, Inception-ResNet-B, Inception-ResNet-C, Reduction-A y Reduction-B.

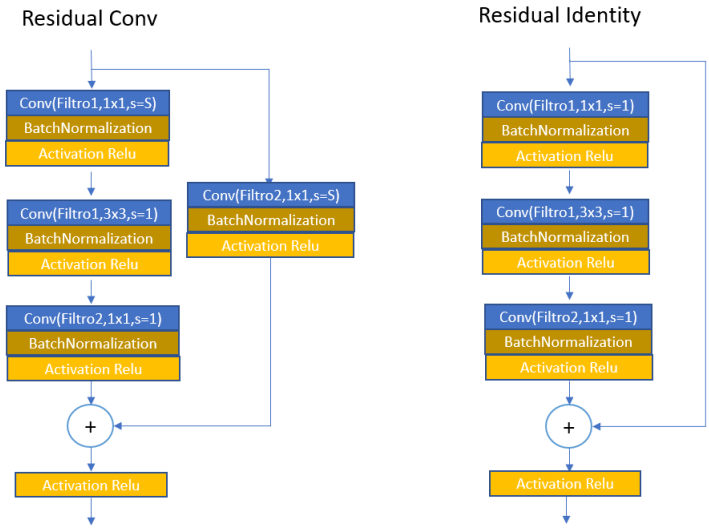


Fig. 4: BLOQUES DE CAPAS RESIDUALES DE LA RESNET-50

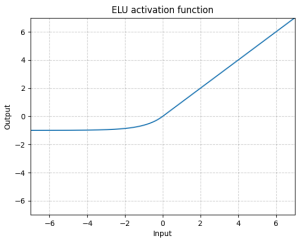


Fig. 5: Gráfica de la función de activación Elu.

a) El procesamiento inicia en el bloque Stem, cuya función es reducir una vez las dimensiones espaciales de la imagen.
b) Posteriormente, los datos ingresan al bloque Inception-ResNet-A, donde los módulos residuales comienzan a extraer características.
c) Luego se aplica el bloque Reduction-A (Figura 11), que disminuye las dimensiones de la imagen en un factor de tres.
d) Los bloques Inception-ResNet-B (Figura 9) y Reduction-B realizan funciones equivalentes a los pasos previos, aunque empleando diferentes configuraciones de filtros para diversificar las características aprendidas.
e) La capa Global Average Pooling sustituye la función de las capas totalmente conectadas mediante una operación de agrupación global.
f) La capa Dropout desactiva aleatoriamente neuronas durante el entrenamiento, favoreciendo un aprendizaje más equilibrado y reduciendo el sobreajuste.
g) Finalmente, el vector resultante ingresa a la capa de clasificación, donde la función Softmax produce

la salida final de la red.

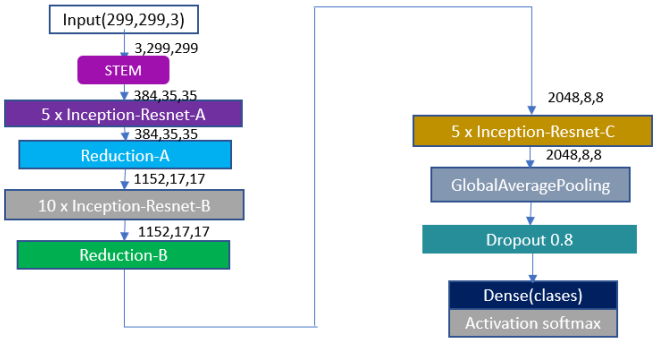


Fig. 6: ARQUITECTURA DE LA INCEPTION-RESNET V2

Bloque: STEM

El bloque Stem constituye la primera sección convolucional de la red y su función principal es reducir las dimensiones espaciales (alto y ancho) de la imagen mediante operaciones de MaxPooling y convoluciones con un desplazamiento de dos píxeles. Las arquitecturas Inception se caracterizan por el uso de múltiples ramas convolucionales que procesan la misma entrada en paralelo; posteriormente, sus salidas se concatenan para generar una representación más completa. En el bloque Stem este mecanismo se aplica en tres ocasiones, logrando una reducción equivalente en las dimensiones de entrada.

