

Santiago de Cali, enero de 2025

Doctor

Gerardo Mauricio Sarria

Director de Carrera

Ingeniería de Sistemas y Computación

Pontificia Universidad Javeriana Cali

Ref. Trabajo de grado

Cordial Saludo

Por medio de este comunicado estamos presentándole a los estudiantes Juan José Aguado y Luis Alfredo Rodríguez quienes desarrollaron su trabajo de grado bajo nuestra dirección. El título de este es “Modelo de clasificación de planos cerebrales en ecografías prenatales utilizando aprendizaje de máquina profundo” el cual está terminado y listo para su respectiva sustentación.

Atentamente,

Cristian Alejandro Torres Valencia

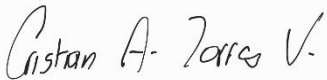
Nombre del director

Docente de planta

Felipe Palta

Nombre del codirector

Docente hora catedra



Firma del director



Firma del codirector

Santiago de Cali, enero de 2025

Señores

Pontificia Universidad Javeriana Cali.

Dr. Gerardo Mauricio Sarria

Director de Carrera de Ingeniería de Sistemas y Computación

Ref. Trabajo de grado

Cordial Saludo

Nos permitimos presentar a su consideración el trabajo de grado terminado, con el título de “Modelo de clasificación de planos cerebrales en ecografías prenatales utilizando aprendizaje de máquina profundo” con el propósito de cumplir los requisitos exigidos por la Universidad para llevar a cabo el proyecto de grado y posteriormente optar al título de Ingeniero de Sistemas y Computación.

Al firmar aquí, doy fe que conocemos y entendemos las directrices para la presentación de Trabajos de Grado de la Facultad de Ingeniería y Ciencias aprobadas el 26 de noviembre de 2009, donde se establecen los plazos y normas para el desarrollo del anteproyecto y del trabajo de grado.

Atentamente,

Juan José Aguado
Nombre estudiante



Firma estudiante

Luis Alfredo Rodríguez
Nombre estudiante



Firma estudiante



Facultad de Ingeniería y Ciencias
Carrera de Ingeniería de sistemas y computación

Modelo de clasificación de planos cerebrales en ecografías prenatales utilizando aprendizaje de máquina profundo

Juan José Aguado Valderrama
aguado4@javerianacali.edu.co

Luis Alfredo Rodríguez López
fredo0522@javerianacali.edu.co

Santiago de Cali, febrero de 2025

Abstract

Prenatal Cranial Ultrasound (PCU) plays a pivotal role in monitoring fetal development. However, evaluating PCU images between 18 and 40 weeks of pregnancy presents significant challenges due to the lack of automated classification tools. Current diagnoses rely heavily on specialist interpretation, often leading to long hours of manual image classification for deeper analysis and data storage. This manual process is particularly burdensome in resource-limited areas, where there is a shortage of skilled professionals and high-quality equipment. Automating the classification of these images has the potential to alleviate specialist workloads and enhance the efficiency and accuracy of PCU image analysis.

This study proposes the development of a deep learning-based tool to classify different brain planes (transventricular, transthalamic, and transcerebellar) in PCU images. The approach includes image preprocessing, segmentation, and classification using convolutional neural networks. Additionally, the model will be evaluated through metrics such as accuracy, precision, recall, and F1-score, with validation by healthcare professionals.

Keywords: Prenatal cranial ultrasound, deep learning, brain plane classification, medical image processing, healthcare automation.

Resumen

Las Ecografías Craneales Prenatales (ECP) desempeñan un papel crucial en el monitoreo del desarrollo fetal. No obstante, la evaluación de imágenes de ECP entre las semanas 18 a 40 del embarazo enfrenta grandes retos debido a la falta de herramientas automatizadas para su clasificación. Actualmente, el diagnóstico depende en gran medida de la interpretación de especialistas, lo que implica largas horas de clasificación manual para análisis más profundos y el almacenamiento de la información. Esta situación se ve agravada en áreas con recursos limitados, donde la escasez de profesionales capacitados y equipos de alta calidad puede comprometer la precisión y accesibilidad de las evaluaciones. La automatización de la clasificación de imágenes ECP podría aliviar la carga de trabajo de los especialistas y mejorar la eficiencia en los análisis.

Este estudio propone desarrollar una herramienta basada en aprendizaje profundo para clasificar los distintos planos del cerebro (transventricular, transtalámico y transcerebeloso) en imágenes de ECP. La metodología incluye el preprocesamiento de imágenes, segmentación y clasificación mediante redes neuronales convolucionales. Además, se evaluará el rendimiento del modelo utilizando métricas como precisión, exactitud, sensibilidad y puntuación F1, con validación por profesionales de la salud.

Palabras clave: Ecografía craneal prenatal, aprendizaje profundo, clasificación de planos cerebrales, procesamiento de imágenes médicas, automatización en salud.

Índice

1. Introducción	12
2. Descripción del problema	13
2.1. Planteamiento del Problema	13
2.2. Objetivos	13
2.2.1. Objetivo General	13
2.2.2. Objetivos Específicos	13
2.3. Justificación	13
2.4. Alcances y límites	14
3. Marco Teórico	15
3.1. Ecografías	15
3.1.1. Importancia de las ecografías prenatales	15
3.1.2. Funcionamiento de las ecografías	15
3.1.3. Desafíos en la evaluación de imágenes ecográficas	16
3.1.4. Importancia de los planos	16
3.2. Procesamiento digital de imágenes	18
3.2.1. Desenfoque gaussiano	18
3.2.2. Máscaras binarias	18
3.2.3. Ecuación adaptativa del histograma	18
3.2.4. Descomposición en planos de bits	19
3.2.5. Corrección gamma	20
3.2.6. Operaciones morfológicas	21
3.3. Visión por computadora	22
3.4. Teoría del Aprendizaje de Máquina	22
3.4.1. Métodos de aprendizaje supervisado, no supervisado y de refuerzo	22
3.4.2. Preprocesamiento de imágenes	23
3.4.3. Aumento de datos	23
3.4.4. Aprendizaje Profundo	24
3.4.5. Estrategias de entrenamiento y validación	27
3.4.6. Métricas de análisis de desempeño	27
4. Trabajos Relacionados	30
5. Metodología para la clasificación de planos cerebrales en ECP	31
5.1. Selección de la base de datos	32
5.2. Preprocesamiento y segmentación de ECP y no ECP	32
5.2.1. Procesamiento de imagen y máscara	33
5.2.2. Realzo de detalles, suavizado y corrección gamma	34
5.2.3. Plano de bits	35
5.2.4. Operaciones morfológicas	36
5.2.5. Reducción de ruido y eliminación de ramas	37
5.2.6. Interpolación de segmentos del cráneo	38
5.2.7. Comparación con curvas de referencia	41
5.3. Modelos de aprendizaje profundo para la clasificación de planos cerebrales	42
5.3.1. Etiquetado	42
5.3.2. Ajustes de características de las imágenes	42
5.3.3. Aumento de datos y balanceo	43
5.3.4. Entrenamiento	45

6. Pruebas y Resultados	52
6.1. Validación del algoritmo de segmentación	52
6.2. Validación de los modelos de aprendizaje automático	52
6.2.1. Métricas	52
6.2.2. Validación con profesionales de la salud	58
6.2.3. Interpretación y comparación	63
7. Conclusiones	65
8. Trabajos Futuros	66
9. Referencias	67
10. Anexos	71

Índice de figuras

1.	Planos cerebrales. Fuente: [1]	16
2.	Diferencias y regiones de interés de cada plano cerebral. Fuente: [9].	17
3.	Máscara binaria aplicada para segmentar un tipo de alimento. Fuente: [51].	18
4.	Imagen oscura antes de ecualizarse. Fuente: [55].	19
5.	Imagen después de aplicarle ecualización adaptativa del histograma. Fuente: [55].	19
6.	Planos de bits de una imagen. Fuente: [12].	19
7.	Efecto de la corrección gamma en una imagen. Fuente: [10].	20
8.	Operaciones de dilatación y erosión en una imagen. Fuente: [35].	21
9.	Operaciones de apertura y cierre respectivamente en una imagen. Fuente: [35].	21
10.	Operación esqueletización en una imagen. Fuente: [35].	22
11.	Arquitectura interna DenseNet.[58].	25
12.	Bloque de construcción del aprendizaje residual [25].	26
13.	Se divide la imagen en una cuadrícula $S \times S$ y para cada celda de la cuadrícula predice B recuadros delimitadores, la confianza de esos recuadros y probabilidades de clase C . Estas predicciones se codifican como $S \times S \times (B \cdot 5 + C)$ [46].	27
14.	Tabla de matriz de confusión.	28
15.	Secuencia del pipeline para la elaboración del proyecto	31
16.	Secuencia del pipeline para la clasificación de ECP	32
17.	Pipeline del algoritmo de segmentación	33
18.	Ejemplo del procesamiento de imagen y su máscara	34
19.	Diferente valores gamma para la imagen después de incorporar CLAHE y el filtro Gaussiano	35
20.	Diferentes planos de bits para la imagen procesada	35
21.	Operaciones morfológicas en diferentes planos cerebrales	36
22.	Ejemplo de reducción de ruido y eliminación de ramas	38
23.	Límite superior e inferior respecto a un umbral del 20 %	39
24.	Ejemplo 1: Adquisición de coordenadas borde superior e inferior del cráneo	39
25.	Ejemplo 2: Adquisición de coordenadas borde superior e inferior del cráneo	40
26.	Ejemplo 1: Interpolación cúbica de imagen ecográfica	40
27.	Ejemplo 2: Interpolación cúbica de imagen ecográfica	41
28.	Imágenes de curvas de las 10 curvas de referencia para borde de cráneo superior	41
29.	Imágenes de curvas de las 10 curvas de referencia para borde de cráneo superior	42
30.	Secuencia del pipeline para la aumentación de datos	43
31.	Histograma de clases antes del balanceo.	44
32.	Histograma de clases después del balanceo.	44
33.	Contraste de la imagen original en contra las imágenes generadas por el pipeline	45
34.	Estructura jerárquica de la base de datos para el modelo ResNet.	46
35.	Arquitectura de la ResNet-18 para planos cerebrales. Adaptado de [5] con modificaciones.	48
36.	Arquitectura DenseNet. Adaptado de [2] con modificaciones.	49
37.	Formato de etiqueta para imágenes en YOLO.	50
38.	Estructura jerárquica de la base de datos para el modelo YOLO.	50
39.	Arquitectura YOLO: La red de detección tiene 24 capas convolucionales seguidas de 2 capas totalmente conectadas. La alternancia de capas convolucionales 1×1 reduce el espacio de características de las capas precedentes.[46]	51
40.	Matriz de confusión testeo algoritmo de segmentación	52
41.	Métricas por generación entrenamiento ResNet	53
42.	Matriz de confusión testeo ResNet	54
43.	Métricas por generación entrenamiento DenseNet	55
44.	Matriz de confusión testeo DenseNet	55
45.	Métricas de precisión y sensibilidad por generación del modelo YOLO	56

46.	Gráfica de pérdidas por generación en entrenamiento y validación modelo YOLO	57
47.	Matriz de confusión testeo YOLO	58
48.	Matriz de confusión testeo Dr. Juan Camilo Aguado	59
49.	Matriz de confusión testeo Dra. Natalia Andrea Torres	60
50.	Matriz de confusión testeo ResNet	61
51.	Matriz de confusión testeo DenseNet	62
52.	Matriz de confusión testeo YOLO	63
53.	Distribución de precisión de las pruebas con ResNet con subconjuntos de 200 imágenes aleatorias	71
54.	Distribución de precisión de las pruebas con ResNet con subconjuntos de 200 imágenes balanceadas	71

Índice de cuadros

1.	Manipulaciones básicas de las imágenes con una pequeña descripción [62].	24
2.	Tabla de arquitecturas ResNet.[24].	47
3.	Tabla de arquitecturas DenseNet.[27].	49
4.	Informe para la segmentación de imágenes craneales y no craneales	52
5.	Informe de clasificación para cada clase en el modelo ResNet	53
6.	Informe de clasificación para cada clase en el modelo DenseNet	54
7.	Informe de clasificación para cada clase en el modelo YOLO	56
8.	Informe de clasificación para cada clase en las predicciones de Camilo	59
9.	Informe de clasificación para cada clase en las predicciones de Natalia	60
10.	Reporte de clasificación para cada clase en las predicciones del Modelo ResNet	61
11.	Reporte de clasificación para cada clase en las predicciones del Modelo DenseNet	62
12.	Reporte de clasificación para cada clase en las predicciones del Modelo YOLO	63
13.	Informe de clasificación para los modelos ResNet, DenseNet y YOLO	64
14.	Métricas de desempeño para cada clase en el modelo YOLO	72

1. Introducción

En el campo medicinal prenatal, las Ecografías Craneales Prenatales (ECP) tienen un papel fundamental en el monitoreo del desarrollo fetal, específicamente entre las semanas 18 y 40 de la gestación, donde el cráneo y sus estructuras internas pueden analizarse con mayor claridad. Sin embargo, actualmente hay desafíos significativos en este campo, mayormente por la dependencia absoluta de la ejecución e interpretación de la ecografía hecha por especialistas y la falta de herramientas automatizadas para clasificar las mismas.

En Colombia existe una marcada escasez de especialistas (1.5 especialistas por cada mil habitantes, en comparación con lo recomendado por la OMS que es 3.5)[47], de los cuales únicamente el 9% de los mismos se dedican a la ginecología y obstetricia. Estas condiciones generan una necesidad de optimizar los recursos médicos disponibles e introducir herramientas que permitan abarcar el problema.

El presente proyecto abarca la problemática por medio del desarrollo de una herramienta automatizada para el procesamiento y segmentación de imágenes de ultrasonido fetal y clasificación de los distintos planos cerebrales (específicamente de las Ecografías Craneales Prenatales). La solución propuesta en este proyecto se basa en implementar una combinación de visión por computadora y aprendizaje automático, utilizando una base de datos con 12.400 imágenes provenientes de 1.792 pacientes. Con este conjunto de datos y la validación con profesionales del campo de la salud, se ha logrado el desarrollo de un sistema que permite tratar este problema.

El objetivo de esta herramienta es alivianar la carga de los especialistas de forma que se facilite la clasificación y almacenamiento de grandes volúmenes de imágenes médicas. El proyecto se enfoca en segmentar las imágenes que son ECP de las que no y luego implementar distintos modelos (que fueron comparados con el rendimiento de los profesionales de la salud) para clasificar los en 3 principales planos cerebrales, los cuales son el plano transtálamico, el plano transcerebeloso y el plano transventricular.

La segmentación se hizo utilizando técnicas de procesamiento digital de imagen para diferenciar las imágenes ECP de las no ECP. Luego se usaron 3 modelos de aprendizaje automático para clasificar los planos. Los modelos usados fueron las redes neuronales convolucionales ResNet, DenseNet y YOLO la cual tuvo una precisión 97% en las pruebas.

2. Descripción del problema

2.1. Planteamiento del Problema

Las Ecografías Craneales Prenatales (ECP) desempeñan un papel trascendental en el monitoreo del desarrollo fetal. Sin embargo, la evaluación de las imágenes ecográficas del cráneo durante las semanas 18 a 40 del embarazo se enfrenta a desafíos significativos debido a la falta de herramientas precisas para la automatización de su clasificación. Actualmente, la clasificación se basa en la interpretación de un especialista, lo que puede implicar largas horas de trabajo dedicadas a clasificar datos para realizar análisis más detallados, almacenar la información para estudios posteriores o para alimentar bases de datos. Además, en áreas con recursos limitados, la escasez de profesionales capacitados y equipos de alta calidad puede comprometer la precisión y la accesibilidad a estas evaluaciones.

La creciente demanda de servicios de atención prenatal y el aumento en el número de embarazos han ejercido una presión adicional sobre los recursos y el personal médico especializado. A medida que más mujeres acceden a este tipo de atención, la carga de trabajo para los especialistas encargados de analizar e interpretar las ecografías prenatales se ha incrementado considerablemente [29]. Esta situación puede llevar a retrasos en los diagnósticos, errores por fatiga o sobrecarga de trabajo, y una atención prenatal menos oportuna.

La automatización de la clasificación de estas imágenes ecográficas podría aliviar la carga de trabajo de los especialistas y facilitar el almacenamiento de la información clasificada, lo que permitiría mayor eficiencia y resultados en análisis posteriores de la información. Sin embargo, se debe mencionar que en el ámbito del procesamiento de imágenes y el aprendizaje automático, uno de los principales desafíos técnicos radica en la diversidad que se muestra en las ecografías craneales prenatales. Estas diferencias incluyen variaciones en la calidad de las imágenes debido al uso de distintos equipos de ultrasonido utilizados, las configuraciones de adquisición y condiciones clínicas del paciente. Además, las características anatómicas del feto cambian significativamente a lo largo de las semanas de gestación, lo que complica la creación de modelos robustos que puedan generalizar adecuadamente.

2.2. Objetivos

2.2.1. Objetivo General

Desarrollar una herramienta para el procesamiento de imágenes de ultrasonido fetal con el fin de reconocer imágenes de ECP y clasificar los diferentes planos cerebrales, basado en técnicas de visión por computadora y aprendizaje automático.

2.2.2. Objetivos Específicos

- Estructurar una base de datos de imágenes de ultrasonido ECP y no ECP para el entrenamiento y validación de los algoritmos de identificación de planos cerebrales.
- Desarrollar una estrategia de preprocesamiento de imágenes de ultrasonido para la diferenciación de imágenes ECP y no ECP, basadas en operaciones en escala de grises.
- Implementar dos modelos de aprendizaje profundo para la clasificación de planos cerebrales en imágenes ECP.
- Validar el desempeño de la herramienta de aprendizaje automático para la clasificación de ECP mediante personal de la salud y métricas de evaluación de algoritmos de clasificación.

2.3. Justificación

Las ecografías prenatales desempeñan un papel trascendental en el monitoreo del desarrollo fetal, gracias a que son poco invasivas, rápidas y presentan información importante dependiendo de la etapa

de desarrollo del feto, tales como el peso y fecha probable de parto. Este tipo de exámenes generan una gran cantidad de imágenes que el especialista necesita procesar y clasificar manualmente.

Actualmente, estas pruebas son totalmente dependientes del operador y se recomiendan ser realizadas por un ginecólogo, según la resolución 3280 de 2018, página 268 [48]. Esto supone un problema, ya que en Colombia solo hay 1.5 especialistas por cada mil habitantes, cuando lo recomendado por la OMS es 3.5 [47], además solo el 9% de estos especialistas son del área de ginecología y obstetricia [39]. Las ecografías son fundamentales en todos los procesos de embarazo, pero hay una falta de herramientas de apoyo para los profesionales en pro de la clasificación de estas imágenes, al crear una solución a esto se puede liberar a los especialistas de esta carga, lo que permitirá que tengan más disponibilidad para atender a más pacientes.

En adición a la importancia propia que tiene la ecografía, hay componentes más importantes que otros para hacer un análisis del estado del feto como lo es la ecografía craneal prenatal, la cual brinda información importante, pues se recopila información de las medidas del cráneo y de los distintos planos del cerebro los cuales pueden proveer de información crítica de interés para los especialistas. Alguna persona se podría cuestionar cuál es la importancia de plantear soluciones de clasificación si no tiene un impacto directo en el diagnóstico y la salud del feto y la madre. Lo anterior es una pregunta razonable, sin embargo, tener más herramientas de apoyo en la clasificación implica un paso en el avance de la relación entre la medicina y la informática, que puede llevar a en un futuro tener ecógrafos que puedan clasificar en tiempo real las distintas partes del cuerpo, con sus respectivos planos e incluso detectar anomalías que se le pueden notificar al doctor. Por ello, a pesar de que se hayan planteado ciertas soluciones para la clasificación, es de vital importancia comparar distintas herramientas y seguir experimentando en este campo.

Además, en la página de Zenodo se dispone de una amplia variedad de datos e imágenes relacionadas con las ecografías prenatales, en específico hay un set de datos que cuenta con 12.400 imágenes que fueron sacadas de 1.792 pacientes por especialistas [19]. No solo esto, sino que también se cuenta con dos profesionales de la salud, el Dr. Juan Camilo Aguado y la Dra. Natalia Andrea Torres, dispuestos a hacer evaluaciones sobre los modelos y dándole legitimidad a los resultados que este provea. Por lo anterior, se puede decir que es viable crear una herramienta entrenada con los suficientes datos y que tenga una evaluación apropiada de los resultados por un profesional.

Por esta razón, crear soluciones basadas en aprendizaje de máquina para clasificar ecografías craneales prenatales tendrá un impacto en la eficiencia y almacenamiento correcto de estas pruebas, siendo una herramienta precisa, veloz y creando más autonomía con respecto al operador bajando la carga del profesional.

2.4. Alcances y límites

El proyecto consta de un modelo de aprendizaje profundo en el cual se clasifican los distintos planos de las imágenes del cerebro identificadas en las ecografías que entran al modelo. Este modelo se basará en el desarrollo de un pipeline con datos de distintas ecografías de mujeres que se encuentran entre las semanas 18 a la 40 del embarazo, en donde se encuentran ecografías de distintas partes del cuerpo de estos fetos, como también distintos planos del cráneo.

Los datos para el entrenamiento provienen de una base de datos un estudio en España en el cual se realizó una investigación dentro de dos hospitales, que recoge 12.400 imágenes de 1.792 pacientes clasificadas por distintas partes del cuerpo (abdomen, cerebro, fémur y tórax) y el cerebro etiquetado mediante sus distintos planos (transtalámico, transcerebeloso, transventricular)[19].

3. Marco Teórico

En este marco teórico se aborda la importancia de las ecografías prenatales como una herramienta fundamental en la detección temprana de anomalías en el desarrollo fetal, destacando su relevancia en la evaluación del cráneo fetal, como también los desafíos presentes debido a la complejidad de las estructuras anatómicas y la variabilidad en la calidad de las imágenes. Con este contexto, se explora el potencial de la visión por computadora y los métodos de aprendizaje de máquina, incluyendo enfoques supervisados, no supervisados y de refuerzo, para mejorar la precisión y la eficiencia en la clasificación de estas imágenes, donde específicamente se examina las redes neuronales convolucionales como la mejor herramienta a utilizar, junto con estrategias de entrenamiento y validación de modelos para optimizar su desempeño.

3.1. Ecografías

Las ecografías son un tipo de imagen diagnóstica no invasiva, las cuales se caracterizan por el uso de ondas sonoras de alta frecuencia. Estas imágenes pueden ser de órganos, tejidos y estructuras corporales, lo cual es útil para monitorear embarazos, diagnosticar la causa de distintas afecciones médicas y guiar ciertos procedimientos de biopsia. [36].

3.1.1. Importancia de las ecografías prenatales

Las ecografías prenatales desempeñan un papel fundamental en el monitoreo y seguimiento del desarrollo fetal durante el embarazo. Estas pruebas de diagnóstico por imagen utilizan ondas de sonido de alta frecuencia para generar imágenes en tiempo real del bebé en desarrollo dentro del útero materno, que permite a los médicos y obstetras evaluar de manera no invasiva el crecimiento y la salud del feto a lo largo de las diferentes etapas del embarazo[19].

A través de las ecografías prenatales, los especialistas pueden detectar una amplia gama de anomalías estructurales, como defectos congénitos, problemas en el crecimiento o la posición del bebé, así como monitorear indicadores clave como la longitud del cuello uterino, el flujo sanguíneo y la cantidad de líquido amniótico[21]. Esta información es crucial para identificar posibles complicaciones que puedan requerir un seguimiento más cercano o incluso un tratamiento prenatal oportuno, lo que contribuye a mejorar significativamente los resultados para la salud de la madre y el bebé.

Adicionalmente, las ecografías prenatales juegan un papel importante en la determinación de la edad gestacional y la fecha estimada de parto, lo que permite a los profesionales de la salud planificar y brindar una atención prenatal adecuada a lo largo del embarazo.

3.1.2. Funcionamiento de las ecografías

Las ecografías son exámenes médicos que utilizan ultrasonido para visualizar estructuras internas del cuerpo humano y son ampliamente empleadas durante el embarazo, ya que diversos estudios han demostrado su seguridad total en este contexto [8].

El principio detrás del ultrasonido es similar a la ecolocalización utilizada por los murciélagos, donde el eco de las ondas sonoras se emplea para calcular la distancia de los objetos. En este caso, un transductor emite ondas de ultrasonido que se reflejan al entrar en contacto con diferentes estructuras. Al recibir los ecos reflejados, el transductor mide el tiempo que tardan en regresar y, con la velocidad del sonido, calcula la distancia entre el dispositivo y el límite de los tejidos. De este modo, se generan imágenes bidimensionales que permiten visualizar los tejidos, órganos y estructuras internas [41]. Dado que el ultrasonido se basa en el eco de ondas sonoras y no en la luz, las imágenes obtenidas son representadas en blanco y negro.

3.1.3. Desafíos en la evaluación de imágenes ecográficas

Los principales desafíos en la evaluación de imágenes ecográficas del cráneo durante las semanas 18 a 40 del embarazo se pueden resumir de la siguiente manera:

1. Falta de herramientas precisas para la automatización de la clasificación: Actualmente, el diagnóstico se basa en la interpretación manual de un especialista, lo que conlleva arduas horas de trabajo y un proceso lento.
2. Variabilidad intrínseca de la prueba: Las imágenes ecográficas pueden presentar diferencias debido a la diversidad de equipos y frecuencias utilizadas en los centros de atención médica. Esta variabilidad inherente a la técnica ecográfica representa un desafío para el desarrollo de soluciones de automatización robustas.
3. Escasez de profesionales capacitados y equipos de alta calidad: En algunas áreas con recursos limitados, la falta de especialistas entrenados y la insuficiencia de equipos de ecografía de alta calidad pueden comprometer la precisión y accesibilidad de estas evaluaciones[29].
4. Creciente demanda de servicios de atención prenatal: El aumento en el número de embarazos y la mayor accesibilidad a los servicios de atención prenatal han ejercido una presión adicional sobre los recursos y el personal médico especializado[29].