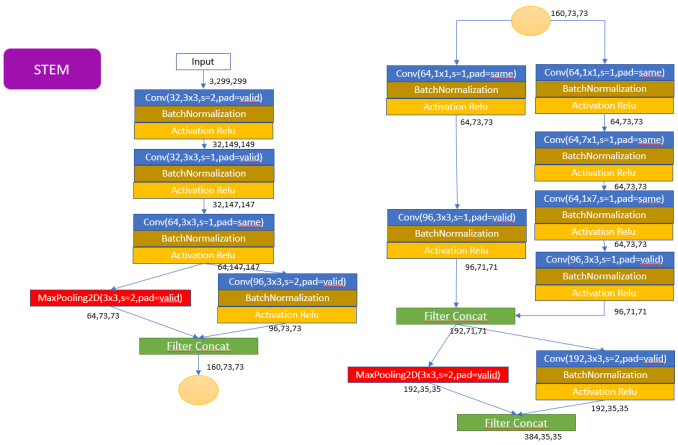


Fig. 7: BLOQUE STEM

Bloques: Inception-ResNet-A, Inception-ResNet-B e Inception-ResNet-C

Estos bloques integran dos elementos clave. El primero es la conexión residual, donde cada módulo incorpora una ruta principal —compuesta por varias convoluciones— y una ruta secundaria que preserva la identidad o aplica una convolución de ajuste; ambas rutas se combinan mediante una suma. El segundo es el paralelismo inherente a la familia Inception: cada módulo incluye múltiples ramas convolucionales que comparten la misma entrada, cuyas salidas son concatenadas y posteriormente sumadas con la rama residual. Dado que estos módulos no alteran la resolución espacial de las características, se aplican repetidamente en cadena. En este proyecto, los bloques Inception-ResNet-A, Inception-ResNet-B e Inception-ResNet-C se repiten 5, 10 y 5 veces respectivamente (Figuras 8-10).

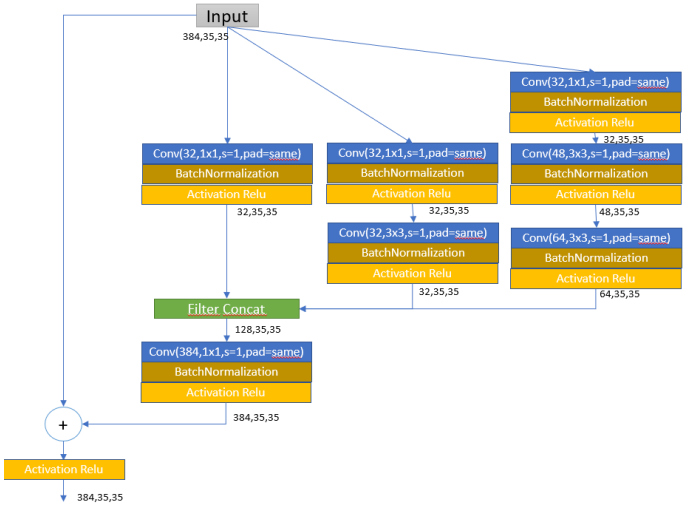


Fig. 8: BLOQUE INCEPTION-RESNET A

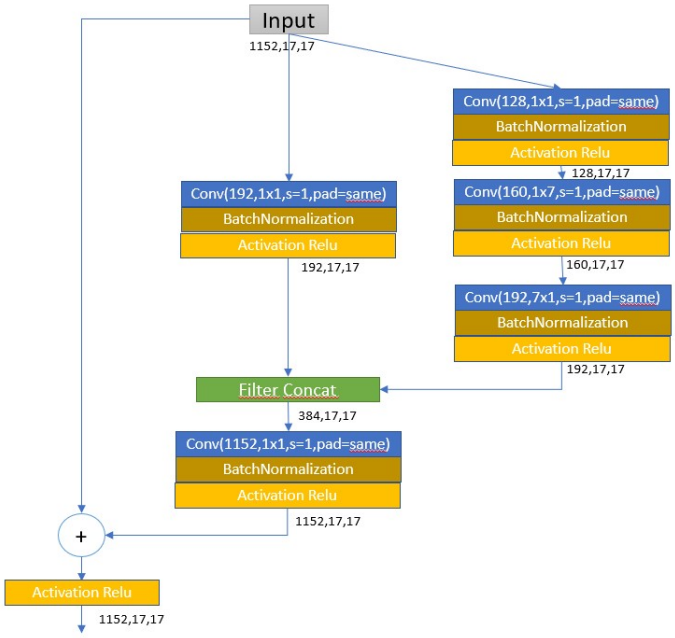


Fig. 9: BLOQUE INCEPTION-RESNET B

Bloques: Reduction-A y Reduction-B

Los bloques Reduction-A y Reduction-B cumplen una función similar al bloque Stem, ya que su propósito

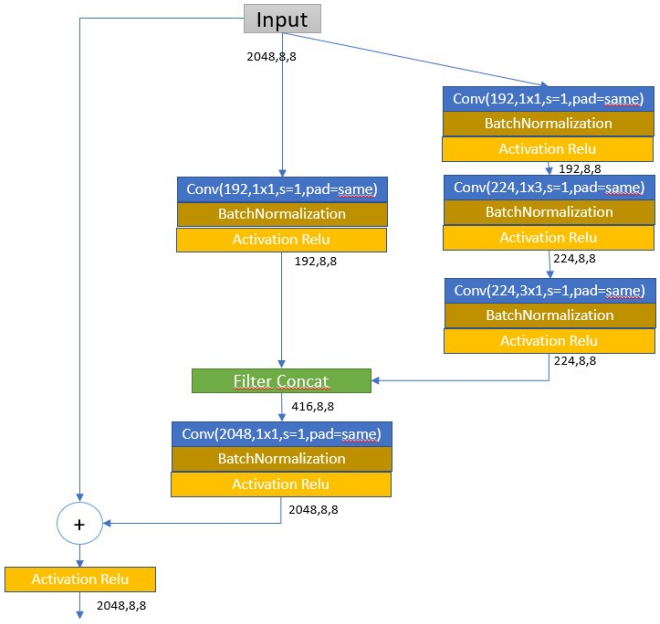


Fig. 10: BLOQUE INCEPTION-RESNET C

es disminuir las dimensiones espaciales de la imagen. Sin embargo, a diferencia de Stem —que efectúa tres reducciones consecutivas—, cada uno de estos bloques realiza una única reducción por vez. Ambos mantienen el principio de las ramas paralelas; Reduction-A utiliza tres rutas convolucionales y Reduction-B cuatro, lo que incrementa la diversidad de características generadas respecto al máximo de dos ramas presentes en el bloque Stem.

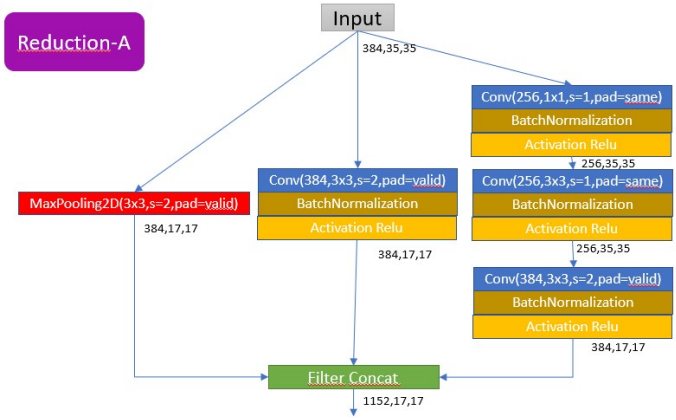


Fig. 11: BLOQUE REDUCTION A

2) **Entrenamiento de la Inception-ResNet v2**
El entrenamiento del modelo se llevó a cabo en dos fases. En la primera se calcularon los pesos sinápticos iniciales de la red y en la segunda se realizó un ajuste adicional para mejorar el rendimiento del clasificador. Los parámetros utilizados durante este proceso se

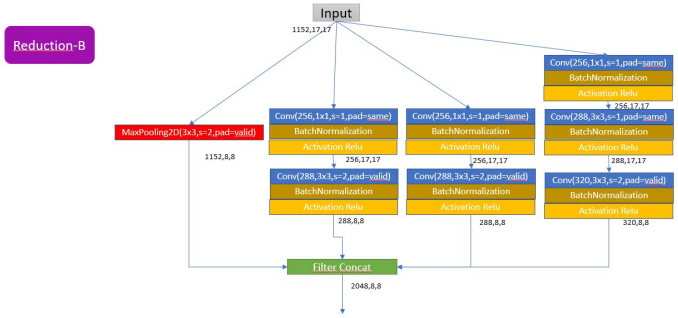


Fig. 12: BLOQUE REDUCTION B

detallan en la Tabla III.

TABLE III: Parámetros de entrenamiento de la Inception-ResNet v2

Entrenamiento	α	Épocas
1	1×10^{-4}	10
2	1×10^{-5}	15

3) Modelo propuesto para detectar tumores y clasificarlos: Este modelo fue construido con base a 3 estructuras de datos mostradas en las redes neuronales ResNet-50 y Inception-ResNet V2. De la ResNet-50 se usaron los bloques de Residual Conv y Resicual Identity. De Inception-ResNet V2 se uso el bloque de Reduction B, estos bloques fueron modificados en sus funciones de activación, sustituyendo Relu por Elu, en [16] donde $\alpha = 1.0$, donde fue utilizado en las capas de Convolución, en las capaz densas todavia se utilizaron la funciones *ReLu*. Donde se modela *Elu* por la ecuación siguiente, como el gradiente de la función:

$$f(x) = \begin{cases} x & x > 0 \\ \alpha(exp(x) - 1) & x \leq 0 \end{cases}$$

$$f'(x) = \begin{cases} 1 & x > 0 \\ f(x) + \alpha & x \leq 0 \end{cases}$$

Se muestra la estructura completa del modelo propuesto en la Figura 13, donde en este caso se utilizaron una mayor cantidad de capaz densas, al igual que *Dropout*, donde