Además, distintos artículos relacionados con este proyecto resaltan distintas dificultades como lo son entrenamiento clínico, la variabilidad en las características de la máquina de ultrasonido, aumento de la grasa de la madre, orientación del feto[15], incremento del volumen cerebral del feto, sinaptogénesis¹[28] y datos limitados[7].

3.1.4. Importancia de los planos

Las ecografías se realizan desde diferentes ángulos, lo que permite obtener planos específicos de interés. En el caso del cerebro, los planos más relevantes son el plano transcerebeloso (o transcerebelar), transventricular y transtalámico. A continuación, se detalla la importancia de cada uno de estos planos y su utilidad en la evaluación del desarrollo cerebral.

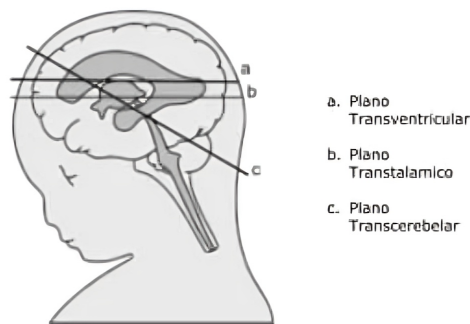


Figura 1: Planos cerebrales. Fuente: [1]

¹La sinaptogénesis es el proceso por el cual se crean sinapsis, es decir, conexiones entre una neurona y otra célula del sistema nervioso, o bien entre dos neuronas. Aunque la sinaptogénesis es especialmente intensa durante el desarrollo temprano, las influencias ambientales influyen en la consolidación y la desaparición de sinapsis durante toda la vida.” [50]

Plano Transventricular: Este plano permite observar las porciones anterior y posterior de los ventrículos laterales. En la porción anterior, las astas frontales de los ventrículos laterales se visualizan como estructuras llenas de líquido con forma de comas, separadas por el *cavum septum pellucidum* (CSP). Esta cavidad, que se visualiza ecográficamente desde las 16 semanas de gestación, suele cerrarse hacia el final del embarazo. La falta de visualización del CSP fuera de las semanas 18 a 37 es un hallazgo común y no necesariamente indica una anomalía [45].

La observación del CSP es importante, ya que la ausencia o alteración de esta estructura puede estar relacionada con condiciones cerebrales como la holoprosencefalia, agenesia del cuerpo calloso o hidrocefalia severa. Además, en la porción posterior del ventrículo, se visualiza el glomus del plexo coroideo, una estructura ecogénica que llena gran parte de la cavidad ventricular, con una pequeña cantidad de líquido presente en condiciones normales [45].

Plano Transcerebeloso: Este plano, obtenido en un nivel ligeramente inferior al transventricular, permite visualizar el cerebelo fetal. El cerebelo se aprecia como una estructura en forma de mariposa, compuesta por los hemisferios cerebelosos y el vermis, una estructura central que muestra una mayor ecogenicidad. Detrás del cerebelo se encuentra la cisterna magna, un espacio lleno de líquido que debe tener una profundidad entre 2 y 10 mm en la segunda mitad de la gestación [45].

Durante las primeras semanas del embarazo, el vermis cerebeloso puede no haber cubierto completamente el cuarto ventrículo, lo que puede generar una impresión errónea de defecto. Sin embargo, en etapas avanzadas del embarazo, una alteración en la cisterna magna puede ser indicativa de una patología del cerebelo [45].

Plano Transtalámico: El plano transtalámico, también conocido como el plano del diámetro biparietal, se utiliza principalmente para realizar mediciones de la cabeza fetal. Este plano permite la visualización de estructuras como los cuernos frontales de los ventrículos laterales, el *cavum septum pellucidum*, los tálamos y los giros del hipocampo. Aunque no proporciona información anatómica adicional significativa respecto a los otros planos, es útil para obtener mediciones precisas del diámetro biparietal, particularmente en las etapas avanzadas del embarazo, lo que ayuda a monitorizar el crecimiento fetal [45].

En la Figura 2 se observan claramente las diferencias y regiones de interés de cada plano las cuales son importantes para la clasificación de cada plano.

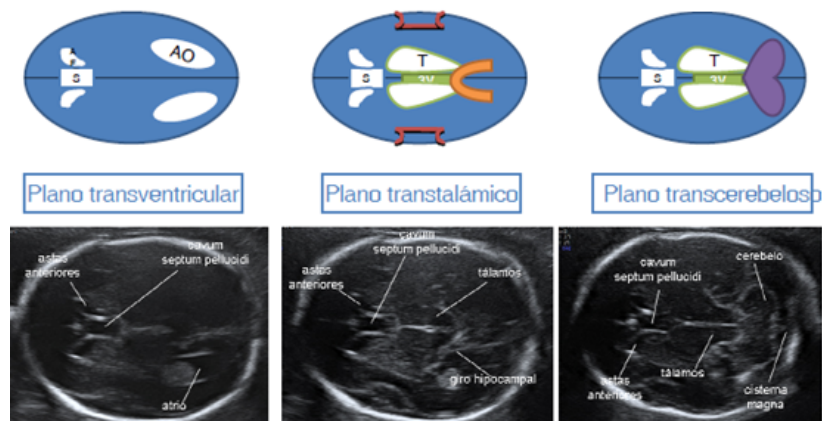


Figura 2: Diferencias y regiones de interés de cada plano cerebral. Fuente: [9].

3.2. Procesamiento digital de imágenes

El procesamiento digital de imágenes (PDI) es un conjunto de técnicas, procesos y algoritmos destinados a modificar, extraer o resaltar información en una imagen mediante operaciones computacionales [17]. Desde el siglo XX, estas técnicas han sido estudiadas y aplicadas para eliminar ruido, mejorar la claridad, extraer características o formas, entre otros objetivos [33]. El PDI ha demostrado ser particularmente útil en áreas como la visualización de imágenes médicas, donde se busca optimizar la precisión y claridad visual, lo cual resulta conveniente para este proyecto. Este campo abarca diversas operaciones, de las cuales se destacan algunas ampliamente utilizadas en imágenes en escala de grises para fines de preprocesamiento.

3.2.1. Desenfoque gaussiano

El desenfoque gaussiano es una operación que afecta a todos los píxeles de una imagen, asignando a cada píxel un valor basado en el promedio de los píxeles cercanos dentro de un radio definido. Cuanto mayor sea este radio, mayor será el grado de desenfoque aplicado a la imagen creando así una imagen borrosa[22]. Estas operaciones pueden resultar útiles cuando se busca suavizar imágenes.

3.2.2. Máscaras binarias

Las máscaras binarias son una técnica de PDI que permite extraer selectivamente una región de una imagen para delimitar una región de interés, esta se puede crear de distintas formas, tanto por delimitación de un polígono por foto-interpretación o con algoritmos de inteligencia artificial [30].

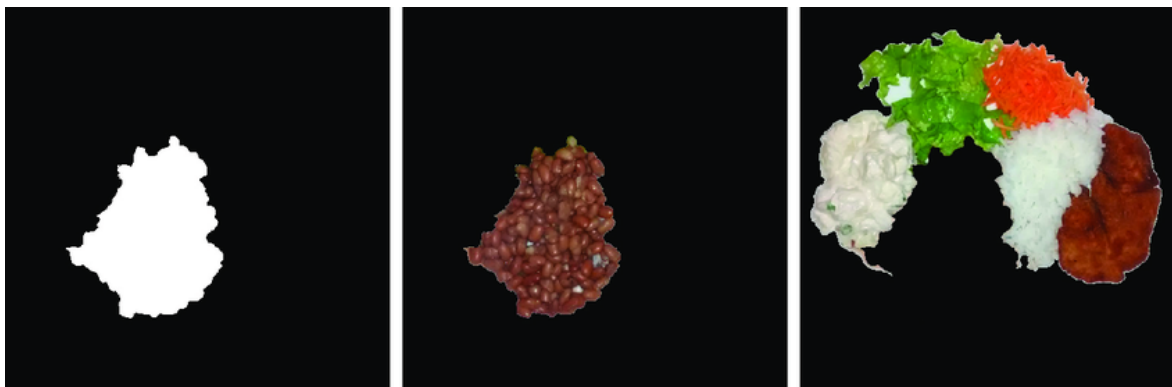


Figura 3: Máscara binaria aplicada para segmentar un tipo de alimento. Fuente: [51].

Como se observa en la Figura 3 la máscara binaria permite seleccionar solo la región de interés de un tipo de alimento y eliminarla del resto de la imagen, esto puede llegar a ser útil para deshacerse de información innecesaria.

3.2.3. Ecuación adaptativa del histograma

Las imágenes en blanco y negro pueden representarse como matrices de valores entre 0 y 255, donde 0 corresponde al negro y 255 al blanco. A partir de esta matriz se puede generar un histograma de la imagen, al cual es posible aplicar una ecuación adaptativa. Esta operación ajusta el histograma, expandiendo su distribución dentro del rango completo de 0 a 255 para mejorar el contraste, mientras preserva la información relevante de la imagen[38].

Como se observa en la Figura 4, el histograma está más concentrado en el lado izquierdo, lo que indica que la imagen es oscura. Esta distribución puede ser ecualizada para mejorar su visibilidad.

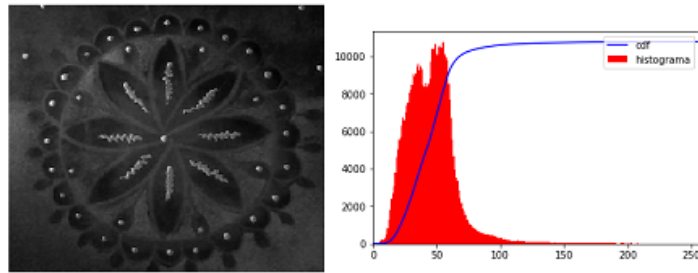


Figura 4: Imagen oscura antes de ecualizarse. Fuente: [55].

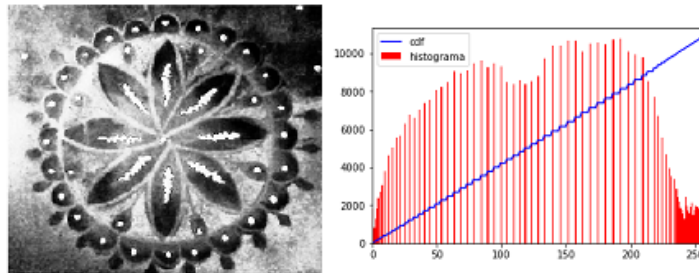


Figura 5: Imagen después de aplicarle ecualización adaptativa del histograma. Fuente: [55].

En la Figura 5, se puede observar cómo cambia el histograma, creando una distribución más uniforme de los valores de gris sin perder información. Es decir, los píxeles de un mismo nivel de gris se transforman a otros niveles de gris de manera que se maximiza su separación, lo que contribuye a una mejor representación visual[11].

3.2.4. Descomposición en planos de bits

La descomposición en planos de bits es un proceso que consiste en descomponer una imagen en planos de bits las cuales corresponden a cada posición de los valores de los píxeles. Es decir, que el primer plano de bits va a corresponder a los bits menos significativos de cada píxel de la imagen, el segundo plano consiste en el segundo bit menos significativo de cada píxel y así sucesivamente [44].

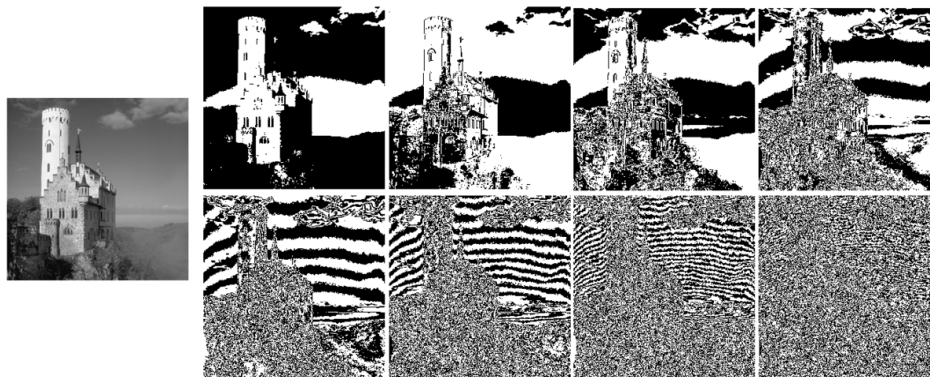


Figura 6: Planos de bits de una imagen. Fuente: [12].

Como se observa en la Figura 6 la imagen de abajo a la derecha representa el plano de bit 1 el cual

contiene el bit menos significativo. Es evidente cómo entre más significativo sea el bit, más información le aporta a la imagen, lo que hace que se vea más clara y similar a la original.

3.2.5. Corrección gamma

La corrección gamma es una operación no lineal utilizada para modificar la luminancia² de una imagen, haciéndola más oscura o más brillante como se puede evidenciar en la Figura 7. La fórmula utilizada en su expresión más simple es la siguiente ley de potencias:

$$V_{\text{out}} = A \cdot V_{\text{in}}^{\gamma}$$

donde A es una constante y V_{in} es el valor de entrada. Cuando el gamma es mayor a 1, se denomina gamma de decodificación, y la aplicación de la ley de potencias se llama expansión gamma, lo que hace que la imagen se vea más oscura. En cambio, cuando gamma es menor a 1, se llama gamma de codificación, y la aplicación de la ley de potencias se llama compresión gamma, lo que hace que la imagen sea más brillante [10].



Figura 7: Efecto de la corrección gamma en una imagen. Fuente: [10].

²La luminancia es una medida de la intensidad de luz percibida emitida o reflejada por una superficie en una dirección específica, que determina la claridad o el brillo que observamos en una imagen.

3.2.6. Operaciones morfológicas

Las operaciones morfológicas son un amplio conjunto de operaciones las cuales ayudan a procesar imágenes haciendo que cada píxel de esta se ajuste en función de los valores de otros píxeles cercanos[34]. Estas operaciones resultan útiles a la hora de simplificar características de las imágenes y sacar formas o patrones que resulten de interés. Las operaciones más utilizadas en estos contextos son la dilatación, erosión, cierre, apertura y esqueletización.

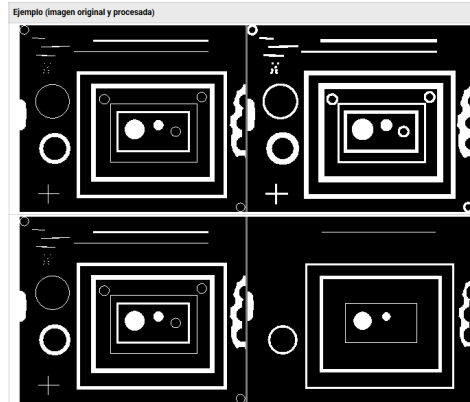


Figura 8: Operaciones de dilatación y erosión en una imagen. Fuente: [35].

La dilatación sirve para engrosar los límites de los objetos de forma que los objetos se vean mas grandes, esto se logra definiendo el valor de un píxel como el valor máximo de todos los píxeles de su entorno. La erosión hace lo contrario, define los píxeles como el valor mínimo de su entorno lo que genera que se eliminen píxeles flotantes y solo queden las figuras mas predominantes[35].

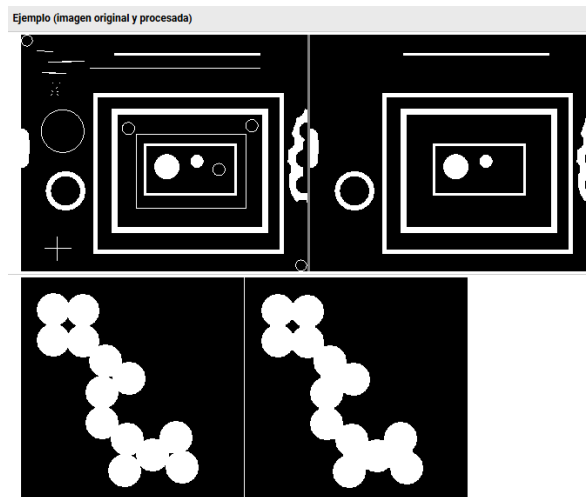


Figura 9: Operaciones de apertura y cierre respectivamente en una imagen. Fuente: [35].

Las operaciones de cierre y apertura son combinaciones de las anteriores operaciones. La apertura consiste en aplicarle erosión y luego dilatación a una imagen lo que sirve para eliminar objetos pequeños y líneas estrechas pero que a la vez conservan el tamaño de los objetos más grandes de la imagen original. El cierre en contraste, aplica dilatación y luego erosión lo que permite rellenar huecos en los objetos y una líneas cercanas que no estuvieran conectadas, conservando la forma de los objetos originales.[4].

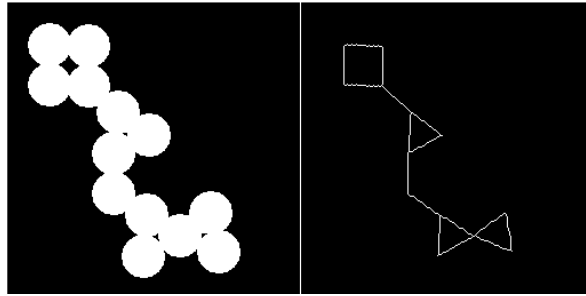


Figura 10: Operación esqueletización en una imagen. Fuente: [35].

Finalmente la esqueletización erosiona todos los objetos hasta obtener líneas centrales que conservan la estructura de los objetos, teniendo en cuenta también los huecos o ramas de dicho objeto.

3.3. Visión por computadora

La visión por computadora, también conocida como computer vision (CV) en inglés, es una rama de la inteligencia artificial que se centra en capacitar a las máquinas para interpretar y comprender el contenido visual del mundo que las rodea, de manera similar a cómo lo hacen los seres humanos. Utilizando algoritmos y técnicas de procesamiento de imágenes, la visión por computadora permite a las computadoras analizar y extraer información significativa de imágenes y videos, como identificar objetos, reconocer patrones, entender el movimiento y comprender el contexto visual de imágenes o videos que se le presenten.

Esta tecnología se utiliza hoy en día en una gran variedad de aplicaciones del mundo real, como por ejemplo el reconocimiento óptico de caracteres, la inspección rápida de partes mecánicas, en la logística de almacenes e imágenes médicas, como el registro de imágenes preoperatorias e intraoperatorias [56].

Este enfoque de análisis por medio de visión computadora tiene ciertas ventajas comparadas con la interpretación tradicional, puesto que puede abarcar más información de la que la haría un humano, incluso un profesional, y en menos tiempo, elimina la necesidad de comprensión humana para tareas repetitivas y puede reducir costos a largo plazo en comparación de contratar a múltiples especialistas. Sin embargo, la visión por computadora no busca reemplazar a los profesionales, sino funcionar como herramienta para mejorar la eficiencia de distintas tareas. Algunos ejemplos en este contexto de imágenes médicas son *Echocardiographic Image Segmentation for Diagnosing Fetal Cardiac Rhabdomyoma During Pregnancy Using Deep Learning*[53] el cual tiene como objetivo crear una interpretación automatizada de ecocardiogramas, lo cual puede resultar útil para áreas con falta de expertos o recursos limitados, *Computer-Aided Diagnosis with Deep Learning Architecture: Applications to Breast Lesions in US Images and Pulmonary Nodules in CT Scans*[6] trata sobre el diagnóstico diferencial de lesiones mamarias malignas o benignas en imágenes de ultrasonido y *Computer-aided diagnosis of human brain tumor through MRI: A survey and a new algorithm*[13] trata sobre la clasificación de tumores cerebrales en las categorías de benigno y maligno por medio de extracción de características y segmentación.

3.4. Teoría del Aprendizaje de Máquina

3.4.1. Métodos de aprendizaje supervisado, no supervisado y de refuerzo

El aprendizaje supervisado se basa en un conjunto de datos de entrenamiento que consta de pares de entradas y salidas conocidas, denominadas etiquetas[23]. El algoritmo de aprendizaje supervisado aprende a mapear estas entradas a sus correspondientes salidas deseadas. El objetivo es que, una vez entrenado, el modelo pueda tomar nuevas entradas y predecir correctamente sus salidas. Algunos ejemplos de algoritmos de aprendizaje supervisado son la regresión lineal, los árboles de decisión y

las redes neuronales. Estas técnicas se utilizan comúnmente para resolver problemas de clasificación, donde se asigna una categoría o clase a una entrada, y de regresión, donde se predice un valor numérico de salida[23]. Para este proyecto se utilizó este enfoque, ya que se buscan clasificaciones precisas para conjuntos de datos etiquetados y de los cuales ya se conocen sus clases, además de que se usaron redes neuronales, las cuales son comúnmente utilizadas para este método.

A diferencia del aprendizaje supervisado, en el aprendizaje no supervisado no se cuenta con datos etiquetados. El objetivo en este caso es que el algoritmo sea capaz de encontrar patrones y estructuras inherentes en los datos, sin tener conocimiento previo de las salidas deseadas. Algunos ejemplos de algoritmos de aprendizaje no supervisado son los métodos de agrupamiento (clustering), como K-means y DBSCAN, que permiten agrupar los datos en función de sus similitudes[37], que ayudan a visualizar y explorar grandes volúmenes de datos. El aprendizaje no supervisado se utiliza para descubrir nuevas perspectivas y patrones ocultos en los datos, en tareas como segmentación de clientes, detección de anomalías y compresión de datos.

Por otro lado, el aprendizaje por refuerzo se basa en la interacción del algoritmo con un entorno dinámico. El algoritmo toma acciones en este entorno y recibe recompensas o castigos en función de los resultados de sus acciones[37]. El objetivo es que el algoritmo aprenda a maximizar la recompensa a largo plazo, adaptando su comportamiento a través de la experiencia. A diferencia de los enfoques supervisados y no supervisados, en el aprendizaje por refuerzo no se requiere un conjunto de datos de entrenamiento etiquetado[37]. En su lugar, se define una función de recompensa que guía el aprendizaje. Esta técnica permite que los agentes aprendan de forma autónoma, sin depender de supervisión humana.

3.4.2. Preprocesamiento de imágenes

Es una etapa fundamental en la preparación de las imágenes antes de ser utilizadas por los modelos de aprendizaje de máquina o aprendizaje profundo para la tarea de clasificación. Este proceso ayuda a mejorar la eficiencia del modelo y su capacidad de generalización. A continuación, se describen las principales técnicas de preprocesamiento utilizadas:

- **Normalización o Estandarización:** Las imágenes están representadas por valores de píxeles en el rango de 0 a 255. Para facilitar el entrenamiento del modelo, es común normalizar estos valores dividiéndolos entre 255, ajustando los píxeles al rango $[0, 1]$. Alternativamente, se puede estandarizar restando la media y dividiendo entre la desviación estándar del conjunto de datos, lo que permite que las imágenes tengan valores centrados en torno a 0.
- **Redimensionamiento:** Las CNN requieren un tamaño de entrada fijo para las imágenes. Por tanto, es necesario redimensionar todas las imágenes a un tamaño predeterminado (por ejemplo, 224x224 píxeles) para garantizar su compatibilidad con la arquitectura del modelo, especialmente si se utilizan modelos preentrenados.
- **Corrección de colores y contraste:** Algunas imágenes pueden tener problemas de iluminación o un desequilibrio en los colores. Aplicar ajustes de brillo, contraste y saturación puede ayudar a que el modelo sea menos sensible a variaciones irrelevantes y se concentre más en las características importantes para la clasificación.
- **Eliminación de ruido:** Dependiendo de la calidad del conjunto de datos, algunas imágenes pueden contener ruido o artefactos indeseados. El uso de filtros o técnicas de suavizado puede reducir este ruido, permitiendo que el modelo se enfoque en las características relevantes de las imágenes.

3.4.3. Aumento de datos

Es una técnica que se utiliza para aumentar la cantidad y diversidad de datos de entrenamiento al aplicar transformaciones a los datos existentes. Esto es especialmente útil en el caso de datasets

pequeños o limitados, ya que permite mejorar la generalización del modelo sin necesidad de recolectar más datos. Los métodos considerados se muestran con una descripción concisa en el Cuadro 1.

Methods	Description
Flipping	Flip the image horizontally, vertically, or both.
Rotation	Rotate the image at an angle.
Scaling Ratio	Increase or reduce the image size.
Noise injection	Add noise into the image.
Color space	Change the image color channels.
Contrast	Change the image contrast.
Sharpening	Modify the image sharpness.
Translation	Move the image horizontally, vertically, or both.
Cropping	Crop a sub-region of the image

Cuadro 1: Manipulaciones básicas de las imágenes con una pequeña descripción [62].

Sin embargo, existen inconvenientes al usar este tipo de técnicas. En primer lugar, tiene sentido aplicar manipulaciones básicas de la imagen solo bajo el supuesto de que los datos existentes obedecen a la distribución cercana a la distribución real de los datos. En segundo lugar, algunos métodos básicos de manipulación de imágenes, como la traslación y la rotación, sufren el efecto de relleno[62]. Es decir, tras la operación, algunas zonas de las imágenes se desplazarán fuera de los límites y se perderá esta información.

3.4.4. Aprendizaje Profundo

Es un método de la inteligencia artificial (IA) que entrena a las computadoras para procesar datos teniendo como inspiración el cerebro humano. Los modelos de aprendizaje profundo son capaces de reconocer patrones complejos en imágenes, textos, sonidos y otros datos, a fin de generar información y predicciones precisas. Es posible utilizar métodos de aprendizaje profundo para automatizar tareas que habitualmente requieren inteligencia humana, como la descripción de imágenes o la transcripción a texto de un archivo de sonido.