TABLE IV: Parámetros de entrenamiento para el modelo propuesto

Entrenamiento	α	Épocas
1	$1x10^{-4}$	10
2	$1x10^{-5}$	15
3	$1x10^{-6}$	15

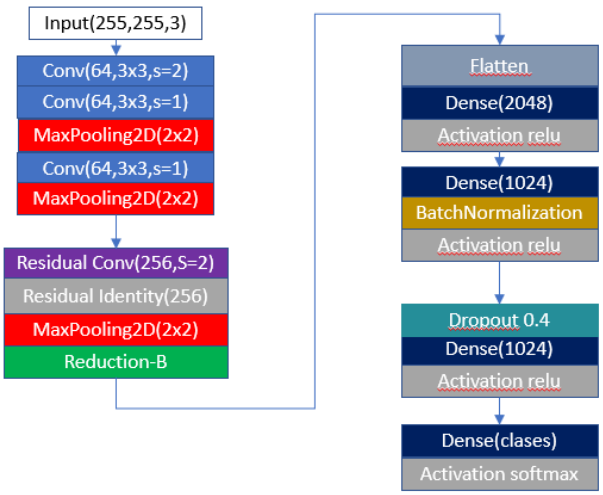


Fig. 13: MODELO PROPUESTO.

C. Procedimiento de Entrenamiento

El entrenamiento se realizó utilizando imágenes normalizadas en el rango $[0, 1]$ mediante `ImageDataGenerator(rescale=1/255)`, donde se entrenaron los modelos 2 veces, una vez para los datos sin procesar y la otra con el metodo propuesto.

1) Función de Pérdida: Debido a que el problema es de clasificación binaria con dos clases mutuamente excluyentes, se empleó la función de pérdida *categorical cross-entropy* [17] definida como:

$$\mathcal{L}(\theta) = -\sum_{i=1}^{N}\sum_{c=1}^{2} y_{i,c}\log \hat{y}_{i,c}, \tag{14}$$

donde $y_{i,c}$ es la etiqueta codificada en one-hot y $\hat{y}_{i,c}$ es la probabilidad predicha por el modelo para la clase c, obtenida mediante una capa *softmax* al final de la red:

$$\hat{y}_{i,c} = \frac{\exp(z_{i,c})}{\sum_{k=1}^{2}\exp(z_{i,k})}, \tag{15}$$

con $z_{i,c}$ representando el logit correspondiente a la clase c. Esta función penaliza con mayor severidad las predicciones incorrectas con alta confianza, favoreciendo modelos mejor calibrados y estables durante la retropropagación.

2) Optimizador: Se empleó el optimizador Adam [18], que combina los beneficios de AdaGrad y RMSProp mediante promedios móviles de primer y segundo momento:

$$m_t = \beta_1 m_{t-1} + (1-\beta_1) g_t, \tag{16}$$

$$v_t = \beta_2 v_{t-1} + (1-\beta_2) g_t^2, \tag{17}$$

$$\theta_{t+1} = \theta_t - \alpha \frac{m_t}{\sqrt{v_t}+\epsilon}, \tag{18}$$

donde $g_t = \nabla_\theta \mathcal{L}(\theta_t)$ es el gradiente en el paso t. La tasa de aprendizaje se ajustó manualmente siguiendo la secuencia:

$$\alpha = 10^{-i}, \qquad i \in [4, \alpha_{\text{final}}], \tag{19}$$

evaluando el desempeño del modelo tras cada ciclo y conservando los pesos con mayor *validation accuracy*.

3) Hiperparámetros: Los hiperparámetros principales utilizados durante el entrenamiento se resumen en la Tabla V.

TABLE V: Hiperparámetros del Entrenamiento

Hiperparámetro	Valor
Tamaño de batch	32
Épocas iniciales	10
Épocas posteriores por α	11–15 (según arquitectura)
Optimizador	Adam
α inicial	10^{-4}
α final	10^{-6} o 10^{-7}
Métrica	Accuracy

4) Inicialización de Pesos: Todas las capas convolucionales utilizaron la inicialización por defecto de Keras, *Glorot Uniform* [19] (también conocida como *Xavier Uniform*). Este método inicializa los pesos a partir de una distribución uniforme acotada:

$$W \sim \mathcal{U}\left(-\sqrt{\frac{6}{n_{\text{in}}+n_{\text{out}}}}, \sqrt{\frac{6}{n_{\text{in}}+n_{\text{out}}}}\right), \tag{20}$$

donde n_{in} y n_{out} representan el número de unidades de entrada y salida del kernel convolucional, respectivamente.