Redes neuronales convolucionales Las redes neuronales convolucionales (CNN) son una arquitectura de aprendizaje profundo especialmente efectiva para el análisis y procesamiento de imágenes. A diferencia de las redes neuronales tradicionales, donde cada neurona está conectada a todas las neuronas de la capa anterior, en una CNN las neuronas solo se conectan a un pequeño subconjunto de neuronas de la capa anterior. Esto permite que la red capture de manera más eficiente las relaciones espaciales y los patrones locales presentes en las imágenes. La arquitectura de una CNN típicamente consta de varias capas convolucionales que aplican un conjunto de filtros a la entrada, seguidas por capas de agrupamiento que reducen la dimensionalidad de los mapas de características, y finalmente capas densamente conectadas que combinan estas características para generar las salidas finales[37].

Un aspecto clave de las CNN es su capacidad para aprender características relevantes de forma automática a partir de los datos de entrenamiento, sin necesidad de definir manualmente características específicas. Esto las convierte en herramientas poderosas para una amplia gama de aplicaciones, como reconocimiento de objetos, detección de rostros y segmentación de imágenes médicas, entre otras. Además, las CNN han demostrado ser altamente escalables y capaces de manejar grandes volúmenes

de datos, lo que las hace particularmente adecuadas para problemas que involucran el procesamiento de imágenes de alta resolución o conjuntos de datos de gran tamaño[32].

Redes convolucionales densas (DenseNet) Las redes convolucionales densas, también conocidas como DenseNet, son una arquitectura de redes neuronales diseñada para abordar problemas comunes en las redes convolucionales profundas, como la desaparición de gradientes (pérdida de información a medida que se transmite por las capas) y la redundancia en la extracción de características (capas que aprenden lo mismo que ya se ha descubierto en capas anteriores). A diferencia de las arquitecturas tradicionales, donde la información fluye de manera secuencial entre capas, DenseNet establece conexiones directas entre todas las capas que comparten el mismo tamaño de mapa de características (es decir, todas las capas dentro de un mismo bloque denso). Esta conectividad densa permite que cada capa reciba como entrada tanto el mapa de características de todas las capas anteriores como la salida original.

Una de las principales ventajas de esta arquitectura es la mejora en la propagación de características, dado que las capas posteriores tienen acceso directo a las características aprendidas por las capas anteriores, lo que reduce la pérdida de información durante el entrenamiento. Asimismo, fomenta la reutilización de características, es decir, si una capa inicial detecta una característica relevante, las capas subsiguientes pueden reutilizar esta información en lugar de aprenderla de nuevo. Este enfoque no solo optimiza el proceso de entrenamiento, sino que también mejora la eficiencia del modelo al reducir el número de parámetros necesarios para representar las características, lo que puede llevar a una mejor generalización[27].

Las arquitecturas DenseNet se componen de una capa inicial de convolución, una serie de bloques densos, capas de transición entre estos bloques y una capa de clasificación al final. Los bloques densos son las unidades fundamentales que permiten esta densa conectividad entre las capas.

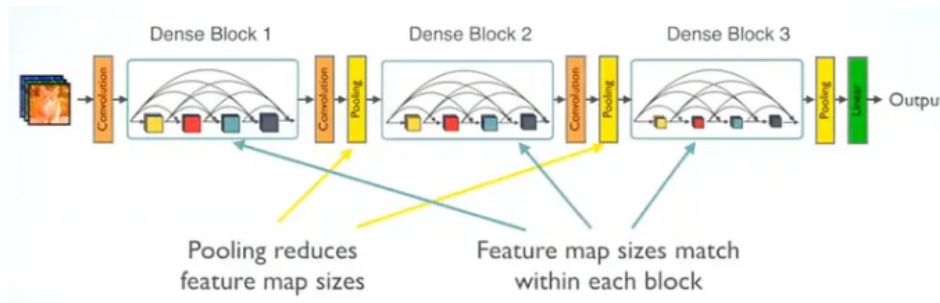


Figura 11: Arquitectura interna DenseNet.[58].

Como se muestra en la Figura 11, cada bloque denso está compuesto por múltiples capas de convolución. La primera capa recibe la entrada original, mientras que las capas siguientes reciben tanto la entrada original como la salida de todas las capas anteriores. Esto asegura que la información y las características se preserven a medida que se propagan a través de la red, lo que es crucial para tareas complejas de reconocimiento de patrones.

Las capas de transición, que se encuentran entre los bloques densos, consisten en capas de convolución y pooling que reducen la dimensionalidad de los mapas de características. Esta reducción no solo minimiza la complejidad computacional, sino que también ajusta las dimensiones de la entrada para que puedan ser procesadas en el siguiente bloque denso. Finalmente, la capa de clasificación aplica una operación de pooling global para obtener una salida de tamaño 1×1 , que luego se alimenta a una capa completamente conectada, la cual genera la predicción final.

DenseNet ha demostrado ser particularmente efectiva en tareas de clasificación de imágenes y ha sido ampliamente utilizada como base para modelos en dominios como la visión por computadora y la biomedicina debido a su capacidad para extraer y reutilizar características a lo largo de la red.

Redes neuronales residuales (ResNets) Las redes residuales, o ResNets por su traducción en inglés *Residual Networks*, son un tipo de arquitectura de red neuronal profunda. La principal característica de las ResNets es la introducción del aprendizaje residual, que aborda el problema de la degradación y la desaparición de gradientes en las redes neuronales profundas. “En lugar de esperar que cada una de las capas apiladas se ajuste directamente a un mapeo subyacente deseado, permitimos explícitamente que estas capas se ajusten a un mapeo residual” [25].

Las redes profundas tradicionales intentan aprender una correspondencia directa entre la entrada y la salida. En cambio, las ResNets aprenden el residuo, es decir, la diferencia entre la entrada y la salida deseada. Esto se consigue mediante el uso de “conexiones de salto” que se utilizan para añadir la entrada de una capa a la salida de una capa más profunda. De este modo, la red aprende más fácilmente un mapa de identidad y se asegura de que las capas más profundas funcionen al menos tan bien como las menos profundas, evitando así el problema de la degradación.

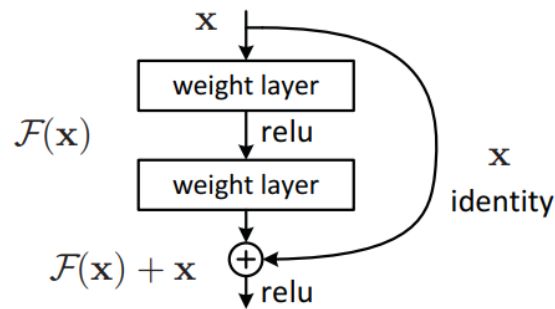


Figura 12: Bloque de construcción del aprendizaje residual [25].

El problema de la degradación hace referencia a la observación de que, a medida que aumenta la profundidad de la red, la precisión se satura y luego se degrada. Esto ocurre cuando los gradientes son demasiado pequeños para que las actualizaciones de pesos sean eficaces [25].

Detección de objetos: YOLO De sus siglas en inglés *You Only Look Once*, es una técnica de detección de objetos en tiempo real que revolucionó la visión por computadora al ofrecer una forma rápida y precisa de identificar y localizar objetos en imágenes y videos. A diferencia de otros métodos de detección de objetos que suelen abordar la detección como un problema de clasificación seguido de una localización, YOLO trata la detección de objetos como un problema de regresión a cuadros delimitadores separados espacialmente y los asocia a una clase de probabilidades[46]. El modelo predice directamente las coordenadas de los recuadros delimitadores (bounding boxes) y las probabilidades de las clases a partir de imágenes completas en una sola evaluación.

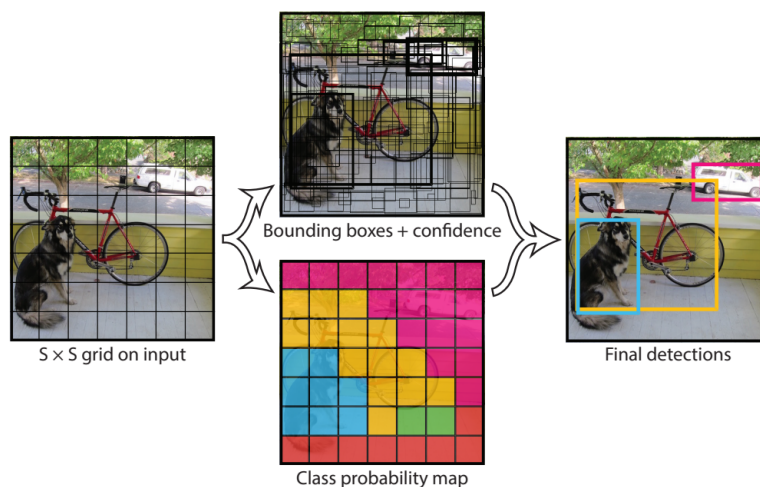


Figura 13: Se divide la imagen en una cuadrícula $S \times S$ y para cada celda de la cuadrícula predice B recuadros delimitadores, la confianza de esos recuadros y probabilidades de clase C . Estas predicciones se codifican como $S \times S \times (B \cdot 5 + C)$ [46].

3.4.5. Estrategias de entrenamiento y validación

El entrenamiento de modelos de aprendizaje de máquina implica ajustar los parámetros del modelo a fin de que pueda hacer predicciones precisas sobre nuevos datos. Esto se logra a través de un proceso iterativo donde el modelo aprende de un conjunto de datos de entrenamiento[37]. Existen diversas técnicas de entrenamiento, como el gradiente descendiente, que ajusta gradualmente los parámetros para minimizar una función de pérdida. Otro enfoque es el aprendizaje basado en instancias, donde el modelo memoriza los ejemplos de entrenamiento y utiliza métricas de similitud para hacer predicciones.

La validación es un paso crucial para evaluar el desempeño del modelo y evitar el sobreajuste (overfitting)[57]. Una técnica común es la validación cruzada, donde se divide el conjunto de datos en subconjuntos de entrenamiento y validación, y se evalúa el modelo en los datos de validación. Esto permite estimar el error de generalización del modelo, es decir, su capacidad de hacer predicciones precisas en datos nuevos y desconocidos. Otras estrategias incluyen el uso de conjuntos de prueba separados, la partición estratificada de los datos y técnicas de remuestreo, como el bootstrap[57].

3.4.6. Métricas de análisis de desempeño

Matriz de confusión Es una tabla que muestra, cuantitativamente, la clasificación correcta e incorrecta de los datos, comparando los valores reales con los predichos por el modelo. Está compuesta por el número de verdaderos positivos (TP), falsos negativos (FN), verdaderos negativos (TN), y falsos positivos (FP) para evaluar el rendimiento general [18], como se muestra en la Figura 14.

		Valor Real	
		Positivo	Negativo
Predicciones	Positivo	Verdaderos positivos	Falsos Positivos
	Negativo	Falsos Negativos	Verdaderos Negativos

Figura 14: Tabla de matriz de confusión.

Exactitud Proporción de predicciones correctas entre todas las predicciones realizadas [18].

$$Exactitud = \frac{\text{Predicciones Correctas}}{\text{Total de Predicciones}}$$

Precisión Mide la proporción de verdaderos positivos (TP) sobre todos los positivos predichos (TP + Falsos Positivos). La fracción de casos realmente positivos de todos los casos que el modelo predijo positivos [18].

$$Precision = \frac{TP}{TP + FP}$$

Exhaustividad (Recall) Mide la proporción de positivos correctamente identificados (TP) sobre todos los casos reales positivos (TP + Falsos Negativos). Indica la capacidad del modelo para capturar todas las instancias positivas [18].

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

F1-score Combina las métricas de precisión y recall en una sola puntuación mediante la media armónica [18]. Es útil cuando existe un desbalance entre clases.

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

Mean Average Precision (mAP50) Es una métrica utilizada principalmente en tareas de detección de objetos, donde se combina la precisión y el recall para evaluar la calidad de las predicciones del modelo. se fija en el umbral de Intersección sobre Unión (IoU) del 50% [31], lo que significa que una predicción es considerada correcta si la intersección entre la caja predicha y la caja real es al menos el 50% del área combinada.

mAP50-95 Más robusta que mAP50, donde se evalúa el rendimiento del modelo en un rango más amplio de umbrales de IoU, desde 0.5 hasta 0.95 [31], con incrementos de 0.05. Esta métrica calcula el valor de mAP en cada uno de estos umbrales y luego toma el promedio.

Perdida (Loss) Es una medida que compara las predicciones del modelo con los valores reales o esperados. Este es el fundamento que usa el modelo para aprender, ya que el objetivo del entrenamiento es minimizar dicha pérdida. Una pérdida más baja indica que las predicciones del modelo están más cerca de los valores reales. Las funciones de pérdida más comunes serían:

- **Error Cuadrático Medio:** Se usa principalmente en problemas de regresión. Calcula la media de los cuadrados de las diferencias entre las predicciones y los valores reales [60].
- **Entropía Cruzada (Cross-Entropy Loss):** Es común en problemas de clasificación. Mide la diferencia entre la distribución de probabilidad predicha y la real. Para la clasificación binaria o multiclase, la entropía cruzada penaliza más fuertemente las predicciones incorrectas [60].

4. Trabajos Relacionados

En esta sección se presentan diferentes investigaciones de referencia que abordan problemas similares a los de este trabajo y emplean modelos adecuados para la resolución de problemas relacionados. **Using Deep-Learning Algorithms to Classify Fetal Brain Ultrasound Images as Normal or Abnormal** [61]

Este artículo investiga la implementación de algoritmos de aprendizaje profundo para clasificar ecografías del cerebro fetal como normales o anormales, alcanzando una precisión del 97.9% en la segmentación de imágenes.

El estudio se relaciona con este proyecto, ya que demuestra aplicaciones reales de la clasificación y segmentación de ecografías craneales, estableciendo un punto de referencia en términos de precisión para el modelo que se desarrollará.

Automatic Deep Learning-Based Pipeline for Delineation and Measurement of Fetal Brain Structures in Routine Mid-Trimester Ultrasound Images [15]

En [15], se presenta el desarrollo de un pipeline basado en aprendizaje profundo que primero clasifica los planos de ecografías craneales, luego delinea estructuras del cráneo y finalmente realiza mediciones. Además, se abordan aspectos clave como el preprocesamiento de datos y las métricas utilizadas para evaluar el rendimiento del modelo en cada plano y área del cerebro fetal.

Este artículo es especialmente relevante para este proyecto, ya que aborda la clasificación de los tres planos cerebrales y emplea la misma base de datos [19].

Fetal Cortical Plate Segmentation Using Fully Convolutional Networks with Multiple Plane Aggregation [26]

En [26], se describe una red neuronal convolucional con múltiples planos diseñada para la segmentación de la placa cortical fetal. El estudio detalla el proceso de preprocesamiento de datos, la arquitectura de la red, las métricas utilizadas, la estrategia de entrenamiento y los resultados obtenidos, evaluados mediante métricas como precisión y recall en un plano fetal específico.

Este trabajo resulta relevante para el proyecto, ya que proporciona una guía estructurada para el desarrollo de un modelo de aprendizaje automático en el ámbito de las imágenes médicas, describiendo detalladamente cada una de las etapas involucradas en su construcción.

Review of Deep Learning and Artificial Intelligence Models in Fetal Brain Magnetic Resonance Imaging [49]

En [49], se presenta un análisis exhaustivo del estado del arte en la segmentación del cerebro fetal mediante resonancia magnética. Aunque este tipo de imagen difiere de las ecografías, los principios empleados en la clasificación son similares. Se utilizan métricas como el índice de Dice, y los estudios referenciados reportan valores que oscilan entre el 70% y el 98.23%, dependiendo de la cantidad de datos y la región cerebral que se desea clasificar.

Este artículo es relevante para el proyecto, ya que ofrece una visión general de las metodologías actuales para la clasificación y segmentación de imágenes médicas, así como de las métricas y estándares a los que se debe aspirar.

5. Metodología para la clasificación de planos cerebrales en ECP

En la Figura 15 se observa el pipeline propuesto para alcanzar el objetivo, que describe la secuencia seguida por las imágenes durante el desarrollo del proyecto:

- **Procesamiento digital de imágenes:** Las imágenes se someten inicialmente a un proceso de transformaciones y análisis digital que permite identificar si corresponden a una ecografía craneal prenatal (ECP).
- **Segmentación y almacenamiento:** Si las imágenes cumplen con los criterios para ser consideradas ECP según el procesamiento digital de imagen (PDI), se almacenan para el procesamiento. Nótese que en la Figura 15 solo se considera el caso en el que la imagen sea ECP puesto que se hace una segmentación, es decir, una discriminación de cuales imágenes son ECP mas no se afirma cuales no lo son. Además, es menester aclarar que para efectos de precisión de los modelos de aprendizaje automático se pasan solo las imágenes ECP clasificadas por los profesionales a la base de datos ECP para los pasos posteriores.
- **Preprocesamiento y aumento de datos:** Las imágenes seleccionadas son sometidas a técnicas de preprocesamiento y aumento de datos, lo que incluye ajustes en sus dimensiones y la generación de variaciones que incrementen la diversidad del conjunto de datos.
- **Creación de la base de datos aumentada:** Las imágenes preprocesadas y aumentadas se integran en una base de datos ampliada, diseñada específicamente para optimizar el entrenamiento de los modelos.
- **Entrenamiento de modelos:** La base de datos aumentada se utiliza para entrenar varios modelos de clasificación.

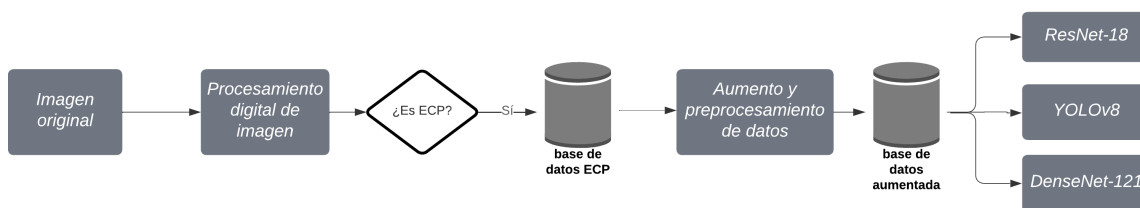


Figura 15: Secuencia del pipeline para la elaboración del proyecto

Luego de terminar el entrenamiento y ajustes se tendrá un pipeline para la clasificación de planos de ECP como el de la Figura 16 que tiene la siguiente secuencia:

- **Recepción de la imagen original:** Se recibe la imagen original a la cual se le aplica un procesamiento digital de imagen(PDI).
- **Identificación como ECP:** En caso de que la imagen corresponda a una ecografía craneal prenatal, pasa a los modelos de clasificación.
- **Clasificación por modelos:** Los modelos asignan un plano craneal a la imagen procesada.

- **Obtención de clasificación final:** Con pesos asignados manualmente a los modelos, se obtiene la clasificación final del plano craneal.

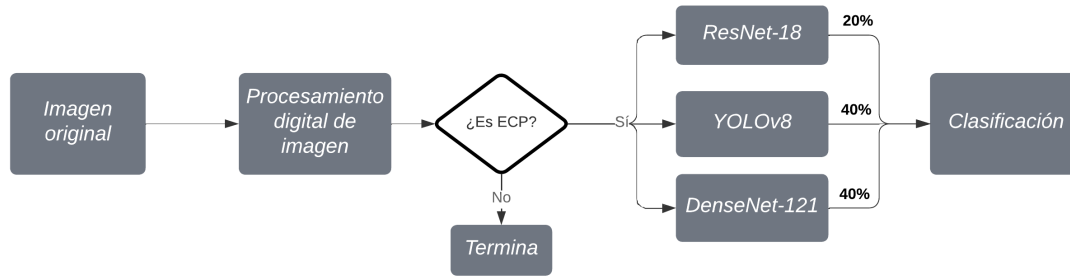


Figura 16: Secuencia del pipeline para la clasificación de ECP

5.1. Selección de la base de datos

Se seleccionó una base de datos abierta disponible en la plataforma Zenodo [19], que consta de 12.400 imágenes obtenidas de 1.792 pacientes de dos hospitales, junto con un documento en formato Excel que registra detalles como el nombre de las imágenes, el número del paciente, el plano de la imagen, el operador, la máquina utilizada y una clasificación sobre si la imagen era de entrenamiento o no (característica que se omitió en este estudio). Todas las imágenes fueron clasificadas manualmente por expertos en la materia.

Esta base de datos fue elegida por ser pública y por ser en el momento de su publicación, la más grande disponible en el campo de los ultrasonidos fetales, destacándose además por ser una de las pocas, si no la única, que clasifica los distintos planos cerebrales de forma específica, lo cual es necesario para poder construir una herramienta de clasificación. Además, esta base de datos se creó originalmente con el propósito de ser usada para entrenar modelos de clasificación y/o segmentación, lo que se adapta a los objetivos de la investigación, puesto que permite plantear una estrategia de aprendizaje supervisado para clasificar los planos de las ECP.

Las imágenes cubren gestaciones entre las semanas 18 y 40, lo que representa el periodo óptimo para la observación detallada de las estructuras cerebrales del feto. Estas se organizan en seis categorías: abdomen, cerebro, fémur, tórax, cuello uterino de la madre y una categoría adicional para planos menos comunes. En el caso de las imágenes del cerebro fetal, se dividen en los tres principales planos: transtalámico, transcerebeloso y transventricular.

5.2. Preprocesamiento y segmentación de ECP y no ECP

Para la segmentación de imágenes ECP y no ECP, se empleó un enfoque centrado en la detección y mapeo de los bordes del cráneo en la imagen, dividiéndolos en borde superior e inferior. Este proceso permite realizar una comparación con curvas de cráneos previamente registradas. Al evaluar estas curvas, podemos obtener un porcentaje de similitud que sirve como criterio para la clasificación; si la similitud supera el 80 %, se puede afirmar con un grado razonable de certeza que la imagen corresponde a un cráneo. En caso contrario, se concluye que no lo es.

Para lograr ese objetivo se tuvo en cuenta un total de 7 pasos como se muestra en la Figura 17. El pipeline inicia con el procesamiento de la imagen y su máscara, donde se aplican técnicas de preprocesamiento para eliminar regiones de no interés y normalizar los datos. Posteriormente, se realizan detalles mediante ajustes de contraste, suavizado y corrección gamma para mejorar la calidad visual. La imagen se convierte a un plano de bits, destacando los bordes relevantes que necesitamos, y luego se aplican operaciones morfológicas como dilatación y erosión para refinar las formas y eliminar pequeñas irregularidades. En las etapas siguientes, se reduce el ruido, se eliminan ramas innecesarias, se esqueletiza la imagen y se realiza una interpolación de los segmentos del cráneo, reconstruyendo los bordes esenciales. Finalmente, el sistema clasifica la imagen mediante un criterio booleano (TRUE/FALSE) basado en el porcentaje de similitud de los bordes conseguidos a las curvas de los cráneos ya registrados.

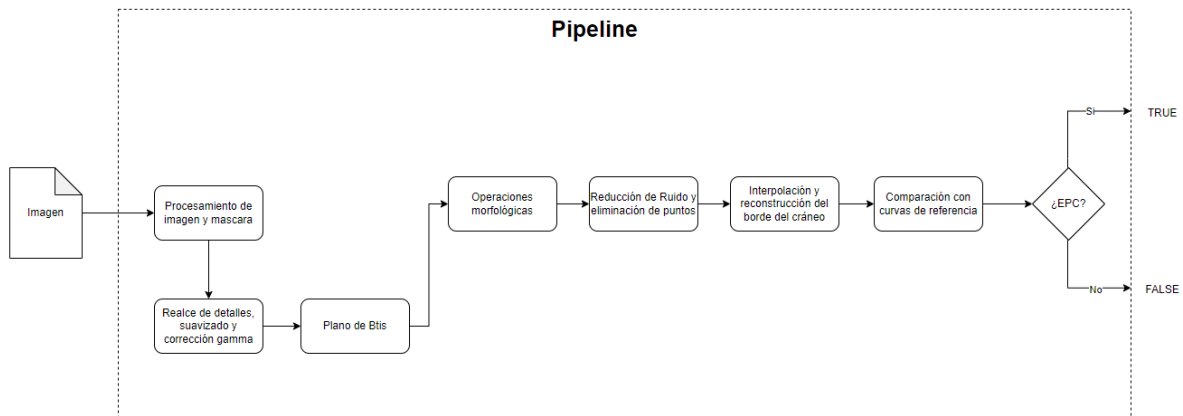


Figura 17: Pipeline del algoritmo de segmentación

5.2.1. Procesamiento de imagen y máscara

En primera instancia, se convierte la imagen original a escala de grises; esta conversión se hace para permitir trabajar con una única matriz de intensidades, simplificando el análisis posterior y reduciendo la complejidad computacional del proceso.

Posteriormente, se implementa una etapa de recorte de la imagen para eliminar posibles etiquetas o marcadores presentes en los márgenes laterales de la ecográfica. Este proceso se realiza mediante la extracción de una región de interés (ROI) definida por un margen ajustable. La imagen se recorta simétricamente desde ambos lados, manteniendo la altura original, pero reduciendo el ancho según el margen especificado.

Luego, se hace la creación de una máscara triangular para aislar el área de interés. Esta máscara se construye mediante la definición de dos triángulos: uno en el lado izquierdo y otro en el derecho de la imagen. Los vértices de estos triángulos se calculan en función de las dimensiones de la imagen, utilizando proporciones específicas para asegurar una cobertura adecuada del área craneal. La máscara se implementa utilizando operaciones de dibujo de polígonos convexos en OpenCV, creando una región blanca (valor 255) en las áreas de interés sobre un fondo negro.

Finalmente, la función aplica la máscara triangular a la imagen recortada mediante una operación de AND bit a bit. Esta operación preserva únicamente los píxeles de la imagen original que corresponden a las regiones definidas por la máscara triangular, efectivamente aislando el área de interés y eliminando las regiones periféricas que no son relevantes para el análisis. La función retorna tanto la imagen procesada como la máscara triangular utilizada. En la Figura 18 se muestra un ejemplo del procesamiento de la imagen y su resultado.

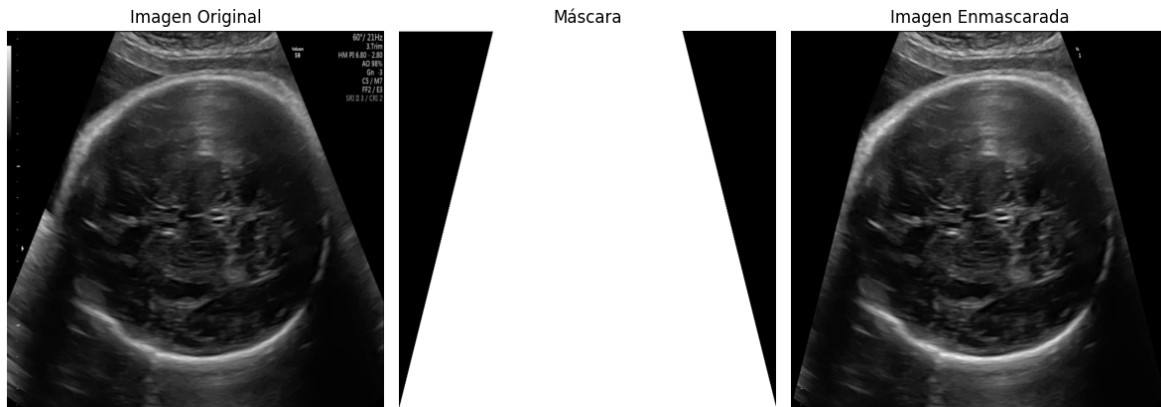


Figura 18: Ejemplo del procesamiento de imagen y su máscara

5.2.2. Realzo de detalles, suavizado y corrección gamma

Después de haber procesado la imagen y su máscara, se comienza el proceso de realce de detalles con la salida del paso anterior; primero se realiza una aplicación de una operación de enmascaramiento inicial, que permite aislar la región de interés definida por la máscara. Para así asegurar que las subsiguientes transformaciones se apliquen exclusivamente en las áreas relevantes de la imagen, preservando la integridad de las regiones circundantes.

Un componente central es el uso del algoritmo CLAHE (Contrast Limited Adaptive Histogram Equalization), una técnica avanzada de equalización de histograma que opera de manera adaptativa sobre regiones locales de la imagen. La configuración del CLAHE se realiza mediante `cv2.createCLAHE` con parámetros ajustados: un límite de recorte (`clipLimit`) de 2.0 y un tamaño de cuadrícula (`tileGrid-Size`) de 5x5 píxeles. Estos parámetros han sido seleccionados después de haber realizado diferentes pruebas con valores distintos para estos dos parámetros y así poder llegar a un balance entre el realce de contraste y la supresión de ruido.

Posterior a la aplicación del CLAHE, se implementa un filtro gaussiano con un kernel de 5x5 píxeles para suavizar cualquier artefacto o ruido introducido por el proceso de equalización, tratando de mantener la nitidez de los bordes de la estructura en la imagen. El uso de un kernel 5x5 en el filtro gaussiano proporciona un equilibrio entre suavizado y preservación de bordes. Un kernel de este tamaño es lo suficientemente grande como para reducir el ruido y los artefactos introducidos por el CLAHE, pero no tan grande como para difuminar en exceso los bordes de las estructuras en la imagen. Adicionalmente, un tamaño de 5x5 permite una mayor suavización que un 3x3, sin perder demasiada información estructural, y es computacionalmente más eficiente que un 7x7 o superior.

Adicionalmente, se incorpora una corrección gamma, una técnica de procesamiento no lineal que permite ajustar la luminancia de la imagen. Se implementa utilizando un valor gamma de 1.7, puesto que este valor es donde se muestran los mejores resultados para la clasificación de los bordes craneales, como se muestra en la Figura 19. Esta corrección se realiza mediante la creación de una tabla de búsqueda (Look-Up Table, LUT) que mapea los valores de intensidad de entrada a sus correspondientes valores corregidos, siguiendo la función potencial característica de la corrección gamma.

Finalmente, se aplica una última operación de enmascaramiento para asegurar que el resultado final mantenga la delimitación espacial definida por la máscara original. Esta operación garantiza la coherencia espacial del procesamiento y evita la introducción de artefactos en las regiones fuera del área de interés.

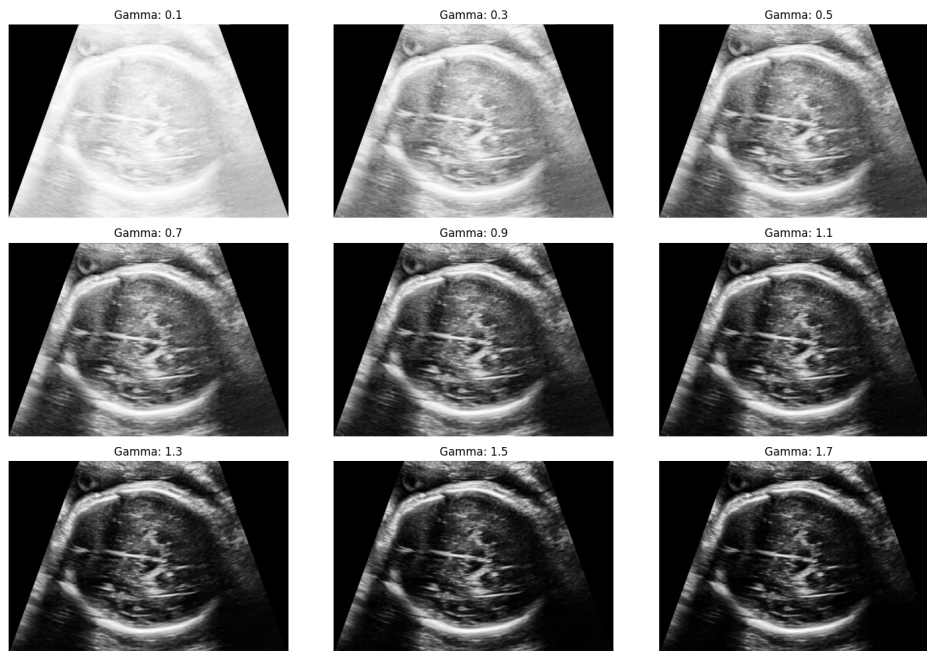


Figura 19: Diferente valores gamma para la imagen después de incorporar CLAHE y el filtro Gaussiano

5.2.3. Plano de bits

Los planos de bits descomponen una imagen en sus niveles binarios, separando la información en diferentes planos según su peso en la representación binaria de los píxeles. El plano de bits más significativo tiende a conservar las estructuras más destacadas de la imagen, eliminando detalles menores y ruido asociados con planos de menor importancia. Teniendo en cuenta lo anterior, este paso tiene como objetivo resaltar los detalles más significativos de la imagen segmentada, centrándose en las características relevantes para la detección de los bordes del cráneo. En otras palabras, los planos de bits nos sirven como un filtro para aislar las características esenciales que guían la segmentación precisa de los bordes del cráneo. En la Figura 20 podemos observar que el plano del bit más significativo (Bit Plane 7) contiene la información más relevante para poder realizar esta segmentación. Ergo, es el plano que se utilizó en este paso respecto a la imagen procesada en los pasos anteriores.

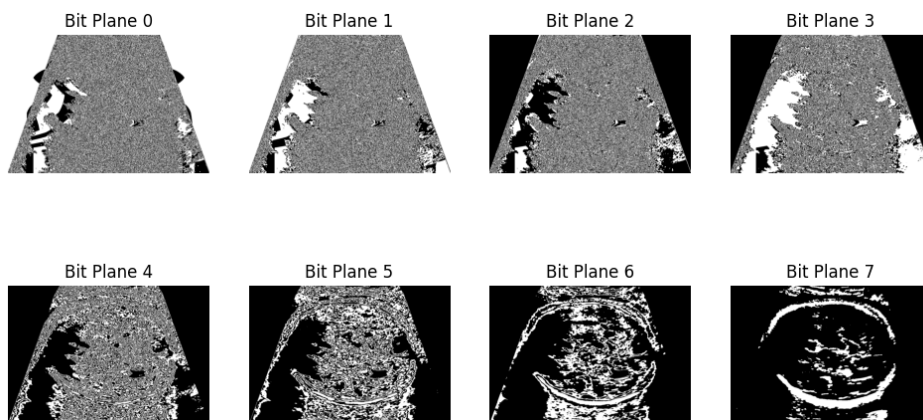


Figura 20: Diferentes planos de bits para la imagen procesada

5.2.4. Operaciones morfológicas

El proceso inicia con la definición de un elemento estructurante mediante la función 'disk' con un radio de 4 píxeles. Este elemento estructurante circular es fundamental para todas las operaciones morfológicas subsecuentes, ya que define la forma y el tamaño de la ventana de procesamiento que se utilizará en cada transformación.

La primera operación morfológica aplicada es la apertura (opening), que consiste en una erosión seguida de una dilatación. Esta operación es especialmente efectiva para eliminar pequeños elementos espurios y suavizar los contornos de las estructuras principales, manteniendo la forma general de los objetos más grandes.

Posteriormente, se aplica una operación de cierre (closing), que realiza una dilatación seguida de una erosión. Esta operación es complementaria a la apertura y tiene como objetivo rellenar pequeños huecos y conectar componentes cercanos que pudieran haber sido separados durante el proceso de segmentación inicial. El cierre ayuda a consolidar las estructuras anatómicas de interés y proporciona una representación más coherente de las regiones segmentadas.

Finalmente, se aplica una operación de erosión (erosion) que reduce el tamaño de los objetos en la imagen binaria. Esta última erosión sirve para refinar los bordes de las estructuras y eliminar cualquier artefacto residual que pudiera haber persistido después de las operaciones anteriores. Al esqueletizar el resultado del conjunto de estas operaciones e implementarlas en el plano de bits resultante de los anteriores paso, obtenemos lo que se muestra en la Figura 21.

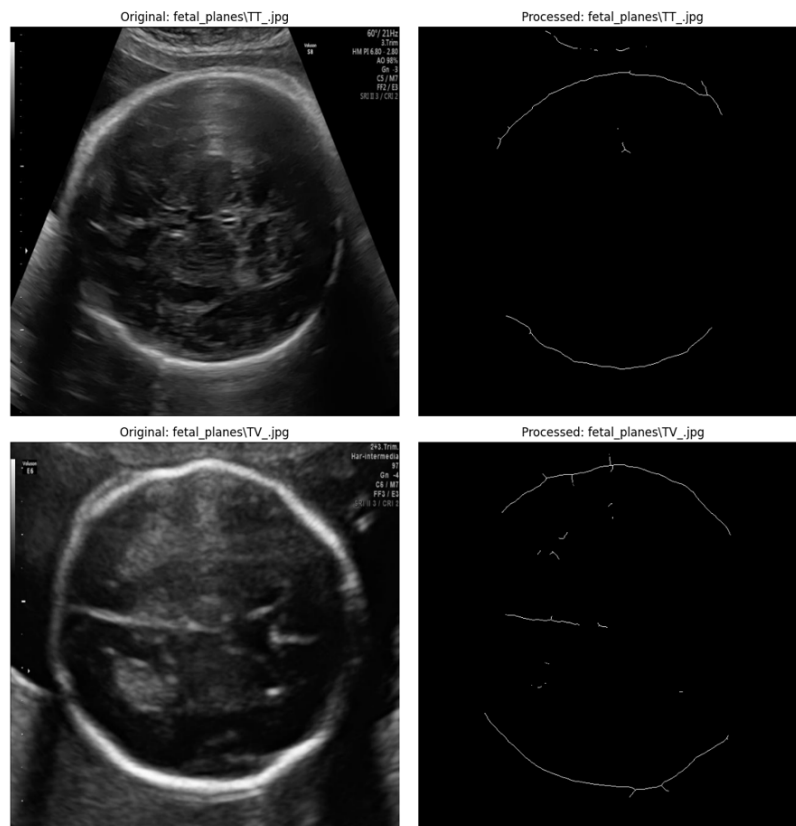


Figura 21: Operaciones morfológicas en diferentes planos cerebrales

5.2.5. Reducción de ruido y eliminación de ramas

Después de hacer las operaciones morfológicas vemos que en las imágenes resultantes siguen habiendo puntos o ramas (colección de puntos adyacentes) que no hacen parte del borde del cráneo, como se observa en la Figura 21 en las imágenes procesadas. Para poder mapear los bordes superiores e inferiores del cráneo tenemos que minimizar la cantidad de ruido que se muestra en la figura y obtener solo, o en lo mayor posible, la silueta de los bordes que necesitamos. Para esto se crearon dos funciones para obtener representaciones más limpias y precisas de la estructura; `clean_noise` y `remove_small_small_branches`.

La función `clean_noise` acepta dos parámetros: la imagen de entrada y un parámetro opcional que establece el tamaño mínimo de los objetos a preservar, con un valor de 100 píxeles. El proceso comienza con una binarización de la imagen mediante una operación de umbralización simple, donde todos los píxeles con valor mayor a cero se convierten en verdadero (1) y los demás en falso (0). Posteriormente, se aplica la función `remove_small_objects`, propia de la librería `skimage`, para eliminar componentes conectados cuya área sea menor que el umbral especificado, efectivamente removiendo partículas de ruido y artefactos pequeños. La función concluye con la eliminación de pequeños agujeros contiguos dentro de la región craneal mediante `remove_small_holes`, establecido por un umbral de área de 100 píxeles.

La función `remove_small_branches` está diseñada para el refinamiento de imágenes esqueletizadas mediante la eliminación de ramificaciones pequeñas mientras se preserva la estructura principal. Esta función acepta como parámetros la imagen del esqueleto y un parámetro opcional que define la longitud mínima de las ramas a preservar, con un valor predeterminado de 70 píxeles. La función implementa un algoritmo iterativo que identifica puntos finales mediante una operación de filtrado, donde los puntos con un valor de 11 en la imagen filtrada representan extremos de ramas. Para cada punto final identificado, el algoritmo realiza un seguimiento de la rama correspondiente mediante un proceso de crecimiento de región, manteniendo un registro de la longitud de la rama. Si la longitud de la rama es menor que el umbral especificado, esta se elimina de la imagen. Este proceso se repite hasta que no se encuentran más puntos finales o no se pueden realizar más modificaciones, asegurando una limpieza exhaustiva de las ramificaciones pequeñas mientras se mantiene la integridad de las estructuras principales.

Realizando una comparación entre la Figura 21 y la Figura 22 podemos observar que la última tiene una representación más limpia y precisa de los bordes del cráneo en las imágenes procesadas, donde se eliminan ramas y ruido innecesario tanto dentro de los bordes de cráneo, como fuera de él, aunque cabe recalcar que algunos elementos siguen siendo visibles que no son necesarios, si aumentamos en gran medida los parámetros de tamaños mínimos que pueden tener un objeto o una rama, corremos el riesgo de eliminar parte del borde del cráneo, por lo que los tamaños estipulados fueron lo que dieron mejores resultados en las pruebas de los algoritmos.

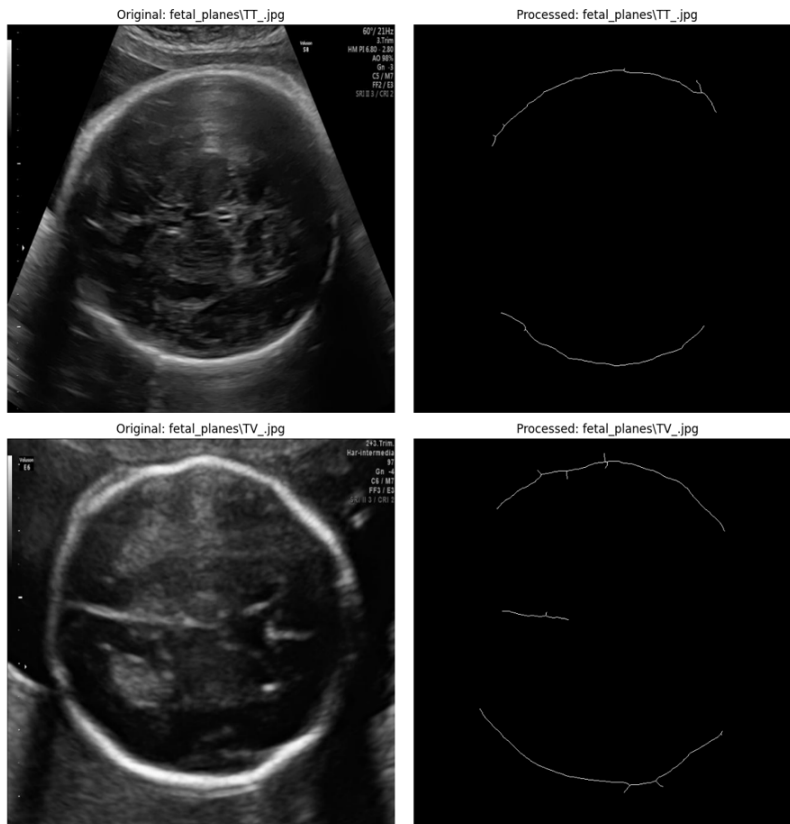


Figura 22: Ejemplo de reducción de ruido y eliminación de ramas

5.2.6. Interpolación de segmentos del cráneo

Obtener las coordenadas del borde superior e inferior del cráneo El proceso comienza con la determinación de las dimensiones de la imagen y el cálculo de la mitad vertical de la imagen. Esta referencia central es crucial para dividir la búsqueda en dos regiones: superior e inferior. La función utiliza un umbral del 20% para establecer límites de búsqueda específicos, calculando una zona de exclusión alrededor de la referencia central. Estos límites se calculan como una proporción de la altura total de la imagen, creando zonas de búsqueda específicas para los bordes superior e inferior del cráneo, como se puede observar en la Figura 23.

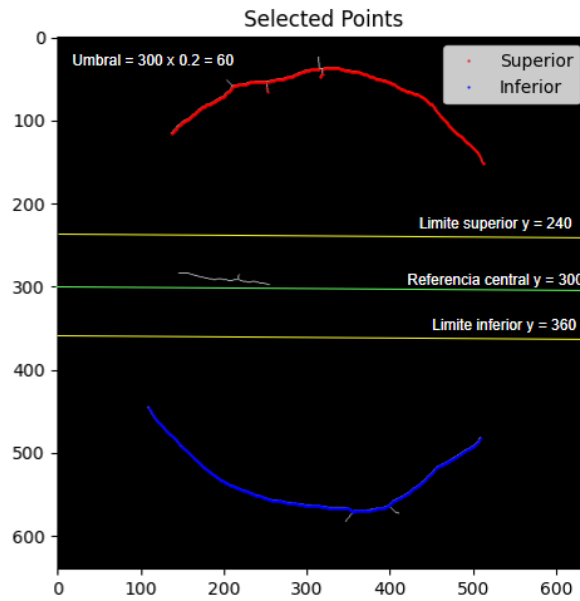


Figura 23: Limite superior e inferior respecto a un umbral del 20 %

El algoritmo implementa un proceso de escaneo columna por columna a través de la imagen. Para cada columna, se realizan dos búsquedas independientes: una ascendente desde la parte superior hasta el límite superior establecido por el umbral, y otra descendente desde la parte inferior hasta el límite inferior. En la región superior, el algoritmo busca el punto más bajo (mayor valor de fila) que tenga un valor de 255 (píxel blanco en la imagen binaria). De manera complementaria, en la región inferior, busca el punto más alto (menor valor de fila) que cumpla con la misma condición.

La función implementa una lógica de validación para asegurar que solo se almacenen coordenadas que cumplan con criterios específicos. Para el borde superior, se verifica que se haya encontrado un punto válido y que esté dentro del límite superior establecido. Para el borde inferior, se verifica que el punto encontrado sea diferente de la altura total de la imagen y esté dentro del límite inferior establecido. Estas coordenadas válidas se almacenan en dos listas separadas, que luego serán usadas como puntos para la creación de la función de interpolación para los bordes.

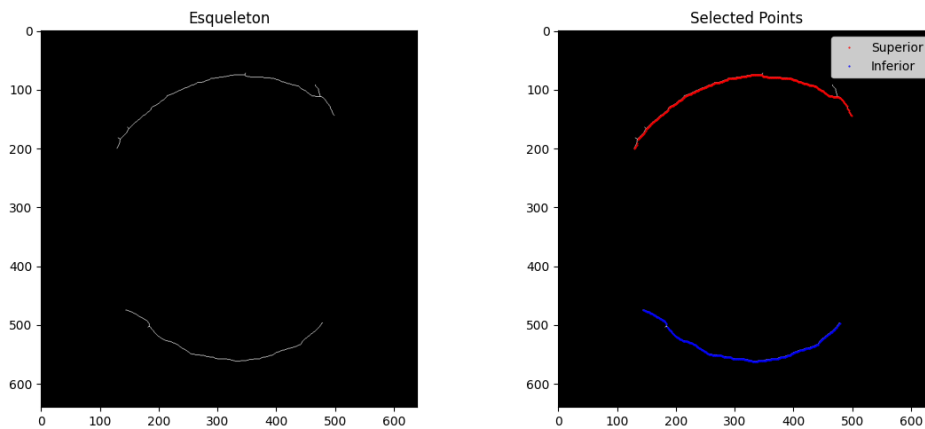


Figura 24: Ejemplo 1: Adquisición de coordenadas borde superior e inferior del cráneo

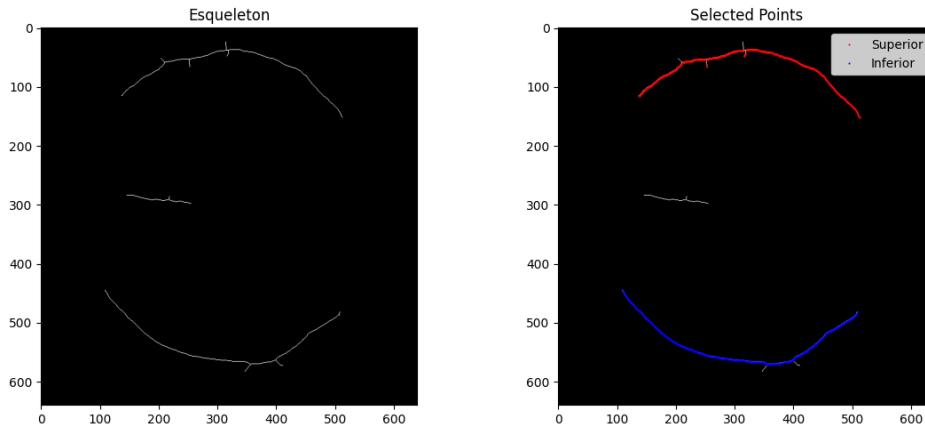


Figura 25: Ejemplo 2: Adquisición de coordenadas borde superior e inferior del cráneo

Interpolación polinomial Para suavizar y generar una tendencia de los bordes del cráneo se hizo una interpolación polinomial de orden 3, respecto a los puntos encontrados con el algoritmo de obtención de coordenadas de los bordes superiores e inferiores. De los puntos encontrados se seleccionaron 10 equidistantes, incluyendo los límites izquierdo y derecho del borde. Al hacer esto obtenemos una función para cada borde que tiene la siguiente forma:

$$P(x) = a_0 + a_1x + a_2x^2 + a_3x^3$$

Donde a_0 , a_1 , a_2 y a_3 son las constantes encontradas durante el proceso de interpolación y X los puntos detectados del borde del cráneo. Es importante mencionar que el ajuste de la interpolación se realizó probando polinomios de diferente orden y finalmente se estableció que el polinomio de orden 3 era el que mejor representaba las curvas del borde del cráneo. Algunos ejemplos se muestran en la Figura 26 y Figura 27.

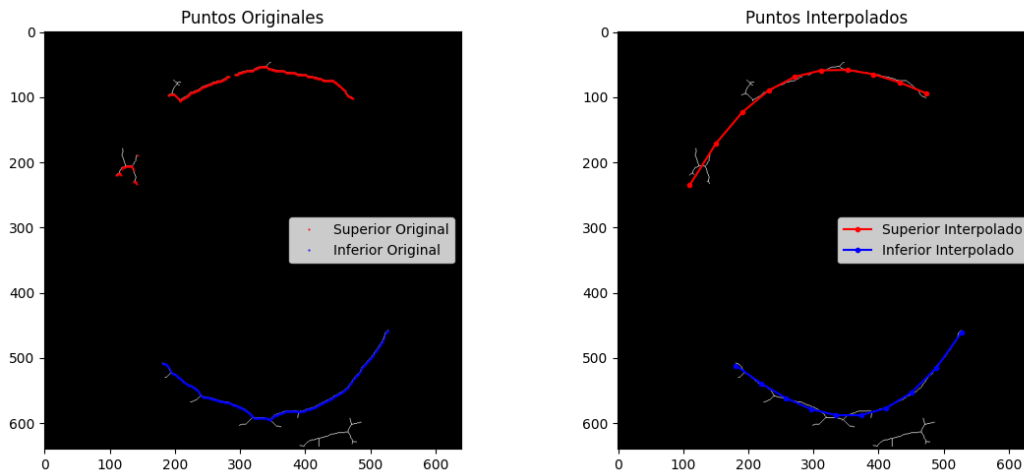


Figura 26: Ejemplo 1: Interpolación cúbica de imagen ecográfica

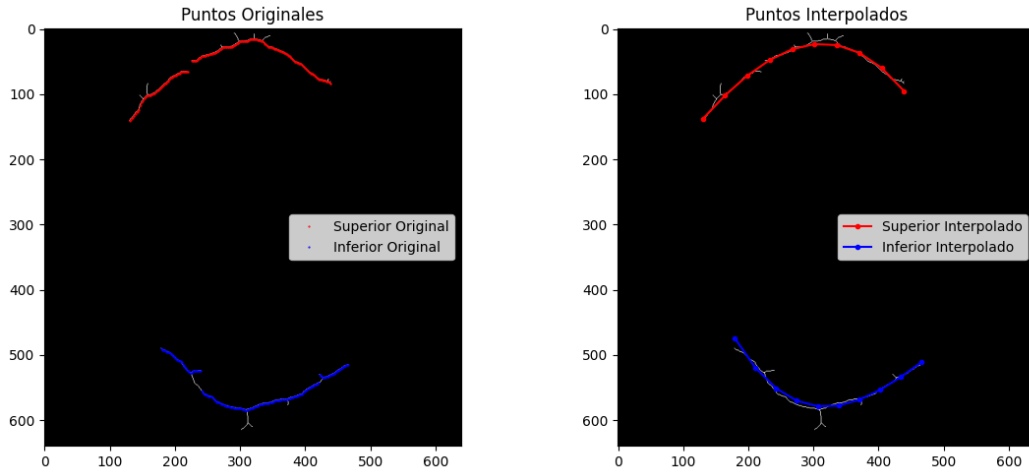


Figura 27: Ejemplo 2: Interpolación cúbica de imagen ecográfica

5.2.7. Comparación con curvas de referencia

Para realizar la comparación de la interpolación de los bordes superior e inferior de las imágenes ecográficas, se seleccionaron 20 imágenes adicionales que servirán como curvas de referencia. Estas curvas se distribuyen de la siguiente manera: 10 imágenes para el borde superior (Figura 28) y 10 imágenes para el borde inferior (Figura 29.) Una vez definidas estas curvas de referencia, se procede a mapearlas para prepararlas para el análisis comparativo.