La inicialización Glorot Uniform busca mantener constante la varianza de las activaciones a través de las capas durante la propagación hacia adelante y hacia atrás. Esto evita la desaparición o explosión del gradiente y proporciona una estabilidad significativa durante el entrenamiento de arquitecturas profundas como ResNet, Inception-ResNet y variantes personalizadas empleadas en este trabajo.

5) Selección del Mejor Modelo: Durante el entrenamiento, los pesos con mayor exactitud en validación fueron almacenados mediante `ModelCheckpoint` utilizando *val_accuracy*. Esto fue implementado para tener los mejores pesos para los modelos que puedan predecir correctamente elementos nunca utilizados para su entrenamiento, dándole una mayor importancia a la predicción.

IV. Resultados

En esta sección se presentan los resultados obtenidos mediante validación cruzada de 5 grupos (5-fold cross-validation). En cada iteración, aproximadamente el 80% de las imágenes se utilizó para entrenamiento y el 20% restante para validación, lo cual garantiza una evaluación estable e independiente del particionado de los datos. Para cada fold, la red neuronal fue entrenada desde cero y se seleccionó el modelo con mejor *validation accuracy*.

Dado que el presente trabajo aborda un problema de diagnóstico médico (detección de tumores en imágenes de resonancia magnética), el uso exclusivo de *accuracy* resulta insuficiente para evaluar un modelo de manera clínica. Por ello, se emplearon como métricas principales **Sensitivity** y

Specificity, ampliamente utilizadas en la literatura médica debido a su relevancia en escenarios donde los falsos negativos y los falsos positivos tienen implicaciones críticas (por ello no fue utilizado en los resultados **Precision** y **Recall**). En concreto, la *Sensitivity* mide la capacidad del modelo para detectar correctamente los casos positivos (tumor), mientras que la *Specificity* cuantifica la capacidad para identificar correctamente los casos negativos (no tumor). La métrica de *Accuracy* se incluye como complemento, pero no se considera la métrica principal debido al posible desbalance de clases común en este tipo de conjuntos de datos.

Las tres métricas se definieron de la siguiente forma:

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \tag{21}$$

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}}, \tag{22}$$

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}. \tag{23}$$

Todas las métricas reportadas corresponden al promedio y desviación estándar obtenidos sobre los 5 folds.

A. Metodología de Evaluación

La metrica fue calculada para cada fold y luego agregada mediante: $\bar{m} = \frac{1}{5}\sum_{i=1}^{5} m_i$

Se reportan resultados como: la media de todas las tablas. Además, todas las figuras y curvas de aprendizaje se generaron utilizando únicamente los datos correspondientes a validación dentro de cada fold.

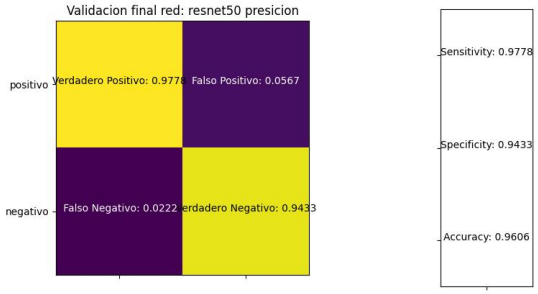
B. Comparación con la Línea Base

No se contó con un modelo base previo para este conjunto de datos. Sin embargo, se eligió como referencia un modelo convolucional clasico del estado del arte (ResNet-50) con el fin de comparar el desempeño. A partir de esta línea base, se evaluaron otra arquitectura compleja como Inception-ResNet-v2, donde se hizo un cambio en la ResNet50 cambiando sus funciones de activación por ELU y creando un modelo propio basado en la Inception-ResNet-v2 y ResNet-50 como ya se habia mencionado previamente.

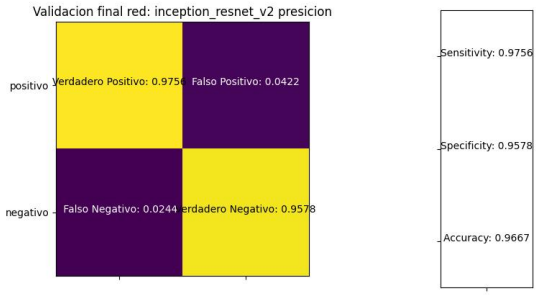
La Figura 14 muestra el desempeño general del modelo al hacer el promedio de los parametros encontrados, promediados de los 5 folds, mientras que la Tabla VI resume las métricas obtenidas para hacer una comparación posteriormente, donde todos estos modelos utilizaron las imágenes sin procesar para el entrenamiento.