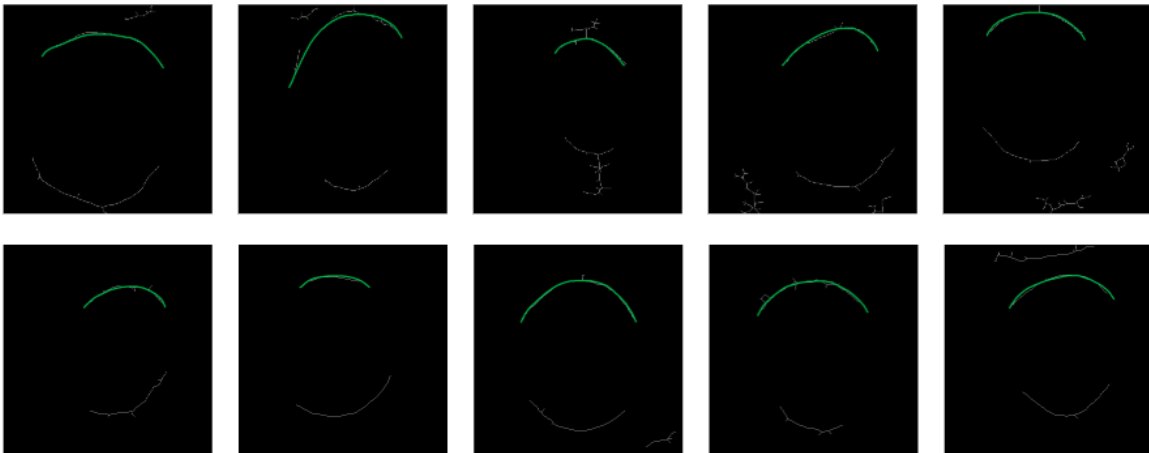


Figura 28: Imágenes de curvas de las 10 curvas de referencia para borde de cráneo superior

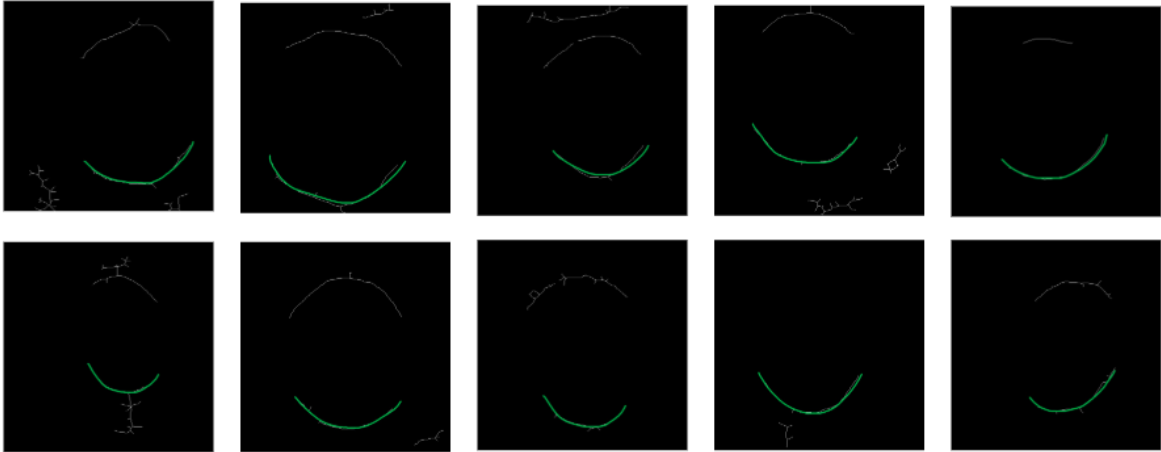


Figura 29: Imágenes de curvas de las 10 curvas de referencia para borde de cráneo superior

El siguiente paso consiste en realizar una interpolación de las curvas de referencia. Este proceso permite obtener una cantidad uniforme de puntos en cada curva, lo cual es indispensable para la comparación con las curvas obtenidas en los pasos previos. La interpolación garantiza que ambas curvas, tanto las de referencia como las obtenidas, tengan la misma cantidad de puntos, facilitando así una comparación más precisa.

Antes de proceder con la comparación, es necesario normalizar los puntos de las curvas. Esta normalización asegura que las curvas estén en el mismo rango y escala, eliminando posibles sesgos debido a diferencias en las dimensiones originales. Con las curvas ya interpoladas y normalizadas, se lleva a cabo la comparación entre las curvas obtenidas y las curvas de referencia correspondientes. En otras palabras, las curvas del borde superior se comparan únicamente con las curvas de referencia de borde superior, mientras que las del borde inferior se comparan con las de referencia de borde inferior.

La comparación se realiza calculando el error cuadrático medio (MSE, por sus siglas en inglés) entre las curvas. Este valor mide la discrepancia entre las curvas y, a partir de él, se calcula un porcentaje de similitud utilizando la fórmula: $\max(0, (1 - MSE) * 100)$. Este porcentaje permite cuantificar la similitud entre las curvas, donde un valor más alto indica una mayor correspondencia entre ellas.

5.3. Modelos de aprendizaje profundo para la clasificación de planos cerebrales

5.3.1. Etiquetado

Después de la segmentación, la base de datos filtrada fue cargada en la plataforma Roboflow, donde se realizó la clasificación manual de los cuadros delimitadores que indicaban la ubicación del cerebro y el plano correspondiente. Este proceso se apoyó en la información proporcionada por el archivo Excel de la base de datos y contó con la colaboración del doctor Juan Camilo Aguado para resolver casos en los que las imágenes no eran claras. Finalmente, se lograron clasificar 2,853 imágenes del cerebro, que fueron utilizadas para entrenar el modelo YOLO.

5.3.2. Ajustes de características de las imágenes

Para optimizar el rendimiento de los modelos, se aplicaron dos procedimientos de preprocesamiento a todas las imágenes desde la plataforma Roboflow. En primer lugar, se utilizó un proceso de auto-orientación para estandarizar la orientación de las imágenes, asegurando que todas tuvieran una

alineación uniforme antes de pasar a las etapas posteriores. Lo que hace este algoritmo es revisar los metadatos EXIF, los cuales guardan datos de rotación como la posición en que fueron capturadas las imágenes, y enderezarlas automáticamente, de forma que al guardar los datos, todos queden iguales. Esto permitió que las imágenes pudieran ser modificadas de manera consistente en el siguiente paso.

En segundo lugar, se redimensionaron todas las imágenes a un formato estándar de 640x640 píxeles, convirtiéndolas en cuadradas. Además, el tamaño elegido reduce el peso de la imagen (y por ende la cantidad de recursos computacionales necesarios para trabajar con la misma) sin afectar significativamente los detalles críticos en los planos cerebrales, necesarios para clasificar cada uno de ellos. Este ajuste es crucial para garantizar la compatibilidad con los modelos YOLO, ResNet y U-Net, ya que estos modelos de detección y segmentación de objetos requieren que las imágenes tengan un formato uniforme para realizar transformaciones y análisis con mayor precisión. La forma cuadrada facilita la aplicación de convoluciones y operaciones de pooling, mejorando la capacidad del modelo para capturar las características espaciales relevantes en cada plano cerebral.

Además, el tamaño reducido de las imágenes mejora significativamente la velocidad de procesamiento durante el entrenamiento y la inferencia. Al reducir la resolución a 640x640, se logra un equilibrio entre la calidad de la imagen y la eficiencia computacional, lo que permite que los modelos procesen más imágenes en menos tiempo sin comprometer la precisión.

5.3.3. Aumento de datos y balanceo

El proceso de aumento de datos se realizó aplicando una serie de transformaciones aleatorias a las imágenes y a sus correspondientes anotaciones delimitadoras (bounding boxes), con el fin de generar nuevas variaciones de las imágenes originales. Esto ayudó a mejorar la capacidad de generalización del modelo y evitar el sobreajuste (overfitting) de clases específicas. Las transformaciones utilizadas en el pipeline fueron:

- **Flips:** Inversión horizontal y/o vertical con una probabilidad del 50%.
- **Sal y Pimienta:** Añade ruido a la imagen alterando una pequeña cantidad de píxeles (0% - 0.22% de los píxeles en la imagen).
- **Rotación:** Rotaciones de la imagen en 90°, 180°, o 270° con una probabilidad del 40%.
- **Crop:** Zoom sobre la imagen recortando de 0% - 30% de su tamaño.
- **Brillo:** Ajusta el brillo de la imagen en $\pm 15\%$.
- **Desenfoco Gaussiano:** Aplica desenfoque gaussiano con un valor sigma de hasta 2.5 píxeles.
- **Traslación:** Desplaza la imagen de 0% - 20% en las direcciones horizontal y vertical.

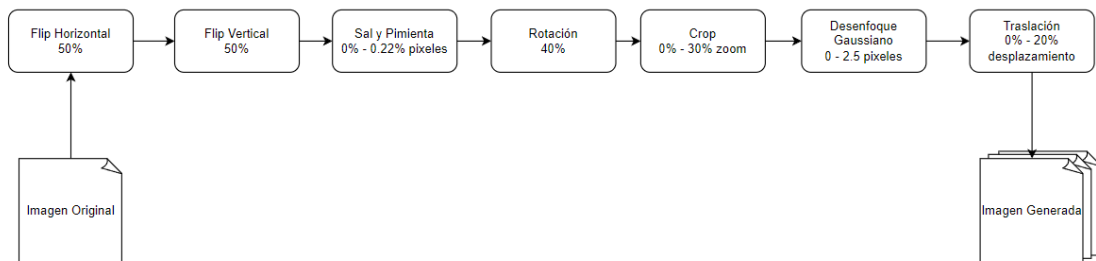


Figura 30: Secuencia del pipeline para la aumentación de datos

Para el balanceo, se organizaron las imágenes en tres clases (transcerebellum, transthalamic y transventricular) según las anotaciones asignadas a cada imagen, de manera que cada imagen se asoció

a su clase correspondiente. Luego, se contó cuántas imágenes había en cada clase para identificar la clase con menor cantidad de imágenes. Este número se utilizó para determinar el máximo de imágenes a generar, teniendo en cuenta que se crearon diez imágenes aumentadas por cada imagen original, con el objetivo de incrementar las imágenes en las clases menos representadas, igualar la cantidad de imágenes por clase y conocer el número de imágenes aumentadas necesarias para balancear las tres clases.

Para combatir el efecto de relleno y optimizar la duración del pipeline, se implementó un proceso de validación durante la generación de imágenes. Se verificó que la caja delimitadora (bounding box) estuviera completamente dentro de la imagen generada. Si la caja delimitadora no se ajusta correctamente, se vuelve a pasar la imagen original por el pipeline para generar una nueva imagen. Este proceso de realimentación tiene un límite máximo de 15 intentos para asegurar que se obtenga una imagen adecuada sin sobrecargar el sistema o prolongar innecesariamente el tiempo de procesamiento.

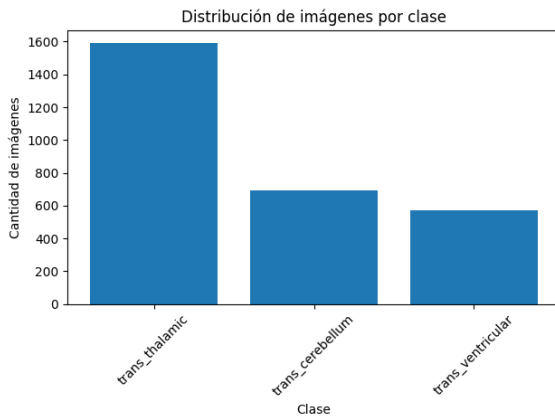


Figura 31: Histograma de clases antes del balanceo.

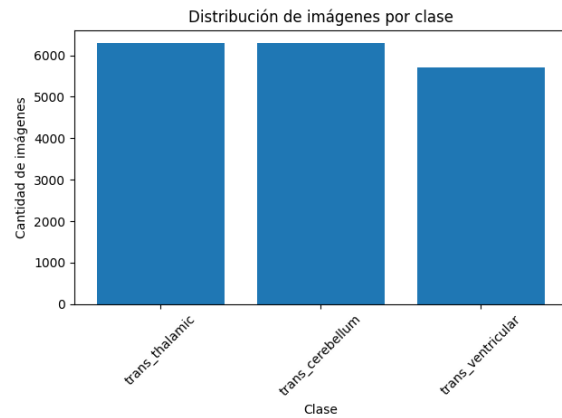


Figura 32: Histograma de clases después del balanceo.

En la base de datos original, como se muestra en la Figura 31, se observa un marcado desbalance entre las clases. La clase transthalamic es la más numerosa, con 1590 imágenes, mientras que las clases transcerebellum y transventricular tienen solo 691 y 572 imágenes respectivamente. Tras aplicar el pipeline de transformaciones, Figura 32, se obtuvo que las clases transthalamic y transcerebellum alcanzaron un total de 6292 imágenes cada una, mientras que la clase transventricular se obtuvieron 5708 imágenes, un ejemplo de las imágenes generadas se puede observar en la Figura 33.

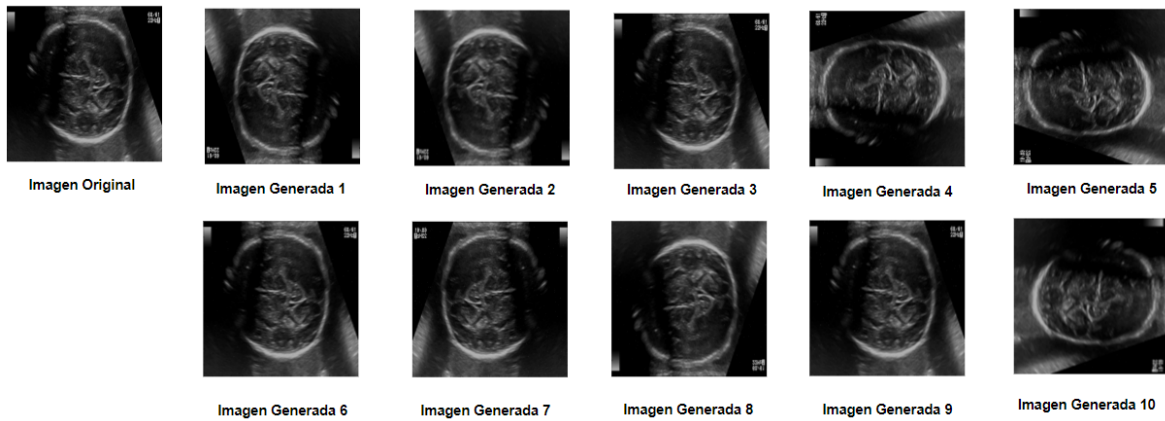


Figura 33: Contraste de la imagen original en contra las imágenes generadas por el pipeline

5.3.4. Entrenamiento

Partición de Datos Los datos se dividieron en tres conjuntos: 70 % de las imágenes fueron asignadas para entrenamiento, 20 % para las pruebas y 10 % para la validación. Esta partición tiene en cuenta la necesidad de mantener un balanceo entre las diferentes clases lo que nos permite generar modelos robustos con una cantidad significativa y representativa de imágenes que plasmen las características esenciales para diferenciar cada plano cerebral. Este enfoque equilibra la cantidad de datos disponibles para el entrenamiento y la necesidad de una evaluación efectiva, asegurando una evaluación robusta del rendimiento del modelo.

ResNet Se utilizó la librería de `PyTorch` para la creación del modelo y hacer transformaciones a las imágenes, incluyendo el redimensionamiento a 224×224 píxeles y su conversión a tensores, para los conjuntos de datos de entrenamiento, validación y pruebas. Se redimensionó a 224×224 ya que las imágenes eran pesadas y tuvieron que pasar por 18 capas, por lo que esta decisión permitió entrenar al modelo teniendo en cuenta un balance entre tiempo, eficiencia computacional y desempeño del modelo sin bajar demasiado la calidad de las imágenes[43]. Además, ese fue el tamaño que se usó para preentrenar el modelo con el conjunto de datos *ImageNet*. Se usaron `DataLoader` para cargar estos conjuntos en mini-lotes de tamaño 32, asegurando que los datos de entrenamiento se mezclaran aleatoriamente para mejorar la generalización del modelo.

Teniendo las etiquetas de las imágenes, los datos se estructuraron según la clase perteneciente de la imagen, obteniendo la jerarquía mostrada en la Figura 34.

```

/dataset
  /train
    /trans_cerebellum
    /trans_thalamic
    /trans_ventricular
  /valid
    /trans_cerebellum
    /trans_thalamic
    /trans_ventricular
  /test
    /trans_cerebellum
    /trans_thalamic
    /trans_ventricular

```

Figura 34: Estructura jerárquica de la base de datos para el modelo ResNet.

Se utilizó una ResNet-18 (ResNet con 18 capas) preentrenada en el conjunto de datos *ImageNet*. La arquitectura de la librería PyTorch proviene del artículo titulado *Deep Residual Learning for Image Recognition*, desarrollado por Kaiming He, Xiangyu Zhang, Shaoqing Ren y Jian Sun en 2015, que explora el uso de redes residuales para mejorar la precisión en la clasificación de imágenes [24]. La última capa totalmente conectada fue modificada para predecir las 3 clases necesarias en lugar de las 1000 clases originales de *ImageNet*. Esto implica que solo las últimas capas de la red se ajustaron, mientras que las capas más profundas conservaron los pesos preentrenados. A pesar de esto, en el artículo origen de ResNet también usan técnicas de aumento de color y se usan imágenes RGB, a diferencia de en este proyecto que se adaptó a imágenes en escala de grises, lo que puede hacer que el desempeño del modelo varíe en comparación con otros conjuntos de datos.

Se eligió ResNet-18 ya que, por su arquitectura, permite recordar características a través de sus múltiples capas, evitando el problema del gradiente descendiente. Además, se escogieron las 18 capas, puesto que era el modelo más liviano, lo que permite que pueda ser integrado en máquinas con menos recursos o ecógrafos en un futuro.

La CNN se entrenó con el conjunto ImageNet; sin embargo, ha conseguido primer lugar en competencias como *ImageNet localization*, *ImageNet detection*, *COCO(Common Objects in Context) detection* y *COCO segmentation*[24]. Esto sugiere una buena generalización del modelo, sin embargo, la diferencia del tipo de imágenes puede afectar el desempeño, como en el artículo *Bird image classification based on improved ResNet-152 image classification model*[54] en el cual se alcanzó una precisión del 96.5% con el conjunto de datos a color *BIRDS 525 SPECIES* mientras que en *Image Classification With The Hybrid PVCN*[3] se obtuvo una precisión del 91.3% usando imágenes en blanco y negro y a color, aunque esta variación también puede ser atribuida a la diferencia de número de capas y a la combinación de arquitectura con otros modelos. En *White blood cells segmentation and classification using a random forest and residual networks implementation*[20] se utilizó ResNet 50 para segmentar y clasificar células blancas y se obtuvo una precisión de 99.3%, sin embargo, estas imágenes eran a color, pero probó ser útil para aplicaciones médicas.

En el Cuadro 2 se presenta una tabla que describe la estructura de las diferentes variantes de ResNet, tal como se detalla en el artículo *Deep Residual Learning for Image Recognition* [24], incluida la ResNet de 18 capas utilizada en este trabajo. La ResNet-18 se compone de 5 bloques de convolución, con la primera capa siendo una convolución de 7×7 , seguida de una capa de max pooling. A partir de ahí, las capas de convolución aplican filtros de 3×3 , con canales que comienzan en 64 y se duplican en cada bloque, llegando hasta 512. Esto permite que el modelo capture características cada vez más complejas a medida que se avanza a través de las capas.

En la columna de tamaño de salida de la tabla se puede observar cómo las dimensiones espaciales de las características se reducen progresivamente en cada bloque, hasta que finalmente la salida es

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	$7 \times 7, 64$ stride 2				
conv2_x	56×56	3×3 maxpool, stride 2				
		$\begin{bmatrix} 3 \times 3 & 64 \\ 3 \times 3 & 64 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3 & 64 \\ 3 \times 3 & 64 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1 & 64 \\ 3 \times 3 & 64 \\ 1 \times 1 & 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1 & 64 \\ 3 \times 3 & 64 \\ 1 \times 1 & 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1 & 64 \\ 3 \times 3 & 64 \\ 1 \times 1 & 256 \end{bmatrix} \times 3$
conv3_x	28×28	$\begin{bmatrix} 3 \times 3 & 128 \\ 3 \times 3 & 128 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3 & 128 \\ 3 \times 3 & 128 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1 & 128 \\ 3 \times 3 & 128 \\ 1 \times 1 & 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1 & 128 \\ 3 \times 3 & 128 \\ 1 \times 1 & 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1 & 128 \\ 3 \times 3 & 128 \\ 1 \times 1 & 512 \end{bmatrix} \times 8$
conv4_x	14×14	$\begin{bmatrix} 3 \times 3 & 256 \\ 3 \times 3 & 256 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3 & 256 \\ 3 \times 3 & 256 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1 & 256 \\ 3 \times 3 & 256 \\ 1 \times 1 & 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1 & 256 \\ 3 \times 3 & 256 \\ 1 \times 1 & 1024 \end{bmatrix} \times 23$	$\begin{bmatrix} 1 \times 1 & 256 \\ 3 \times 3 & 256 \\ 1 \times 1 & 1024 \end{bmatrix} \times 36$
conv5_x	7×7	$\begin{bmatrix} 3 \times 3 & 512 \\ 3 \times 3 & 512 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3 & 512 \\ 3 \times 3 & 512 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1 & 512 \\ 3 \times 3 & 512 \\ 1 \times 1 & 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1 & 512 \\ 3 \times 3 & 512 \\ 1 \times 1 & 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1 & 512 \\ 3 \times 3 & 512 \\ 1 \times 1 & 2048 \end{bmatrix} \times 3$
	1×1	average pool, 1000-d fc, softmax				
Flops		$1,8 \times 10^9$	$3,8 \times 10^9$	$7,6 \times 10^9$	$11,3 \times 10^9$	

Cuadro 2: Tabla de arquitecturas ResNet.[24].

de tamaño 1×1 . Este proceso de reducción de tamaño permite condensar la información espacial y facilitar la clasificación en las capas finales.

El modelo implementado en este proyecto conserva la misma estructura de ResNet-18, pero con una modificación en la capa final totalmente conectada, que originalmente tiene 1000 salidas (correspondientes a las clases del conjunto de datos *ImageNet*). En este caso, la capa final ha sido ajustada para que tenga solo 3 salidas, correspondientes a las tres categorías de clasificación de las imágenes cerebrales fetales: transventricular, transtalámico y transcerebeloso.

El entrenamiento ocurrió durante 100 épocas, cada una con dos fases: una fase de entrenamiento, donde el modelo se actualizaba, y una fase de validación, donde el modelo era evaluado sin actualización de los pesos. Se realizó así, ya que primero se quería analizar el comportamiento del modelo durante las épocas y luego se entrenaron las demás con las mismas épocas para conservar uniformidad.

Durante cada iteración, se calcularon la pérdida, exhaustividad, F1-score y la precisión tanto en el conjunto de entrenamiento como en el de validación. Si la precisión en la validación mejoraba, los pesos del modelo se guardaban como los mejores encontrados hasta el momento.

El modelo se entrenó usando la función de pérdida **CrossEntropyLoss**, que es comúnmente utilizada para tareas de clasificación de múltiples clases. El optimizador seleccionado fue **Adam** con una tasa de aprendizaje inicial de 0.001, esto ya que se quería evitar que el modelo oscile o diverja. En adición, el modelo fue entrenado en 100 epochs, por lo que se optó por elegir una tasa de aprendizaje más pequeña, pero que entra dentro de los valores comúnmente utilizados dentro de este tipo de redes. Además, se utilizó un programador de tasa de aprendizaje **StepLR**, que redujo la tasa de aprendizaje en un factor de 0.1 cada 7 épocas, la cual fue escogida para evitar que el modelo sea más estable y no se quede atascado en mínimos locales durante su entrenamiento para que ajuste y refine los pesos hacia el final del entrenamiento.

En la Figura 35 se muestra de forma gráfica como funciona esta arquitectura por cada imagen.

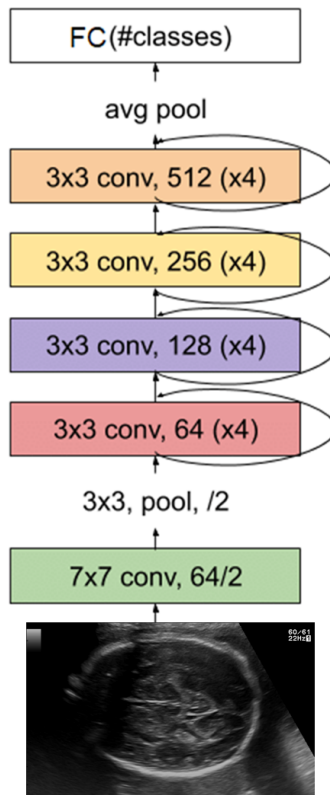


Figura 35: Arquitectura de la ResNet-18 para planos cerebrales. Adaptado de [5] con modificaciones.

DenseNet Para este modelo también se utilizó la librería de `PyTorch` para la creación del modelo y hacer transformaciones a las imágenes, incluyendo el redimensionamiento a 224×224 píxeles, transformación a escala de grises en un canal (en lugar de RGB, es decir a color) para que fuera una entrada válida para el modelo y su conversión a tensores, para los conjuntos de datos de entrenamiento, validación y pruebas. Se redimensionó a 224×224 ya que las imágenes eran pesadas y este modelo es reconocido por su densidad y alta conectividad entre capas por lo que esta decisión permitió entrenar al modelo sin comprometer su desempeño, al igual que el de la ResNet-18. Se usaron `DataLoader` para cargar estos conjuntos en mini-lotes de tamaño 32, asegurando que los datos de entrenamiento se mezclaran aleatoriamente para mejorar la generalización del modelo.

Se utilizó una DenseNet121 (DenseNet con 121 capas) preentrenada cuya arquitectura proviene del artículo *Densely Connected Convolutional Networks*, desarrollado por Gao Huang, Zhuang Liu y Laurens van der Maaten en 2018, en el cual introducen las redes convolucionales densas que pretenden mejorar la precisión y eficiencia a la hora de entrenar redes convolucionales para el reconocimiento de objetos [27].

En el Cuadro 3 se presenta una tabla que describe la estructura de las diferentes variantes de DenseNet, tal como se detalla en el artículo *Densely Connected Convolutional Networks* [27], incluida la DenseNet de 121 capas utilizada en este trabajo. La DenseNet121 está compuesta de 4 bloques densos, 3 capas de transición y una capa de clasificación. Los bloques densos se encargan de hacer procesos de convolución repetidas veces en donde se tratan de encontrar características relevantes automáticamente, mientras que en los bloques de transición lo que se hace es reducir la dimensionalidad de la imagen para poder pasarla al siguiente bloque denso. El modelo propuesto conserva la estructura del original a excepción de la capa de clasificación, la cual es una capa totalmente conectada que originalmente tiene predicciones para 1000 clases al haber sido entrenado con el dataset *ImageNet*, sin

Layers	Output size	DenseNet-121	DenseNet-169	DenseNet-201	DenseNet-264
Convolution	112×112	7×7 conv, stride 2			
Pooling	56×56	3×3 max pool, stride 2			
Dense Block (1)	56×56	$\begin{bmatrix} 1 \times 1 \\ 3 \times 3 \end{bmatrix} \text{ conv} \times 6$	$\begin{bmatrix} 1 \times 1 \\ 3 \times 3 \end{bmatrix} \text{ conv} \times 6$	$\begin{bmatrix} 1 \times 1 \\ 3 \times 3 \end{bmatrix} \text{ conv} \times 6$	$\begin{bmatrix} 1 \times 1 \\ 3 \times 3 \end{bmatrix} \text{ conv} \times 6$
Transition Layer (1)	56×56 28×28	1×1 conv 2×2 average pool, stride 2			
Dense Block (2)	28×28	$\begin{bmatrix} 1 \times 1 \\ 3 \times 3 \end{bmatrix} \text{ conv} \times 12$	$\begin{bmatrix} 1 \times 1 \\ 3 \times 3 \end{bmatrix} \text{ conv} \times 12$	$\begin{bmatrix} 1 \times 1 \\ 3 \times 3 \end{bmatrix} \text{ conv} \times 12$	$\begin{bmatrix} 1 \times 1 \\ 3 \times 3 \end{bmatrix} \text{ conv} \times 12$
Transition Layer (2)	28×28 14×14	1×1 conv 2×2 average pool, stride 2			
Dense Block (3)	14×14	$\begin{bmatrix} 1 \times 1 \\ 3 \times 3 \end{bmatrix} \text{ conv} \times 24$	$\begin{bmatrix} 1 \times 1 \\ 3 \times 3 \end{bmatrix} \text{ conv} \times 32$	$\begin{bmatrix} 1 \times 1 \\ 3 \times 3 \end{bmatrix} \text{ conv} \times 48$	$\begin{bmatrix} 1 \times 1 \\ 3 \times 3 \end{bmatrix} \text{ conv} \times 64$
Transition Layer (3)	14×14 7×7	1×1 conv 2×2 average pool, stride 2			
Dense Block (4)	7×7	$\begin{bmatrix} 1 \times 1 \\ 3 \times 3 \end{bmatrix} \text{ conv} \times 16$	$\begin{bmatrix} 1 \times 1 \\ 3 \times 3 \end{bmatrix} \text{ conv} \times 32$	$\begin{bmatrix} 1 \times 1 \\ 3 \times 3 \end{bmatrix} \text{ conv} \times 32$	$\begin{bmatrix} 1 \times 1 \\ 3 \times 3 \end{bmatrix} \text{ conv} \times 48$
Classification Layer	1×1	7×7 global average pool 1000D fully-connected, softmax			

Cuadro 3: Tabla de arquitecturas DenseNet.[27].

embargo, para este caso solo hay 3 clases correspondientes a las tres categorías de clasificación de las imágenes cerebrales fetales: tranststalámico, tranststalámico y tranststalámico.

En la columna de tamaño de salida de la tabla se puede observar cómo las dimensiones espaciales de las características se reducen progresivamente en cada bloque, hasta que finalmente la salida es de tamaño 1×1 . Este proceso de reducción de tamaño permite condensar la información espacial y facilitar la clasificación en las capas finales.

Al igual que en la ResNet-18 el modelo se entrenó usando la función de pérdida **CrossEntropyLoss**. El optimizador seleccionado fue **Adam** con una tasa de aprendizaje inicial de 0.001. Además, se utilizó un programador de tasa de aprendizaje **StepLR**, que redujo la tasa de aprendizaje en un factor de 0.1 cada 7 épocas. El entrenamiento ocurrió de igual forma que con el modelo anterior durante 100 épocas, cada una con dos fases: una fase de entrenamiento y una fase de validación. Durante cada iteración, se calcularon la pérdida, exhaustividad, F1-score y la precisión tanto en el conjunto de entrenamiento como en el de validación. Si la precisión en la validación mejoraba, los pesos del modelo se guardaban como los mejores encontrados hasta el momento.

Se utilizaron los mismos valores en hiperparámetros y épocas para poder comparar los modelos uniformemente, ya que ambos se escogieron en su versión más liviana (con menos capas) y por su capacidad de clasificar evitando el problema del gradiente descendente.

En la Figura 36 se puede ver gráficamente como es la arquitectura DenseNet donde *TT* se refiere al plano tranststalámico y las capas de transición están compuestas por la convolución y operaciones de pooling.



Figura 36: Arquitectura DenseNet. Adaptado de [2] con modificaciones.

YOLO Para el proceso de entrenamiento se utilizó el modelo de YOLOv8 de la librería de Python llamada **Ultralytics**, la cual provee una interfaz de alto nivel para el entrenamiento y uso de modelos de detección de objetos basados en YOLO. Este paso permite acceder a todas las funcionalidades

de la librería, incluyendo la posibilidad de cargar modelos preentrenados, configurar parámetros de entrenamiento, realizar predicciones y exportar modelos.

Para llevar a cabo el entrenamiento del modelo, se deben preparar los datos en el formato de datos requerido por YOLOv8 que incluye las imágenes y sus correspondientes etiquetas, donde las anotaciones deben seguir el formato YOLO. En este formato, por cada imagen del conjunto de datos se debe generar un archivo `.txt` que contenga las etiquetas de los objetos presentes en la imagen, junto con las coordenadas de los cuadros delimitadores. Cada línea en este archivo representa un objeto y sigue la estructura mostrada en la Figura 37.

```
<clase> <x_centro> <y_centro> <ancho> <alto>
```

Figura 37: Formato de etiqueta para imágenes en YOLO.

Donde las coordenadas y dimensiones están normalizadas entre 0 y 1 con respecto a las dimensiones de la imagen. El conjunto de datos debe estructurarse en directorios que separen las imágenes y las etiquetas para el entrenamiento, la validación y las pruebas, dada por la jerarquía en la Figura 38.

```
/dataset
  data.yaml
  /test
    /images
    /labels
  /train
    /images
    /labels
  /valid
    /images
    /labels
```

Figura 38: Estructura jerárquica de la base de datos para el modelo YOLO.

Se utilizó `yo1ov8n.pt` que contiene los pesos preentrenados del modelo YOLOv8n, los cuales sirven como punto de partida para el entrenamiento sobre los datos personalizados. Una vez terminada la organización, se definió una serie de hiperparámetros clave que influirán en el comportamiento del modelo. Estos parámetros pueden configurarse mediante la función `train()` donde:

- **data:** Especifica la ruta del archivo `data.yaml` que describe el conjunto de datos, incluyendo las rutas a las imágenes de entrenamiento, validación y pruebas, así como los nombres de las clases.
- **epochs:** Define el número de épocas para el entrenamiento, en nuestro caso fueron 100.
- **imgsz:** Establece el tamaño de las imágenes que se utilizarán durante el entrenamiento. En este caso, cada imagen se redimensionará a 640x640 píxeles, dado que este modelo se caracteriza por su velocidad y precisión, así se puede conservar un tamaño mayor sin comprometer el tiempo de computación.[\[59\]](#)
- **batch:** Determina el número de imágenes que se procesarán simultáneamente en cada paso del entrenamiento, para el estudio fueron 32.

Dentro del proceso de entrenamiento, el modelo ajusta sus pesos mediante un proceso iterativo conocido como retropropagación (*backpropagation*). A medida que avanza el entrenamiento, el modelo intenta minimizar la diferencia entre las predicciones que realiza y las etiquetas reales proporcionadas en el conjunto de datos. En cada época, el modelo evalúa su rendimiento utilizando un conjunto de métricas que incluyen la precisión, recall, mAP50 y mAP50-95.

Una vez finalizado el entrenamiento, el modelo es evaluado en el conjunto de validación, lo que permite determinar su capacidad de generalización a datos no vistos previamente. Si el modelo alcanza un rendimiento satisfactorio, los pesos del mejor modelo se guardan automáticamente en un archivo llamado `best.pt`, el cual puede ser utilizado posteriormente para hacer predicciones.

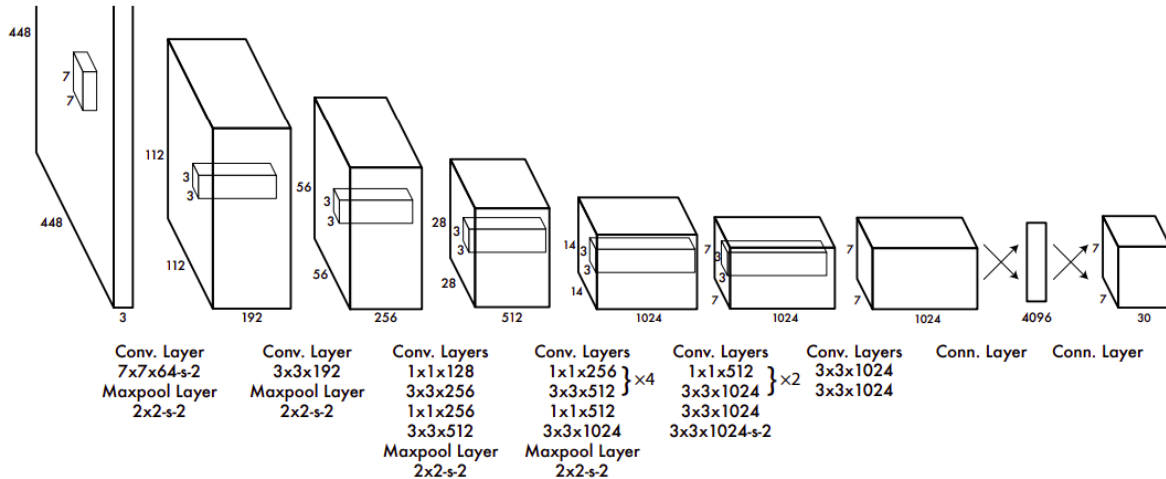


Figura 39: **Arquitectura YOLO:** La red de detección tiene 24 capas convolucionales seguidas de 2 capas totalmente conectadas. La alternancia de capas convolucionales 1×1 reduce el espacio de características de las capas precedentes.[46]

6. Pruebas y Resultados

6.1. Validación del algoritmo de segmentación

El algoritmo propuesto para la segmentación de ECP y no ECP muestra un desempeño aceptable, evidenciado en el Cuadro 4 con una precisión de 92 % y un Recall de 78 %. Esto indica que el modelo tiene una alta capacidad para evitar falsos positivos, pero muestra cierta limitación al identificar correctamente todos los casos positivos (falsos negativos). El F1-Score resultante de 0.84 refleja un equilibrio razonable entre precisión y sensibilidad.

	Precision	Recall	F1-score	Support
ECP	0.92	0.78	0.84	4658
Accuracy			0.84	4658
Macro avg	0.92	0.78	0.84	4658

Cuadro 4: Informe para la segmentación de imágenes craneales y no craneales

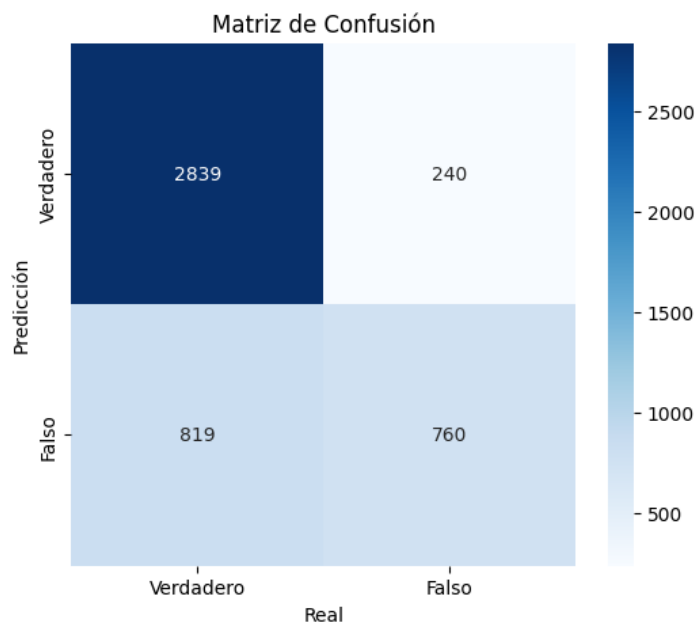


Figura 40: Matriz de confusión testeando algoritmo de segmentación

6.2. Validación de los modelos de aprendizaje automático

6.2.1. Métricas

Para comparar el desempeño de los distintos modelos, se elaboraron tablas que incluyen las métricas de exactitud (Precision), exhaustividad (Recall) y F1-score más la precisión global (Accuracy), las cuales proporcionan una visión clara de la capacidad de cada modelo para clasificar planos craneales prenatales. Además, se incluyó la métrica *Support* para cada plano, que indica el número de imágenes de cada clase en el conjunto de prueba. Como se observa en las tablas, las clases están equilibradas en cuanto al número de ejemplos.

Cada modelo también cuenta con gráficos que muestran la evolución de las métricas a lo largo de las 100 épocas, tanto para el conjunto de entrenamiento como para el conjunto de validación. Estos

gráficos permiten evaluar cómo mejoró o empeoró el rendimiento del modelo a medida que avanzaban las épocas, ayudando a determinar el punto en que sería pertinente detener el entrenamiento para un modelo similar y así ahorrar tiempo y recursos computacionales. Por último, se incluye una matriz de confusión, que permite visualizar la cantidad de clasificaciones correctas por clase y en qué casos se produjeron los mayores errores.

ResNet Resultados de validación del modelo entrenado:

El modelo ResNet-18 alcanzó una exactitud (Accuracy) global del 93%, como se observa en el Cuadro 5. La clase con mayor precisión fue la transcerebelosa, mientras que la clase con menor desempeño fue la transventricular. Esto puede deberse a que esta última tiene un menor número de imágenes en general, con 584 menos que las otras dos clases, y que es la categoría con mayor número de imágenes de prueba (66 más que la clase transcerebeloso y 206 más que la clase transventricular), siendo esta última razón menos probable, ya que no hay una diferencia significativa en la cantidad de imágenes de prueba, y la clase con menor número de imágenes no es la que mejor se desempeña.

	Precision	Recall	F1-score	Support
trans_cerebellum	0.96	0.96	0.96	1244
trans_thalamic	0.91	0.90	0.91	1310
trans_ventricular	0.92	0.93	0.93	1104
Accuracy			0.93	3658
Macro avg	0.93	0.93	0.93	3658

Cuadro 5: Informe de clasificación para cada clase en el modelo ResNet

En los gráficos de la Figura 41 se muestra la evolución de las métricas tanto para validación (Val) como para entrenamiento (Train). No obstante, las métricas son tan similares que se superponen para su respectiva categoría, lo que indica un desempeño consistente del modelo en la detección de planos y en el balance general del modelo. Además, alrededor de la época 20, las métricas de pérdida (loss) y las de desempeño comienzan a estabilizarse. Por lo tanto, si se recreara el experimento o se entrenaran nuevos modelos, sería recomendable limitar el número de épocas a 40. Antes de la época 20, se observan fluctuaciones en las métricas de validación, lo que refleja cómo el modelo ajustaba sus pesos para mejorar su rendimiento.

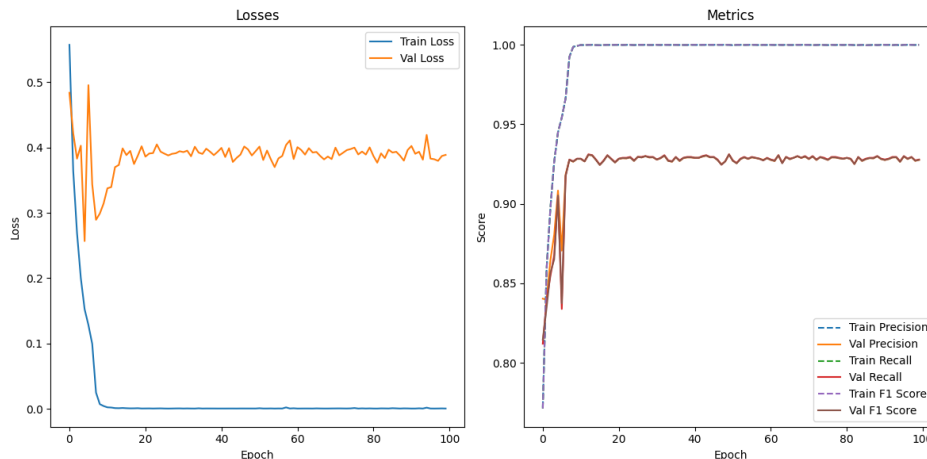


Figura 41: Métricas por generación entrenamiento ResNet

En la matriz de confusión que puede verse en la Figura 42 se puede apreciar de forma gráfica cómo la clase transcerebelosa es la que fue clasificada con mayor éxito, mientras que hay una mayor

confusión entre las clases transventricular y transtálamica, donde la clase transtálamica fue la segunda con mejores resultados.

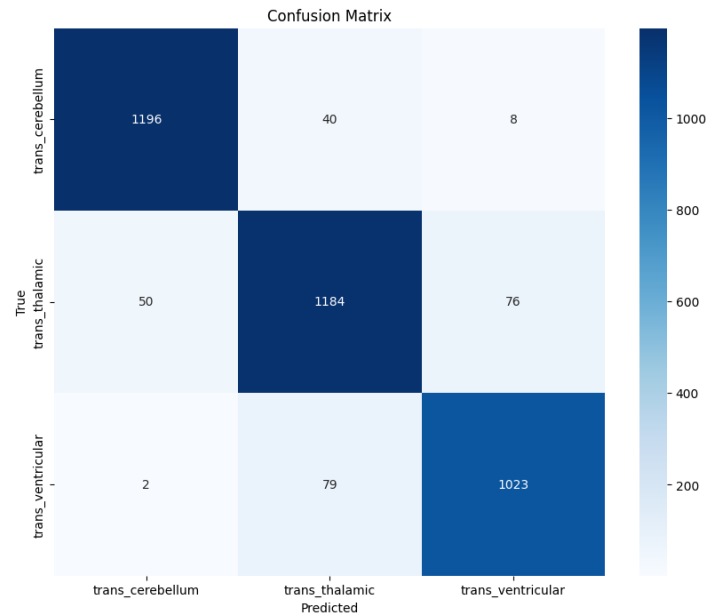


Figura 42: Matriz de confusión testeo ResNet

DenseNet Resultados de validación del modelo entrenado:

La exactitud (Accuracy) global alcanzada por el modelo DenseNet fue de 94%, lo cual es aún mayor que la Resnet. Al igual que en el anterior modelo, este clasificó con mayor exactitud el plano transcerebeloso, superando la sensibilidad (Recall) y el F1-score de la ResNet en un 1% como se ve en el Cuadro 6. Al igual que en el modelo anterior, el plano que se clasificó con menor precisión fue el transtalámico, sin embargo, en este caso superó la precisión anterior por un 3%. En general superó al modelo ResNet en todas las métricas.

	Precision	Recall	F1-score	Support
trans.cerebellum	0.96	0.97	0.97	1244
trans.thalamic	0.93	0.92	0.92	1310
trans.ventricular	0.94	0.94	0.94	1104
Accuracy			0.94	3658
Macro avg	0.94	0.94	0.94	3658

Cuadro 6: Informe de clasificación para cada clase en el modelo DenseNet

En los gráficos de la Figura 43, se observa un comportamiento similar al del modelo ResNet, con la diferencia de que este modelo reduce de manera constante la métrica de pérdida (Loss) y aumenta de forma continua las métricas de desempeño (excepto por una época en la que las métricas empeoraron), lo que indica que necesitó realizar menos ajustes de pesos que el modelo anterior y se adaptó más rápidamente a los datos. La métrica de loss en la validación se estabiliza alrededor de la epoch 30 en un valor de 0.38 aproximadamente.

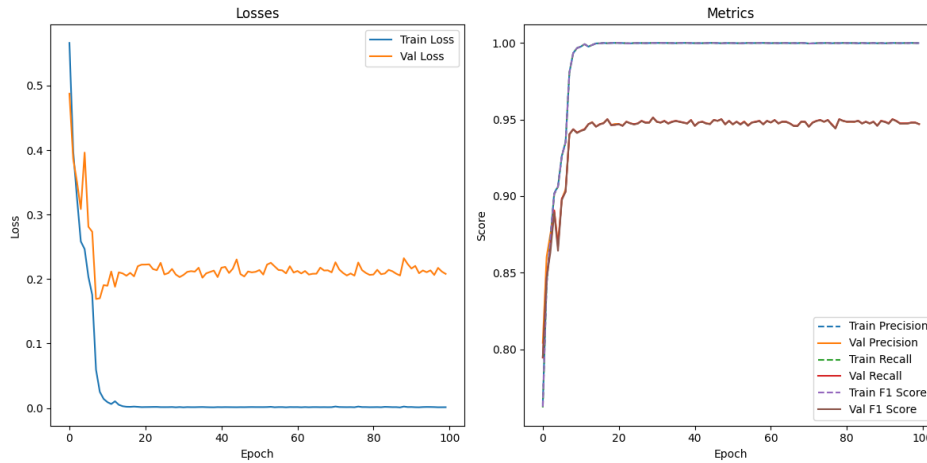


Figura 43: Métricas por generación entrenamiento DenseNet

En la matriz de confusión que puede verse en la Figura 44 se conserva la tendencia donde la clase transcerebelosa es la mejor clasificada, seguida de la transtálamica, con la marcada confusión de esta y la transventricular a pesar de las prometedoras métricas.

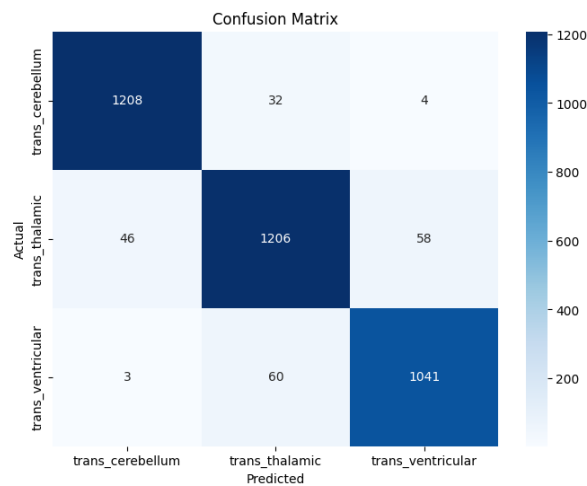


Figura 44: Matriz de confusión testeo DenseNet

YOLO Resultados de validación del modelo entrenado:

La exactitud (Accuracy) global alcanzada por el modelo YOLO fue del 97%, siendo la mejor obtenida entre todos los modelos, superando en todas las métricas a los dos modelos anteriores. Al igual que en los modelos previos, el plano transcerebeloso fue el que se clasificó con mayor precisión; sin embargo, en este caso, el plano transtalámico obtuvo la misma métrica de precisión que el plano transventricular. A pesar de esto, las medidas de sensibilidad (Recall) y F1-score siguen siendo inferiores en el plano transtalámico, como se puede apreciar en el Cuadro 7, lo que sugiere que se trata de un problema general que no se debe al modelo o al entrenamiento, sino que podría estar relacionado con la escasez de imágenes en el dataset o con el tipo de estructuras a identificar en dicho plano en específico.