TABLE VI: Resultados promedio de 5-fold Cross-Validation, para imagenes sin procesar.

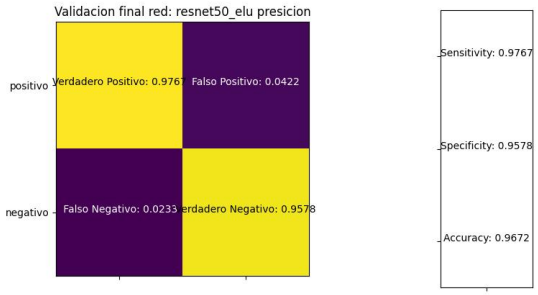
Modelo	Sensitivity	Specificity	Accuracy
ResNet-50	0.9778	0.9433	0.9606
Inception-ResNet-V2	0.9756	0.9578	0.9667
ResNet-50-ELU	0.9767	0.9578	0.9672
Modelo propuesto	0.9856	0.9789	0.9822



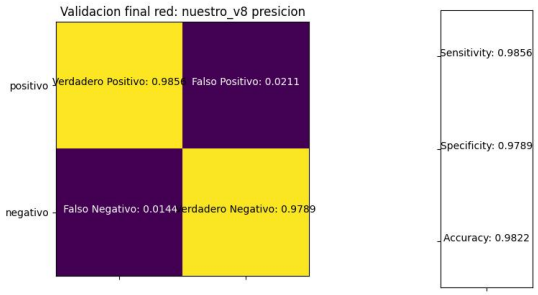
(a) Matriz de confusión de ResNet-50.



(b) Matriz de confusión de Inception-ResNet-v2.

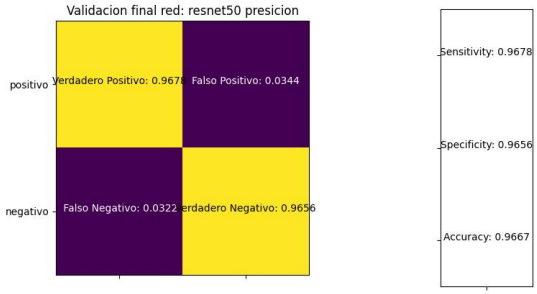


(c) Matriz de confusión de ResNet-50-ELU.

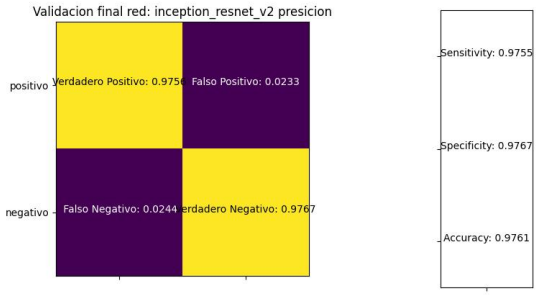


(d) Matriz de confusión del modelo propuesto.

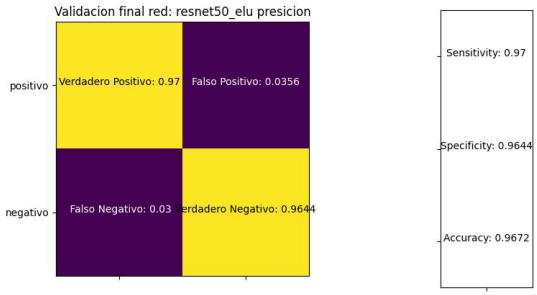
Fig. 14: Matrices de confusiones de las CNN entrenadas con MRI sin procesar.



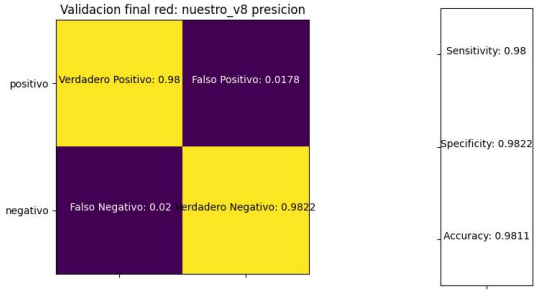
(a) Matriz de confusión de ResNet-50.



(b) Matriz de confusión de Inception-ResNet-v2.



(c) Matriz de confusión de ResNet-50-ELU.



(d) Matriz de confusión del modelo propuesto.