En la Figura 45 se observa que ambas métricas (precisión y recall) muestran un crecimiento continuo y una estabilización cercana a 0.97 tanto para la precisión y el recall, lo que indica que el modelo

	Precision	Recall	F1-score	Support
trans_cerebellum	0.98	0.98	0.98	1244
trans_thalamic	0.96	0.95	0.96	1310
trans_ventricular	0.96	0.98	0.97	1104
Accuracy			0.97	3658
Macro avg	0.97	0.97	0.97	3658

Cuadro 7: Informe de clasificación para cada clase en el modelo YOLO

ha logrado un buen equilibrio entre predecir correctamente la clasificación de las imágenes presentes, evitar errores de predicción (falsos positivos) y evitar errores al no detectar imágenes (falsos negativos). Adicionalmente, las curvas indican que el modelo ha convergido, con un rendimiento cercano al óptimo, sin que se observe un comportamiento errático o caídas en el rendimiento.

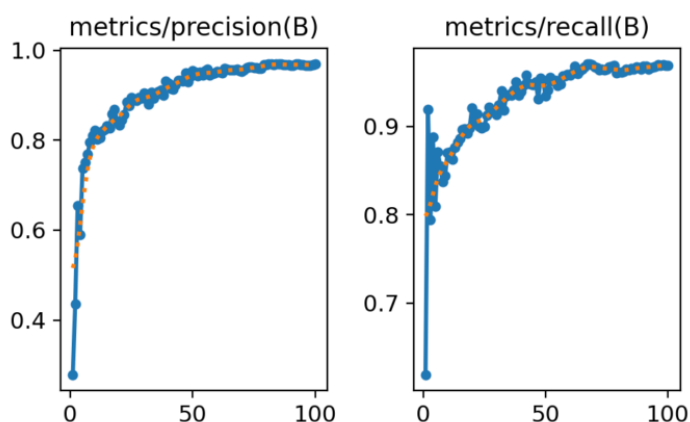


Figura 45: Métricas de precisión y sensibilidad por generación del modelo YOLO

En la Figura 46 se muestra las curvas de pérdida (loss) durante el entrenamiento (train) y validación del modelo (val), dónde se observan las siguientes categorías:

- **box_loss**: Pérdida relacionada con la precisión de las predicciones de las cajas delimitadoras durante el entrenamiento y la validación.
- **cls_loss**: Pérdida asociada con la clasificación de objetos durante el entrenamiento y la validación.
- **dfl_loss**: Pérdida relacionada con la regresión de distribución en la predicción de las coordenadas de los bordes de las cajas (Distance Focal Loss, DFL) durante el entrenamiento y la validación.

Las curvas muestran una disminución consistente tanto en el conjunto de entrenamiento como en el de validación, esto es esperable y un buen signo de que el modelo se está ajustando correctamente. Las pérdidas en el conjunto de validación son bastante similares a las del conjunto de entrenamiento, lo que sugiere que no hay un sobreajuste (overfitting) significativo. Si hubiera un sobreajuste, las pérdidas de validación serían mucho mayores que las de entrenamiento. Adicionalmente, las pérdidas tienden a estabilizarse a medida que avanzan las épocas, esto es un indicador positivo de que el modelo está convergiendo, y probablemente esté cerca de su rendimiento óptimo en estas condiciones.

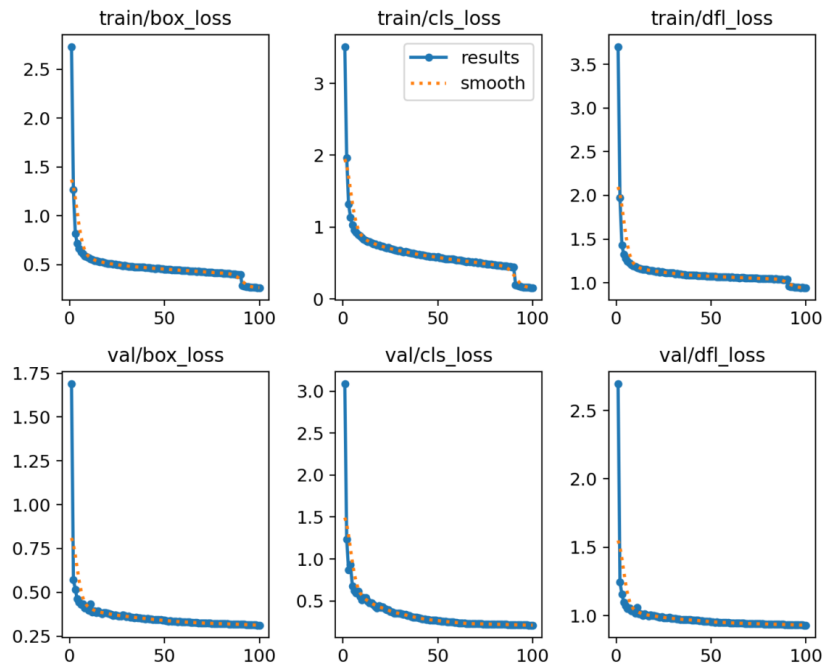


Figura 46: Gráfica de pérdidas por generación en entrenamiento y validación modelo YOLO

En la matriz de confusión que puede verse en la Figura 47 se rompe con la tendencia que seguían los anteriores dos modelos, puesto que, a pesar de que la clase transcerebelosa siga siendo la que conserva la mayor precisión, la que le sigue en este caso es la transventricular. Sin embargo, todavía se conserva la mayor confusión entre las clases transtálmica y transventricular, aunque en menor proporción que en los anteriores modelos.

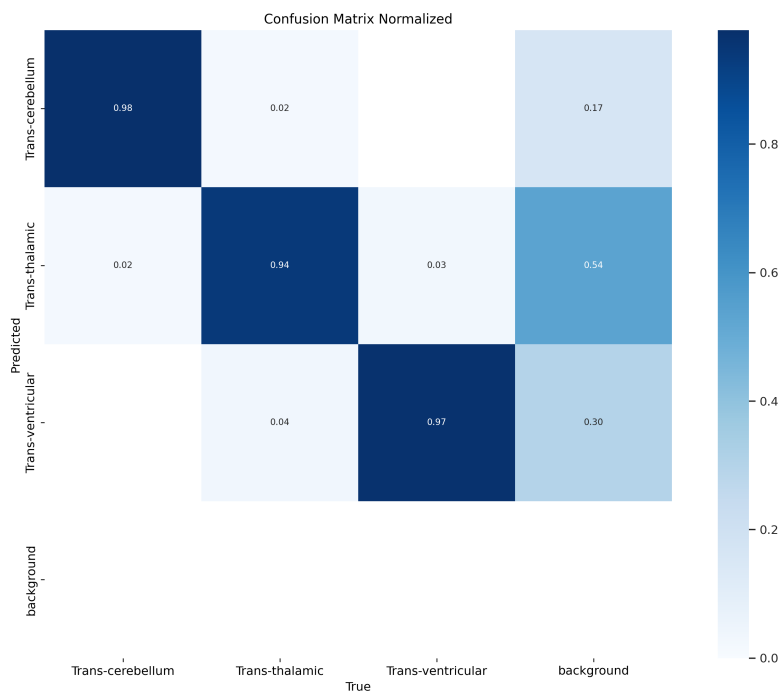


Figura 47: Matriz de confusión testeo YOLO

6.2.2. Validación con profesionales de la salud

Para la validación con los profesionales de la salud, se creó un subconjunto balanceado de la carpeta de pruebas utilizada para evaluar los modelos con 201 imágenes. A cada profesional se le pidió clasificar las imágenes según su criterio, con la posibilidad de utilizar material de apoyo como imágenes de referencia o libros especializados, para garantizar la mayor precisión en sus evaluaciones. Ambos profesionales, que trabajaron de forma independiente y sin comunicación entre ellos, coincidieron en señalar que algunas imágenes no permitían una adecuada visualización de los planos. Esto se debió a que, en su opinión, ciertas imágenes correspondían a cortes angulados, no se apreciaban del todo las características de los planos o, en algunos casos, ni siquiera representaban el cráneo fetal. Con lo anterior, se hizo la evaluación de cada uno teniendo en cuenta solo las imágenes que ambos pudieran clasificar, dejando por fuera un total de 16 imágenes para hacer una evaluación con 186 en total.

Dr. Juan Camilo Aguado El Dr. Juan Camilo Aguado es médico cirujano graduado con honores de la Pontificia Universidad Javeriana de Bogotá en el año 2023. Cuenta con experiencia en liderazgo de grupos de investigación y participación en múltiples eventos académicos. Cuenta con publicaciones en el área de cuidado crítico. Actualmente se desempeña en cirugía de trauma.

La clase que el Dr. Aguado clasificó con mayor precisión fue el plano transcerebeloso, como se muestra en el Cuadro 8. En contraste, el plano transtalámico fue el que presentó el peor desempeño, una tendencia que también se observó en todos los modelos de aprendizaje automático desarrollados previamente. Tal como se esperaba, su precisión global del 70 % estuvo por debajo de la de los modelos analizados.

En la Figura 48 se presenta la matriz de confusión de las clasificaciones realizadas por el Dr. Aguado. Se observa que la clase transcerebelosa es la más fácil de clasificar tanto para los modelos como para los profesionales. Sin embargo, existe un notable porcentaje de confusión entre las clases transtalámicas y transcerebelosas.

	Precision	Recall	F1-score	Support
trans_cerebellum	0.88	0.89	0.89	66
trans_thalamic	0.48	0.56	0.52	52
trans_ventricular	0.72	0.63	0.67	67
Accuracy			0.70	185
Macro avg	0.70	0.69	0.69	185
Weighted avg	0.71	0.70	0.71	185

Cuadro 8: Informe de clasificación para cada clase en las predicciones de Camilo

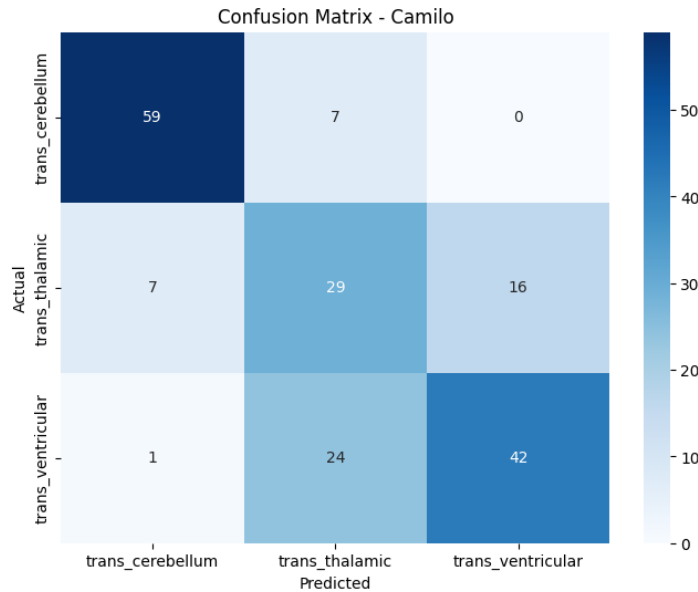


Figura 48: Matriz de confusión testeo Dr. Juan Camilo Aguado

Dra. Natalia Andrea Torres La Dra. Natalia Andrea Torres se graduó de medicina de la Universidad del Quindío en 2014, luego se especializó en ginecología y obstetricia en la Universidad Militar Nueva Granada en 2020 y, por último, se especializó en medicina materna fetal en la Fundación Universitaria Ciencias de la Salud en 2022. Además, ha participado en varias publicaciones, donde destaca *Anatomical structure characterization of fetal ultrasound images using texture-based segmentation technique via an interactive MATLAB application* [40] por su relación con este proyecto. Por lo anterior, su experiencia es relevante para los temas del proyecto.

Como se observa en el Cuadro 9, la Dra. Torres también clasificó con mayor precisión el plano transcerebeloso, lo que continúa la tendencia observada en todos los casos anteriores. Sin embargo, en este caso, la clase peor clasificada fue la transventricular en lugar de la transtalámica. Esto es relevante, ya que la Dra. Torres posee el mayor conocimiento técnico del proyecto, lo que sugiere que las estructuras correspondientes a la clase transtalámica no son las más difíciles de identificar para alguien con un sólido entrenamiento técnico.

En la Figura 49 se presenta la matriz de confusión de las clasificaciones realizadas por la Dra. Torres. Se observa una mejora con respecto a la del Dr. Aguado, atribuible a su mayor experticia en el tema. Además, se ha reducido considerablemente la confusión entre las clases transventricular y transtalámica; sin embargo, persiste la confusión significativa entre las clases transtalámica y transventricular.

	Precision	Recall	F1-score	Support
trans_cerebellum	0.91	0.97	0.94	66
trans_thalamic	0.87	0.65	0.75	52
trans_ventricular	0.82	0.93	0.87	67
Accuracy			0.86	185
Macro avg	0.87	0.85	0.85	185
Weighted avg	0.87	0.86	0.86	185

Cuadro 9: Informe de clasificación para cada clase en las predicciones de Natalia

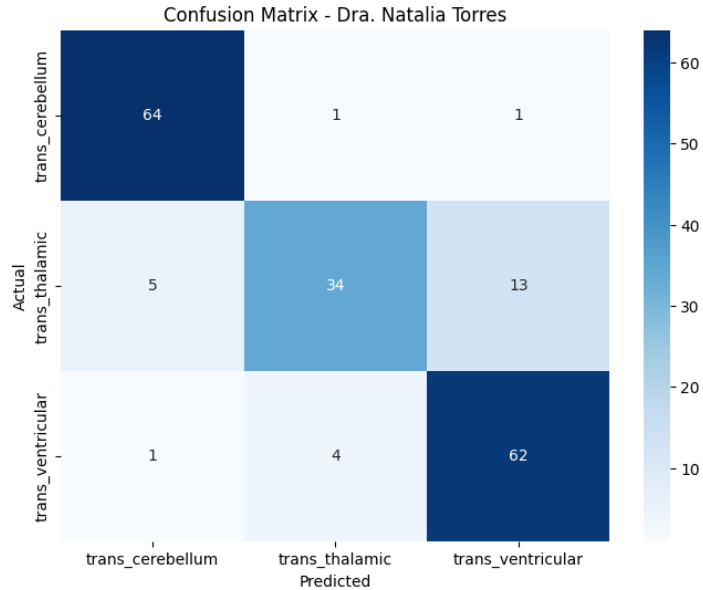


Figura 49: Matriz de confusión testeo Dra. Natalia Andrea Torres

Modelos

En contraste con estos resultados, se han puesto a prueba los modelos entrenados a predecir una clasificación a esas mismas 185 imágenes presentadas para comparar su rendimiento respecto a los profesionales:

ResNet En el Cuadro 10 se observa que el modelo ResNet tiene un rendimiento variable según la clase que se evalúa, sugiriendo que el modelo puede mejorar en la predicción de ciertas clases, en particular la transtalámica.

En el plano del transcerebelo, la ResNet logra una alta precisión (96%), pero un recall del 68%, lo que indica que identifica correctamente la mayoría de las predicciones positivas, pero omite algunas instancias de esta clase, resultando en un F1-score de 80%.

Para el plano transtalámico, como ya fue mencionado, el rendimiento es considerablemente bajo, con una precisión de 64%, pero un recall de solo 13%, lo que significa que el modelo rara vez detecta correctamente esta clase. Esto lleva a un bajo F1-score de 22%, que a su vez indica que el modelo tiene dificultades para predecir correctamente esta clase.

En contraste, la clase transventricular muestra un comportamiento opuesto, con una precisión moderada de 52%, pero un recall muy alto de 99%. Esto significa que el modelo predice correctamente

casi todas las instancias de esta clase, aunque tiene muchas predicciones falsas positivas, lo que resulta en un F1-score de 68 %.

	Precision	Recall	F1-score	Support
trans_cerebellum	0.96	0.68	0.80	66
trans_thalamic	0.64	0.13	0.22	52
trans_ventricular	0.52	0.99	0.68	67
Accuracy			0.64	185
Macro avg	0.7	0.60	0.57	185
Weighted avg	0.71	0.64	0.59	185

Cuadro 10: Reporte de clasificación para cada clase en las predicciones del Modelo ResNet

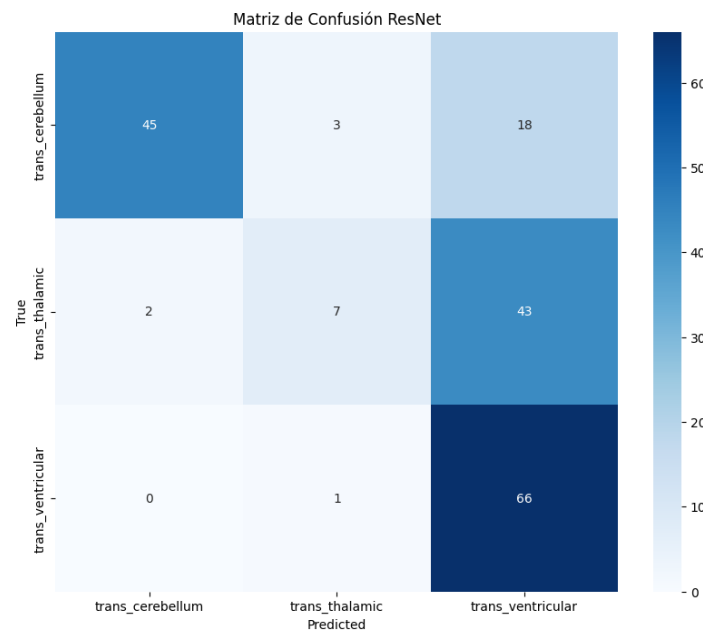


Figura 50: Matriz de confusión testeo ResNet

Se realizaron pruebas exhaustivas para investigar el bajo rendimiento de este modelo. Se llevaron a cabo 100 iteraciones de evaluación, en cada una de las cuales se evaluó el modelo con 200 imágenes, similar al proceso utilizado en la evaluación por parte de los profesionales. De estas iteraciones, 50 utilizaron 200 imágenes completamente aleatorias, mientras que las otras 50 mantuvieron la misma proporción de clases que en la evaluación general de 3658 imágenes.

En ambos casos, los resultados mostraron un rendimiento inferior al esperado, con una precisión promedio de aproximadamente 46 % en todas las iteraciones. Este bajo rendimiento podría atribuirse a la tendencia del modelo ResNet a sobreajustarse (overfitting) en conjuntos de datos pequeños [14]. Dado que el conjunto de datos original contenía solo 2853 imágenes y requirió aumentos de datos, es probable que el modelo no haya podido generalizar adecuadamente. No obstante, es importante destacar que este fue el único modelo que presentó inconvenientes en estas pruebas.

DenseNet En contraste con el modelo ResNet, el modelo DenseNet tiene un rendimiento alto y consistente en la clasificación de todas las clases, como se puede observar en el Cuadro 11.

Para la clase transcerebelo, el modelo alcanza una precisión perfecta (100%), lo que indica que todas las predicciones positivas fueron correctas. Además, su recall es muy alto (97%), lo que significa que casi todas las instancias de esta clase fueron correctamente identificadas, resultando en un F1-score de 98%.

En cuanto al plano transtalámico, el modelo muestra un rendimiento también muy sólido, con una precisión de 94% y un recall de 98%. Esto indica que el modelo es capaz de predecir correctamente la mayoría de las instancias de esta clase, alcanzando un F1-score de 96%.

La clase transventricular presenta un desempeño similar, con una precisión y un recall casi perfectos (99%), lo que da lugar a un F1-score de 99%.

A grosso modo, el modelo DenseNet logra un accuracy de 98%, lo que refleja un rendimiento casi impecable. Las métricas promedio macro y ponderadas (macro avg y weighted avg) también son consistentemente altas, con valores de precisión, recall, y F1-score de 98%. Esto indica que el modelo tiene un rendimiento equilibrado y robusto en todas las clases, con muy pocos errores de predicción.

	Precision	Recall	F1-score	Support
trans_cerebellum	1.00	0.97	0.98	66
trans_thalamic	0.94	0.98	0.96	52
trans_ventricular	0.99	0.99	0.99	67
Accuracy			0.98	185
Macro avg	0.98	0.98	0.98	185
Weighted avg	0.98	0.98	0.98	185

Cuadro 11: Reporte de clasificación para cada clase en las predicciones del Modelo DenseNet

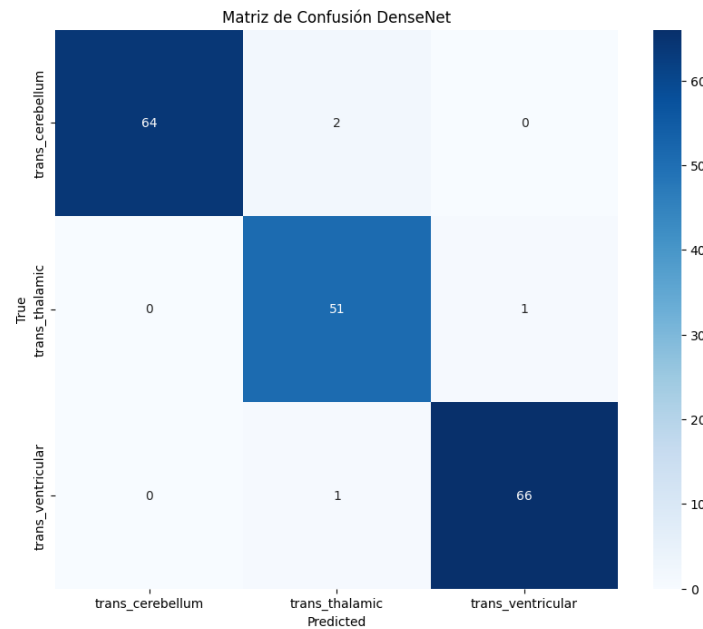


Figura 51: Matriz de confusión testeo DenseNet

YOLO Para el modelo de YOLO, podemos observar en el Cuadro 12 que tuvo un desempeño bastante prometedor para la clasificación de planos. En la Figura 52 se evidencia cómo este modelo logró superar al modelo DenseNet el cual había sido el mejor hasta el momento.

	Precision	Recall	F1-score	Support
trans_cerebellum	1.00	0.97	0.98	66
trans_thalamic	0.94	0.98	0.96	52
trans_ventricular	0.99	0.99	0.99	67
Accuracy			0.98	185
Macro avg	0.98	0.98	0.98	185
Weighted avg	0.0.98	0.98	0.98	185

Cuadro 12: Reporte de clasificación para cada clase en las predicciones del Modelo YOLO

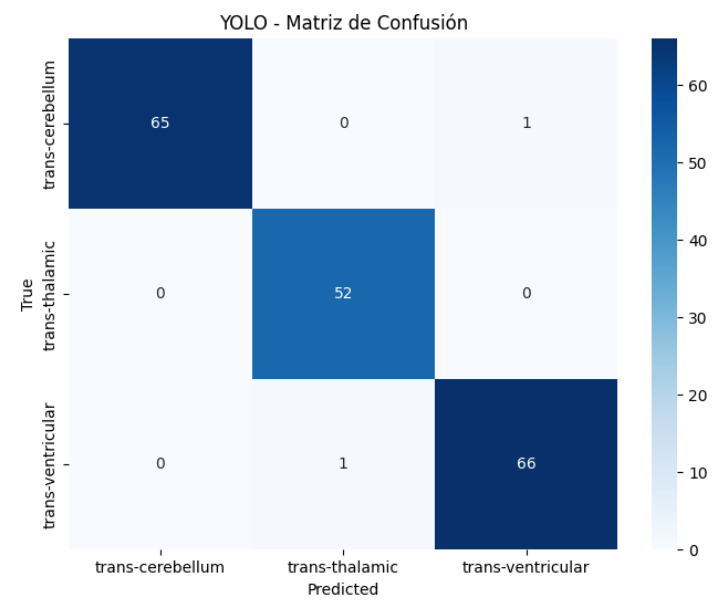


Figura 52: Matriz de confusión testeo YOLO

6.2.3. Interpretación y comparación

En el compendio de resultados, el modelo YOLO superó al resto con creces, como se ve en el Cuadro 13, esto teniendo en cuenta las primeras pruebas del modelo así como la comparación con los profesionales. Demuestra una superioridad en todas las métricas de todos los planos, por lo que es el modelo más confiable para alivianar la carga de los especialistas que deseen almacenar y clasificar este tipo de planos en poco tiempo y con una precisión bastante alta, lo que también fue comprobable en la prueba con los profesionales, en la cual no todos los modelos pudieron superarlos. A pesar de esto, YOLO ha mostrado poder clasificar exitosamente la mayor cantidad de imágenes, pese a los posibles problemas y variaciones de imágenes que señalan los profesionales.

Es importante escoger el mejor modelo para seguir adelante porque en la práctica médica puede tener implicaciones negativas cuando haya una clasificación errónea. Los falsos negativos pueden llegar a ser más o menos críticos dependiendo del uso que se le dé al modelo. Por ejemplo, en caso de usarse para clasificar en masa y soportar procesos de diagnóstico, estos pueden ser incompletos o erróneos, pues se podrían omitir detalles o estructuras importantes que pueden estar presentes en el plano real que se clasificó mal. Lo anterior también afectaría el avance de la implementación de modelos de aprendizaje automático en el campo de la salud, puesto que se perdería confiabilidad.

Detallando las gráficas de cada modelo, es notable que, a pesar de que se entrenaran con más epochs de las necesarias, no hay un indicador claro de sobreajuste, lo que se puede concluir dado que

las métricas de validación no empeoraron con el paso de las epochs, sino que se mantuvieron constantes. Esto indica que se estabilizó el aprendizaje del modelo alcanzando un equilibrio entre el desempeño en el entrenamiento y la validación.

Sin embargo, aunque no haya un indicador definitivo de sobreajuste (especialmente en los modelos ResNet y DenseNet), existen indicios que pueden sugerirlo como la alta diferencia en las métricas de loss entre el conjunto de validación y el de prueba. En las gráficas de pérdida de ResNet y DenseNet las cuales están en las Figuras 41 y 43 respectivamente, se puede observar una mejora en la métrica antes de que el modelo se estabilice, aunque es más notable en el modelo ResNet. Estos indicios y diferencias con otros modelos sumado su bajo desempeño en las pruebas con los profesionales sugieren que el modelo ResNet si tuvo un sobreajuste que los otros dos modelos no padecieron.