Fig. 15: Matrices de confusiones de las CNN entrenadas con MRI con procesamiento propuesto usando Gamma.

TABLE VII: Resultados promedio de 5-fold Cross-Validation, para imagenes procesadas con gamma.

Modelo	Sensitivity	Specificity	Accuracy
ResNet-50	0.9678	0.9656	0.9667
Inception-ResNet-V2	0.9755	0.9767	0.9761
ResNet-50-ELU	0.97	0.9644	0.9672
Modelo propuesto	0.98	0.9822	0.9811

La Figura 15 muestra el desempeño general del modelo al hacer el promedio de los parametros encontrados, promediados de los 5 folds, mientras que la Tabla VII resume las métricas obtenidas para hacer una comparación posteriormente, donde todos estos modelos utilizaron las imágenes ya procesadas con el metodo propuesto usando gamma.

C. Ablaciones y Variaciones

Para estudiar la influencia de las decisiones de diseño, se realizaron dos tipos de experimentos:

- **Variación de función de activación**: Se comparó ResNet50 con ReLU contra ResNet50 con ELU. Los resultados mostraron que ELU ofrece una mayor estabilidad durante el entrenamiento y tendió a mejorar la *Specificity*, sugiriendo una mejor capacidad de detección de pacientes sanos, al igual que mejoro la *Accuracy*, dando una mejoría global tomando en cuanta todos los datos.
- **Efecto del preprocesamiento**: Se evaluó el desempeño utilizando imágenes originales frente a imágenes procesadas mediante un método propio basado en corrección gamma. Este preprocesamiento mejoró la Specificity del modelo, al igual que en los modelos de alta cantidad de capas convolucionales hubo una mejoria en el Accuracy, exceptuando al modelo propuesto el cual usa menos capas.
- **Un modelo propuesto**: Se comparo el modelo propuesto con los ya existentes al entrenarlos al igual que con la variación de las ResNet-50-ELU, donde este modelo es el que obtiene mejores resultados, tenendo cada uno de los parametros de evalución por arriba del 0.98, dando mejores resultados, que apesar que empeoro un poco en la *Sensitiity* y *Accuracy* al aplicar la mejora de imagen, sigue presentando mejores resultados.

D. Interpretación de Resultados

Los resultados obtenidos permiten concluir lo siguiente:

- Las arquitecturas profundas, como *Inception-ResNet-v2* y *ResNet50*, presentan un desempeño elevado; sin embargo, es posible incrementar su rendimiento mediante ajustes específicos, como se observó con la variante modificada. Esto se refleja en las mejoras registradas en *sensitivity*, *specificity* y *accuracy*.
- Aunque modelos profundos como *Inception-ResNet-v2* y *ResNet50-ELU* ofrecen resultados competitivos, los

experimentos mostraron que no es imprescindible contar con redes excesivamente profundas. El modelo implementado, con menor número de capas, logró un rendimiento equiparable o superior en varios indicadores.
- La corrección gamma propuesta tiene un impacto notable en el incremento del *specificity*, característica especialmente valiosa en aplicaciones donde es prioritario reducir la cantidad de falsos negativos.
- En el contexto de esta investigación, la métrica más relevante es la *Sensitivity*, ya que la detección del tumor es indispensable debido al riesgo que implica para la vida del paciente. Bajo este criterio, la mejor alternativa fue el modelo propuesto sin procesamiento adicional de imágenes. No obstante, en escenarios donde también sea esencial identificar con precisión casos sanos, el modelo con la mejora de imagen resulta más adecuado. Esto se debe a que, al contar con menos capas, la red sufre una menor pérdida de información entre las etapas convolucionales y las capas de clasificación.

E. Limitaciones

Entre las principales limitaciones identificadas se encuentran:

- Dependencia en conjuntos de datos relativamente pequeños.
- Sensibilidad a variaciones de calidad en las imágenes MRI.
- Arquitecturas grandes requieren más tiempo de entrenamiento y memoria.
- La baja cantidad de imagenes publicas existentes.

V. Conclusión

Los modelos evaluados mostraron que tanto las arquitecturas profundas como las de menor complejidad pueden alcanzar un desempeño competitivo en la clasificación de imágenes médicas. Esto indica que una mayor profundidad no siempre se traduce en mejores resultados y que modelos más ligeros pueden ser igualmente eficaces. Todo el proyecto se puede observar en https://github.com/RaulPerez298/Deep_learning_mri_cancer_cerebro

El análisis evidencia que las técnicas de preprocesamiento —particularmente la corrección gamma— influyen de manera significativa en las métricas de desempeño. Se observó un aumento notable en la especificidad, lo cual es relevante cuando se busca reducir falsos positivos.