Class	ResNet			DenseNet			YOLO		
	Precision	F1-score	Accuracy	Precision	F1-score	Accuracy	Precision	F1-score	Accuracy
trans_cerebellum	0.96	0.96		0.96	0.97		0.98	0.98	
trans_thalamic	0.91	0.91		0.93	0.92		0.96	0.96	
trans_ventricular	0.92	0.93		0.94	0.94		0.96	0.97	
Accuracy	0.93	0.93	0.93	0.94	0.94	0.94	0.97	0.97	0.97

Cuadro 13: Informe de clasificación para los modelos ResNet, DenseNet y YOLO

En todas las pruebas, tanto en la de los profesionales como en las originales de los modelos, es evidente que el plano transcerebeloso es el que se ha podido identificar con mayor éxito. Esto se debe a que, según el doctor Juan Camilo Aguado, la estructura del cerebelo es bastante notable en la ecografía, por lo que es sencillo identificarla por su forma característica, tamaño y su ecogenicidad³.

El doctor Aguado también afirma que era de esperarse que hubiera mayor confusión entre las clases transventricular y transtálamica, puesto que son planos anatómicamente muy cercanos (véase en la Figura 1). Por lo anterior, cualquier variación en altura o angulación del plano de la ecografía puede crear cortes intermedios que no sean totalmente transtalámicos ni transventriculares. Otro factor determinante es la ganancia⁴, que en varias imágenes de la base de datos era baja, lo que afectaba la visión, en especial en el plano transtalámico.

³La ecogenicidad es la capacidad de las estructuras corporales o tejidos para reflejar ondas sonoras que han sido emitidas por ultrasonido. Lo anterior se determina producto de sus propiedades físicas, tal como su elasticidad, densidad y composición celular[42]

⁴Ganancia es la energía que se aplica al cristal piezoeléctrico y se utiliza para la compensación de la pérdida de Intensidad del ultrasonido”[52]. Es lo que permite regular el brillo de la imagen en la ecografía

7. Conclusiones

- Para hacer posible la construcción de modelos precisos fue necesario escoger la base de datos más grande posible y generar más datos a partir de la misma, para poder tener suficiente información en pro de construir modelos confiables.
- A pesar de que el proceso de la segmentación sea ligero y relativamente sencillo en comparación con los modelos de aprendizaje automático, tiene menor precisión por lo que de disponer de los recursos necesarios sería pertinente optar por estos modelos. En caso contrario, se puede optar por una solución menos robusta, más ligera y con pasos mas sencillos que pueden ser mejor entendidos por un profesional de la salud y por pacientes.
- Se desarrollaron tres modelos de aprendizaje profundo, destacando el modelo YOLO por sus métricas superiores al 96 % y su capacidad para identificar de manera precisa el área de interés en las imágenes.
- A pesar del destacado rendimiento y la capacidad de los modelos de aprendizaje automático, el procesamiento digital de imágenes sigue siendo una alternativa viable para abordar problemas como la segmentación. Este enfoque puede ofrecer resultados satisfactorios con un menor costo computacional, lo que facilita su implementación directa en equipos médicos, haciendo que sea una opción práctica y eficiente en ciertos escenarios.
- A pesar de que el modelo YOLO haya tenido su primera aparición en 2015 al igual que el modelo ResNet, ha tenido mas iteraciones y tiene una estructura mas compleja que los otros dos modelos, lo que lo hace mantenerse en la vanguardia en los campos de detección en tiempo real y clasificación.
- La clase transcerebelosa fue la que obtuvo mejores métricas por parte de la clasificación de los profesionales y de los modelos, lo que puede indicar que tiene estructuras mas claras y reconocibles que pueden facilitar su clasificación.
- En la comparación entre los modelos de aprendizaje profundo y los profesionales, se observa que los modelos pueden clasificar imágenes, incluso en casos complicados o poco claros. Sin embargo, los profesionales tienen la ventaja adicional de discernir cuando una imagen ha sido capturada de forma errónea, presenta un ángulo incorrecto, o no corresponde a un plano cerebral en general.
- En las pruebas hechas con el conjunto de imágenes que se les proporciono a los profesionales, no es tan clara la superioridad del modelo YOLO por lo que en casos reales ambos modelos pueden ser viables.

8. Trabajos Futuros

Si bien se lograron los objetivos inicialmente planteados, además de incorporar desarrollos adicionales en el proyecto, es pertinente considerar las siguientes mejoras para trabajos futuros:

- Realizar pruebas con modelos DenseNet de distintas profundidades (169, 201, 264 capas), así como con variaciones de la arquitectura ResNet (34, 50, 101, 152 capas) para evaluar la precisión y el equilibrio entre desempeño, costo computacional y tiempo de computación de cada uno.
- Aplicar técnicas de Early Stopping para optimizar el balance desempeño/epochs en el entrenamiento de los distintos modelos.
- Integrar el modelo YOLO para la clasificación en tiempo real de los planos craneales prenatales, comenzando con videos y posteriormente adaptándolo para su uso directo en un ecógrafo.
- Reunir clasificaciones de varios profesionales de la salud para obtener un promedio de precisión, acompañadas de matrices de confusión, y comparar estos resultados con los modelos propuestos.
- Adaptar tanto los modelos como el preprocesamiento de los datos para aceptar imágenes en el estándar DICOM⁵. Esto permitirá mejorar la compatibilidad del sistema con los flujos de trabajo médicos convencionales y facilitar su integración en herramientas de diagnóstico.
- Reunir más clasificaciones de ECP para hacer pruebas más robustas y aplicar técnicas que puedan funcionar mejor con mayor cantidad de datos como Cross Validation o Grid Search que evitarían un posible sobreajuste con mayor cantidad y variedad de datos recolectados.

⁵DICOM (Digital Imaging and Communications in Medicine) es el estándar internacional para el almacenamiento, transmisión e intercambio de imágenes médicas, utilizado en radiología y otras disciplinas. [16]

9. Referencias

Referencias

- [1] Corporación Universitaria Remington Andrea Gómez Perdomo. *Neurosonografía fetal*. <https://es.slideshare.net/slideshow/neurosonografia-fetal-148174167/148174167>. Presentación en Slideshare. 2019.
- [2] Amaar Arora. *DenseNet architecture explained with PyTorch implementation from TorchVision*. <https://amaarora.github.io/posts/2020-08-02-densenets.html>. Github.Io. Ago. de 2020.
- [3] B.Pranaya et al. “Image Classification With The Hybrid PVCN”. En: *2023 9th International Conference on Smart Structures and Systems (ICSSS)* (2023), págs. 1-7. DOI: [10.1109/ICSSS58085.2023.10407063](https://doi.org/10.1109/ICSSS58085.2023.10407063).
- [4] Miguel Ballén. “Detección de Alzheimer en Imágenes de Resonancia Magnética por Medio de Morfología Matemática y Computación.” En: (dic. de 2015).
- [5] David Berga, Marc Masana y Joost Weijer. *Disentanglement of Color and Shape Representations for Continual Learning*. Jul. de 2020. DOI: [10.48550/arXiv.2007.06356](https://arxiv.org/abs/2007.06356).
- [6] Jie-Zhi Cheng et al. “Computer-Aided Diagnosis with Deep Learning Architecture: Applications to Breast Lesions in US Images and Pulmonary Nodules in CT Scans”. En: *Scientific Reports* 6 (2016). DOI: [10.1038/srep24454](https://doi.org/10.1038/srep24454).
- [7] Tommaso Ciceri et al. *Review on deep learning fetal brain segmentation from Magnetic Resonance images*. Sep. de 2023. DOI: [10.1016/j.artmed.2023.102608](https://doi.org/10.1016/j.artmed.2023.102608).
- [8] Clínica Universidad de Navarra. *Ecografías*. Recuperado el 2 de septiembre de 2024, de <https://www.cun.es>. 2024. URL: <https://www.cun.es/enfermedades-tratamientos/pruebas-diagnosticas/ecografia>.
- [9] Dr. José María Muñoz Conde. *Neurosonografía*. Todos los derechos reservados. Desarrollado por iNova Cloud. Accedido en agosto de 2024. 2024. URL: <https://www.ginecologocarmona.es/neurosonografia/>.
- [10] Wikipedia contributors. *Corrección gamma*. Wikipedia, The Free Encyclopedia. s/f. URL: https://es.wikipedia.org/w/index.php?title=Correcci%C3%B3n_gamma&oldid=149939499 (visitado 10-11-2024).
- [11] Wikipedia contributors. *Ecualización del histograma*. Accedido: noviembre 2024. n.d.
- [12] Wikipedia contributors. *Plano de bits*. https://es.wikipedia.org/w/index.php?title=Plano_de_bits&oldid=158656550. Accedido: noviembre 2024. n.d.
- [13] E. El-Dahshan et al. “Computer-aided diagnosis of human brain tumor through MRI: A survey and a new algorithm”. En: *Expert Syst. Appl.* 41 (2014), págs. 5526-5545. DOI: [10.1016/J.ESWA.2014.01.021](https://doi.org/10.1016/J.ESWA.2014.01.021).
- [14] databasecamp-blog. *ResNet: Residual Neural Networks - easily explained!* <https://databasecamp.de/en/ml/resnet-en>. Recuperado el 10 de noviembre de 2024. s/f.
- [15] Coronado-Gutiérrez David et al. “Automatic Deep Learning-Based Pipeline for Automatic Delineation and Measurement of Fetal Brain Structures in Routine Mid-Trimester Ultrasound Images”. En: *Fetal Diagnosis and Therapy* 50.6 (dic. de 2023), págs. 480-490. ISSN: 14219964.
- [16] DICOM. *DICOM*. Recuperado el 6 de octubre de 2024, de <https://www.dicomstandard.org/>. n.d. URL: <https://www.dicomstandard.org/>.
- [17] Alejandro Domínguez Torres. “Procesamiento digital de imágenes”. En: *Perfiles Educativos* 72 (1996), abril-junio. ISSN: 0185-2698. URL: <http://www.redalyc.org/articulo.oa?id=13207206>.

- [18] Bradley J. Erickson y Felipe Kitamura. “Magician’s Corner: 9. Performance Metrics for Machine Learning Models”. En: *Radiology: Artificial Intelligence* 3.3 (2021), e200126. DOI: [10.1148/ryai.2021200126](https://doi.org/10.1148/ryai.2021200126). eprint: <https://doi.org/10.1148/ryai.2021200126>. URL: <https://doi.org/10.1148/ryai.2021200126>.
- [19] “Evaluation of deep convolutional neural networks for automatic classification of common maternal fetal ultrasound planes”. En: *Scientific Reports* 10 (1 dic. de 2020). ISSN: 20452322. DOI: [10.1038/s41598-020-67076-5](https://doi.org/10.1038/s41598-020-67076-5).
- [20] Marlon R. Rodrigues Garcia et al. “White blood cells segmentation and classification using a random forest and residual networks implementation”. En: 12857 (2024), págs. 128570M - 128570M-11. DOI: [10.1117/12.3007504](https://doi.org/10.1117/12.3007504).
- [21] Lavendra Gautam y Gull Kaur. “Review on Ultrasound-Image Analysis of Fetus Using Deep Learning”. En: Institute of Electrical y Electronics Engineers Inc., 2023. ISBN: 9798350334012. DOI: [10.1109/I2CT57861.2023.10126500](https://doi.org/10.1109/I2CT57861.2023.10126500).
- [22] Gimp.org. *Desenfoque gaussiano*. <https://docs.gimp.org/2.8/es/plugin-gauss.html>. Recuperado el 11 de noviembre de 2024. n.d.
- [23] Gema Valenzuela González. “Aprendizaje Supervisado: Métodos, Propiedades y Aplicaciones”. En: (2022). URL: https://riuma.uma.es/xmlui/bitstream/handle/10630/25147/TFG_Aprendizaje_Supervisado_GVG.pdf?sequence=4.
- [24] Kaiming He et al. *Deep Residual Learning for Image Recognition*. 2015. arXiv: [1512.03385](https://arxiv.org/abs/1512.03385) [cs.CV]. URL: <https://arxiv.org/abs/1512.03385>.
- [25] Kaiming He et al. “Deep Residual Learning for Image Recognition”. En: *CoRR* abs/1512.03385 (2015). arXiv: [1512.03385](https://arxiv.org/abs/1512.03385). URL: <http://arxiv.org/abs/1512.03385>.
- [26] Jinwoo Hong et al. “Fetal Cortical Plate Segmentation Using Fully Convolutional Networks With Multiple Plane Aggregation”. En: *Frontiers in Neuroscience* 14 (dic. de 2020). ISSN: 1662453X. DOI: [10.3389/fnins.2020.591683](https://doi.org/10.3389/fnins.2020.591683).
- [27] Gao Huang et al. *Densely Connected Convolutional Networks*. 2018. arXiv: [1608.06993](https://arxiv.org/abs/1608.06993) [cs.CV]. URL: <https://arxiv.org/abs/1608.06993>.
- [28] Xiaona Huang et al. “Deep Learning-Based Multiclass Brain Tissue Segmentation in Fetal MRIs”. En: *Sensors* 23.2 (2023). ISSN: 1424-8220. DOI: [10.3390/s23020655](https://doi.org/10.3390/s23020655). URL: <https://www.mdpi.com/1424-8220/23/2/655>.
- [29] Midiam Ibañez-Cuevas et al. “Atención prenatal en grupo en México: Perspectivas Y Experiencias del personal de salud”. En: *Revista de Saúde Pública* (2020). URL: <https://doi.org/10.11606/s1518-8787.2020054002175>.
- [30] IDECOR. *Procesamiento de imágenes: cómo realizar máscaras con SNAP*. Recuperado de IDECOR. Jul. de 2020. URL: <https://www.idecor.gov.ar/procesamiento-de-imagenes-como-realizar-mascaras-con-snap/>.
- [31] Zahid Farooq Khan et al. “Real-Time Polyp Detection From Endoscopic Images Using YOLOv8 With YOLO-Score Metrics for Enhanced Suitability Assessment”. En: *IEEE Access* 12 (2024), págs. 176346-176362. DOI: [10.1109/ACCESS.2024.3505619](https://doi.org/10.1109/ACCESS.2024.3505619).
- [32] Geert Litjens et al. “A survey on deep learning in medical image analysis”. En: *Medical Image Analysis* 42 (2017), págs. 60-88. ISSN: 1361-8415. DOI: <https://doi.org/10.1016/j.media.2017.07.005>. URL: <https://www.sciencedirect.com/science/article/pii/S1361841517301135>.
- [33] MathWorks. *Procesamiento de imágenes digitales*. <https://la.mathworks.com/discovery/digital-image-processing.html>. Recuperado el 10 de noviembre de 2024. s/f.
- [34] Mathworks.com. *Operaciones morfológicas*. <https://la.mathworks.com/help/images/morphological-filtering.html>. Recuperado el 11 de noviembre de 2024. n.d.

- [35] Mathworks.com. *Tipos de operaciones morfológicas - MATLAB & Simulink - MathWorks América Latina*. <https://la.mathworks.com/help/images/morphological-dilation-and-erosion.html>. Recuperado el 11 de noviembre de 2024. n.d.
- [36] MedlinePlus. *Ecografía*. 2023. URL: <https://medlineplus.gov/spanish/pruebas-de-laboratorio/ecografia/>.
- [37] B. Mehlig. “Machine learning with neural networks”. En: (2021). DOI: [10.1017/9781108860604](https://doi.org/10.1017/9781108860604). URL: <http://arxiv.org/abs/1901.05639><http://dx.doi.org/10.1017/9781108860604>.
- [38] Matuk Leonardo Michelone. *Programación lúdica: ecualización del histograma de una imagen*. Accedido el 10 de noviembre de 2024. unocero. Oct. de 2017. URL: <https://www.unocero.com/entretenimiento/programacion-ludica-ecualizacion-del-histograma-de-una-imagen/>.
- [39] Restrepo Miranda et al. “Aproximaciones a la estimación de la oferta y la demanda de médicos especialistas en Colombia, 2015 - 2030”. Colombia, 2017. URL: <https://www.minsalud.gov.co/sites/rid/Lists/BibliotecaDigital/RIDE/VS/TH/Especialistas-md-oths.pdf>.
- [40] S. Molina-Giraldo et al. “Anatomical structure characterization of fetal ultrasound images using texture-based segmentation technique via an interactive MATLAB application”. En: *Journal of Clinical Ultrasound* 52.2 (feb. de 2024). Epub 2023 Nov 23, págs. 189-200. DOI: [10.1002/jcu.23604](https://doi.org/10.1002/jcu.23604).
- [41] National Institute of Biomedical Imaging and Bioengineering. *Ultrasonido*. Recuperado el 2 de septiembre de 2024, de <https://www.nibib.nih.gov>. 2024. URL: <https://www.nibib.nih.gov/espanol/temas-cientificos/ultrasonido>.
- [42] Clínica U. Navarra. *Qué es ecogénico. Diccionario médico*. <https://www.cun.es/diccionario-medico/terminos/ecogenico>. Recuperado el 18 de diciembre de 2024, de <https://www.cun.es/s/f>. URL: <https://www.cun.es/diccionario-medico/terminos/ecogenico>.
- [43] J. Nelson. *You might be resizing your images incorrectly*. Roboflow Blog. Ene. de 2020. URL: <https://blog.roboflow.com/you-might-be-resizing-your-images-incorrectly/>.
- [44] Alejandro Padrón-Godínez et al. “OCULTAMIENTO DE DATOS EN IMÁGENES DIGITALES MEDIANTE BPCS”. En: oct. de 2008.
- [45] D. Paladini et al. “Sonographic examination of the fetal central nervous system: Guidelines for performing the “basic examination” and the “fetal neurosonogram””. En: *Ultrasound in Obstetrics and Gynecology* 29.1 (2007), págs. 109-116. DOI: [10.1002/uog.3909](https://doi.org/10.1002/uog.3909). URL: <https://doi.org/10.1002/uog.3909>.
- [46] Joseph Redmon et al. “You Only Look Once: Unified, Real-Time Object Detection”. En: *CoRR* abs/1506.02640 (2015). arXiv: [1506.02640](https://arxiv.org/abs/1506.02640). URL: <http://arxiv.org/abs/1506.02640>.
- [47] La República. *Actualmente en Colombia solo hay un médico especialista por cada 1.000 habitantes*. online. 2023. URL: <https://www.larepublica.co/economia/actualmente-en-colombia-solo-hay-un-medico-especialista-por-cada-1-000-habitantes-3609487>.
- [48] “Resolución Número 3280 de 2018”. En: *Ministerio de Salud y Protección Social de Colombia* (2018). URL: <https://www.minsalud.gov.co/sites/rid/Lists/BibliotecaDigital/RIDE/DE/DIJ/resolucion-3280-de-2018.pdf>.
- [49] “Review of deep learning and artificial intelligence models in fetal brain magnetic resonance imaging”. En: *World Journal of Clinical Cases* 11 (16 jun. de 2023), págs. 3725-3735. ISSN: 2307-8960. DOI: [10.12998/wjcc.v11.i16.3725](https://doi.org/10.12998/wjcc.v11.i16.3725).
- [50] M. P. Rojas. *Sinaptogénesis: ¿Cómo conectamos las neuronas?* Consultado el 12 de diciembre de 2024. Feb. de 2020. URL: <https://neuro-class.com/sinaptogenesis-como-conectamos-las-neuronas/>.
- [51] Vanessa Santos, Gisele Devetak y Elisangela Manffra. “REPEATABILITY OF GAIT RANGES OF MOTION IN THE PRESENCE OF STROKE”. En: jun. de 2019, págs. 94-106. ISBN: 9788572473828. DOI: [10.22533/at.ed.82819010610](https://doi.org/10.22533/at.ed.82819010610).

- [52] A. Segura-Grau et al. “Curso de ecografía abdominal. Introducción a la técnica ecográfica. Principios físicos. Lenguaje ecográfico”. En: *Semergen* 40 (1 ene. de 2014), págs. 42-46. ISSN: 11383593. DOI: [10.1016/j.semerng.2013.09.008](https://doi.org/10.1016/j.semerng.2013.09.008).
- [53] Sudhakar Sengan et al. “Echocardiographic Image Segmentation for Diagnosing Fetal Cardiac Rhabdomyoma During Pregnancy Using Deep Learning”. En: *IEEE Access* 10 (2022), págs. 114077-114091. DOI: [10.1109/ACCESS.2022.3215973](https://doi.org/10.1109/ACCESS.2022.3215973).
- [54] Huitong Song. “Bird image classification based on improved ResNet-152 image classification model”. En: *Applied and Computational Engineering* (2024). DOI: [10.54254/2755-2721/54/20241530](https://doi.org/10.54254/2755-2721/54/20241530).
- [55] A. Sosa-Costa. *Ecualización de histogramas*. Cursos de Programación de 0 a Experto © Garantizados. Oct. de 2017. URL: <https://unipython.com/ecualizacion-de-histogramas/>.
- [56] Richard Szeliski. *Computer Vision: Algorithms and Applications 2nd Edition*. 2021. URL: <https://szeliski.org/Book,.>
- [57] Akash Takyar. *ENSURING ML MODEL ACCURACY AND ADAPTABILITY THROUGH MODEL VALIDATION TECHNIQUES*. 2023. URL: <https://www.leewayhertz.com/model-validation-in-machine-learning/>.
- [58] R. S. Tiwari. “Transfer Learning - Part -7.0!! Dense net - becoming human: Artificial intelligence magazine”. En: *Becoming Human: Artificial Intelligence Magazine* (jul. de 2022). URL: <https://medium.com/becoming-human/transfer-learning-part-7-0-dense-net-bd23d9f6d4fd>.
- [59] Ultralytics. *Inicio*. Ultralytics.com. Nov. de 2023. URL: <https://docs.ultralytics.com/es>.
- [60] Rohan Varma. (2017) *Picking Loss Functions - A comparison between MSE, Cross Entropy, and Hinge Loss*. URL: <https://rohanvarma.io/Loss-Functions/>.
- [61] H. N. Xie et al. “Using deep-learning algorithms to classify fetal brain ultrasound images as normal or abnormal”. En: *Ultrasound in Obstetrics and Gynecology* 56 (4 oct. de 2020), págs. 579-587. ISSN: 14690705. DOI: [10.1002/uog.21967](https://doi.org/10.1002/uog.21967).
- [62] Suorong Yang et al. *Image Data Augmentation for Deep Learning: A Survey*. 2023. arXiv: [2204.08610](https://arxiv.org/abs/2204.08610) [cs.CV]. URL: <https://arxiv.org/abs/2204.08610>.

10. Anexos

En la Figura 53 se muestra la distribución de precisión de las 50 pruebas que se hicieron para el modelo ResNet con subconjuntos de 200 imágenes aleatorias.

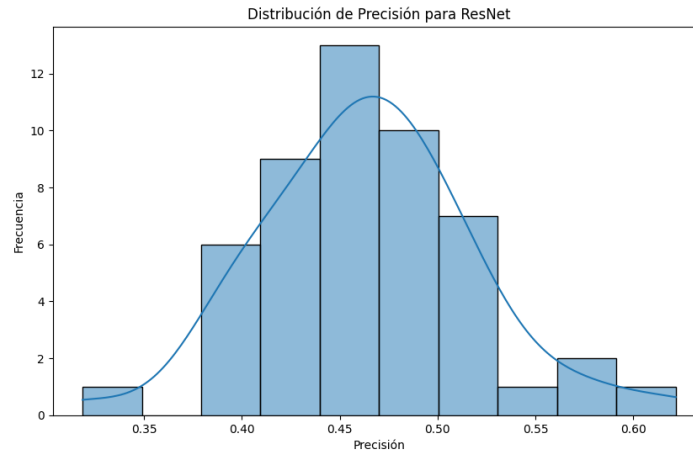


Figura 53: Distribución de precisión de las pruebas con ResNet con subconjuntos de 200 imágenes aleatorias

En la Figura 54 se muestra la distribución de precisión de las 50 pruebas que se hicieron para el modelo ResNet con subconjuntos de 200 imágenes balanceadas entre clases.

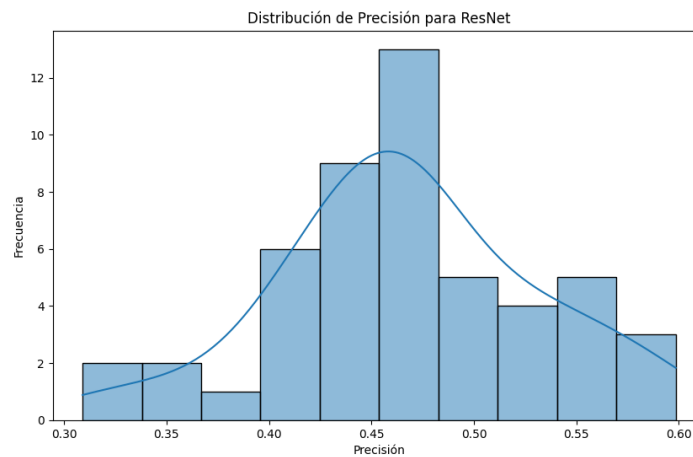


Figura 54: Distribución de precisión de las pruebas con ResNet con subconjuntos de 200 imágenes balanceadas

En el Cuadro 14 se muestran las distintas métricas sacadas para YOLO por cada clase.

Class	Images	Instances	Precision	Recall	mAP50	mAP50-95
all	3658	3658	0.966	0.972	0.989	0.949
Trans-cerebellum	1244	1244	0.980	0.984	0.992	0.948
Trans-thalamic	1310	1310	0.958	0.954	0.986	0.947
Trans-ventricular	1104	1104	0.960	0.979	0.988	0.953

Cuadro 14: Métricas de desempeño para cada clase en el modelo YOLO