Debido a que la detección de un tumor tiene implicaciones directas en la vida del paciente, la sensibilidad se estableció como la métrica de mayor importancia. En este estudio, el modelo propuesto sin preprocesamiento adicional obtuvo el mejor desempeño en esta métrica, por lo que representa una alternativa confiable en escenarios clínicos donde no se pueden omitir casos positivos.

Los resultados demuestran que es posible obtener un rendimiento elevado sin depender de modelos extremadamente profundos. Esto sugiere que arquitecturas más ligeras pueden implementarse eficientemente en sistemas médicos reales, reduciendo el costo computacional sin comprometer la precisión. Este trabajo evidencia que no siempre es necesario recurrir a modelos complejos para lograr un desempeño relevante en aplicaciones médicas.

References

[1] L. Contreras, Epidemiología de tumores, Rev. Med. Clin. condes, vol. 3, nº 28, pp. 332-338, 2017.

[2] INEGI, Estadísticas a propósito del día mundial contra el cáncer, Dirección de Atención a Medios, Ciudad de México , 2021.

[3] T. Wyant, American Cancer Society, 05 05 2020. [En línea]. Available: https://www.cancer.org/es/cancer/tumores-de-encefalo-o-de-medula-espinal/deteccion-diagnostico-clasificacion-por-etapas/tasas-de-supervivencia.html. [Último acceso: 15 05 2025].

[4] A. C. y. M. Maitra, MRI-based brain tumourimage detection using CNN based deep learning method, Neuroscience Informatics, vol. 2, nº 100060, pp. 1-6, 2022.

[5] e. a. I. Wahlang, Brain Magnetic Resonance Imaging Classification Using Deep Learning Architectures with Gender and Age, Sensors, vol. 22, nº 1766, pp. 1-20, 2022.

[6] B. R. y. M. Goswami, Proceedings of the International Conference on Intelligent Systems, de 671, Singapore, 2018.

[7] V. W. y. M. Kolekar, Convolutional Neural Network-Based Automatic Brain Tumor Detection, de Evolving Technologies for Computing, Communication and Smart World, Singapore, Springer Nature, 2021, pp. 463-475.

[8] G. e. al., DESIGN AND IMPLEMENTING BRAIN TUMOR DETECTION USING MACHINE LEARNING APPROACH, de Proceedings of the Third International Conference on Trends in Electronics and Informatics, EUA, 2019.

[9] S. Mehta, RSNA, 20 noviembre 2018. En línea. Available: https://homes.cs.washington.edu/shapiro/sachinchapter_edit_report.pdf. Último acceso: 24 mayo 2022.

[10] M. Nickparvar, "Brain Tumor MRI Dataset," *Kaggle*, 2021. [Online]. Available: https://www.kaggle.com/masoudnickparvar/brain-tumor-mri-dataset

[11] A. Panigrahi, "Brain_Tumor_Detection_MRI," *Kaggle*, 2021. [Online]. Available: https://www.kaggle.com/datasets/abhranta/brain-tumor-detection-mri

[12] S. -C. Huang, F. -C. Cheng and Y. -S. Chiu, "Efficient Contrast Enhancement Using Adaptive Gamma Correction With Weighting Distribution," in IEEE Transactions on Image Processing, vol. 22, no. 3, pp. 1032-1041, March 2013, doi:10.1109/TIP.2012.2226047.

[13] Y. Chang, C. Jung, P. Ke, H. Song, and J. Hwang, "Automatic Contrast-Limited Adaptive Histogram Equalization With Dual Gamma Correction," *IEEE Access*, vol. 6, pp. 11782–11792, 2018. doi: doi:10.1109/ACCESS.2018.2797872.

[14] K. He *et al.*, "Deep Residual Learning for Image Recognition," arXiv:1512.03385, 2015. [Online]. Available: https://arxiv.org/abs/1512.03385

[15] C. Szegedy *et al.*, "Rethinking the Inception Architecture for Computer Vision," arXiv:1602.07261, 2016. [Online]. Available: https://arxiv.org/abs/1602.07261

[16] D. A. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs)," arXiv:1511.07289, 2016. [Online]. Available: https://arxiv.org/abs/1511.07289

[17] C. E. Shannon, "A Mathematical Theory of Communication," *Bell System Technical Journal*, vol. 27, no. 3, pp. 379–423, 1948.

[18] D.P.Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," *arXiv preprint arXiv:1412.6980*, 2014. [Online]. Available: https://arxiv.org/abs/1412.6980

[19] X.Glorot and Y.Bengio, "Understanding the Difficulty of Training Deep Feedforward Neural Networks," in *Proc. AISTATS*, pp. 249–256, 2010.