Pontificia Universidad JAVERIANA Cali

Facultad de Ingeniería y Ciencias

Acta de Correcciones al Proyecto de Grado
Ingeniería de Sistemas y Computación

Fecha: 20 de julio, 2024

Autores: Jhoan Manuel Buitrago Chavez, Juan Felipe González Vélez

Nombre del Proyecto de Grado: Modelo para la detección de cáncer de seno en imágenes histológicas a partir de aprendizaje profundo con múltiples anotadores

Director: Dr. Julián Gil González

Como indica el artículo 2.27 de las Directrices de Trabajo de Grado, he verificado que los estudiantes indicados arriba han implementado todas las correcciones que los Jurados del Proyecto de Grado definieron que se efectuaran, como consta en el Acta de Calificación correspondiente.

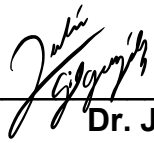
Firma de Director(a) del Proyecto de Grado

Nota de Aceptación

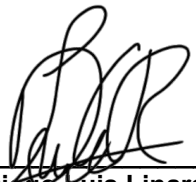
Aprobado por el Comité de Trabajo de Grado en cumplimiento de los requisitos exigidos por la Pontificia Universidad Javeriana para optar el título de Ingeniero de Sistemas y Computación.

Dr. Hernan Camilo Rocha Niño
Decano de la Facultad de Ingeniería

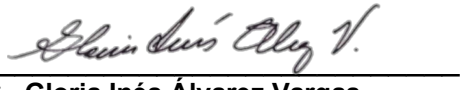
Dr. Gerardo Mauricio Sarria Montemiranda
Director Carrera Ingeniería Sistemas y Computación.



Dr. Julián Gil González
Director(a) Trabajo



ING. Diego Luis Linares Ospina
Jurado 1



ING. Gloria Inés Álvarez Vargas
Jurado 2

Modelo para la detección de cáncer de seno en imágenes histológicas a partir de aprendizaje profundo con múltiples anotadores

Juan Felipe González Vélez
Jhoan Manuel Buitrago Chávez

Pontificia Universidad Javeriana Cali
Facultad de Ingeniería
Departamento de Ingeniería de Sistemas y Computación
Cali
2024

Modelo para la detección de cáncer de seno en imágenes histológicas a partir de aprendizaje profundo con múltiples anotadores

Juan Felipe González Vélez
Jhoan Manuel Buitrago Chávez

Trabajo de Grado

Director:
Julián Gil González

Pontificia Universidad Javeriana Cali
Facultad de Ingeniería
Departamento de Ingeniería de Sistemas y Computación
Cali
2024

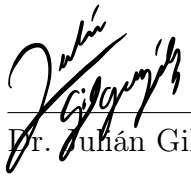
Santiago de Cali, 4 de junio, 2024.

Señores
Pontificia Universidad Javeriana, Cali.
Gerardo Mauricio Sarria Montemiranda
Director de Carrera de Ingeniería de Sistemas y Computación.
Cali.

Cordial Saludo.

Por medio de la presente me permito informarle que los estudiantes de Ingeniería de Sistemas y Computación Jhoan Manuel Buitrago Chávez (cod: 8953846) y Juan Felipe González Vélez (cod: 8947085) trabajaron bajo mi dirección en el proyecto de grado titulado "Modelo para la detección de cáncer de seno en imágenes histológicas a partir de aprendizaje profundo con múltiples anotadores".

Atentamente,



Dr. Julián Gil González

Santiago de Cali, 4 de junio, 2024.

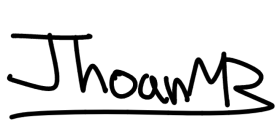
Señores
Pontificia Universidad Javeriana, Cali.
Gerardo Mauricio Sarria Montemiranda
Director de Carrera de Ingeniería de Sistemas y Computación.
Cali.

Cordial Saludo.

Nos permitimos presentar a su consideración el proyecto de grado titulado "Modelo para la detección de cáncer de seno en imágenes histológicas a partir de aprendizaje profundo con múltiples anotadores", con el fin de cumplir con los requisitos exigidos por la universidad para optar al título de Ingeniero de Sistemas y Computación.

Al firmar aquí, damos fe que entendemos y conocemos las directrices para la presentación de trabajos de grado de la Facultad de Ingeniería 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,



Jhoan Manuel Buitrago Chávez
Código: 8953846

Juan Felipe González Vélez
Código: 8947085

Índice general

Resumen **7**

Summary **8**

Introducción **9**

1. Descripción del problema **11**
1.1. Planteamiento del Problema . . . 11
1.1.1. Formulación . . . 13
1.1.2. Sistematización . . . 13
1.2. Objetivos . . . 13
1.2.1. Objetivo General . . . 13
1.2.2. Objetivos Específicos . . . 14
1.3. Justificación . . . 14

2. Marco Teórico **16**
2.1. Cáncer de seno . . . 16
2.2. Imágenes histológicas . . . 17
2.3. Aprendizaje supervisado . . . 17
2.3.1. Aprendizaje con un único anotador . . . 18
2.3.2. Aprendizaje con múltiples anotadores . . . 18
2.4. Redes neuronales . . . 19
2.5. Aprendizaje profundo . . . 21
2.6. Red neuronal convolucional (CNN) . . . 22
2.6.1. VGG16 . . . 23
2.7. Funciones de costo . . . 24
2.7.1. Cross Entropy . . . 24
2.7.2. Generalized Categorical Cross Entropy . . . 25
2.7.3. Focal Loss . . . 26

3. Metodología **28**
3.1. Obtención del conjunto de datos . . . 28
3.2. Preprocesamiento de las imágenes histológicas . . . 29
3.2.1. Normalización de color . . . 30
3.2.2. Segmentación de las imágenes . . . 30
3.3. Análisis de los datos . . . 31

3.4. Extracción de características 34
3.5. Entrenamiento de los modelos 34
3.5.1. Gold standard 36
3.5.2. Múltiples anotadores 37
3.6. Criterios de evaluación 42

4. Resultados **44**
4.1. Rendimiento de los modelos 45
4.1.1. Gold standard 45
4.1.2. Múltiples anotadores 46
4.2. Comparación entre modelos 51

5. Conclusiones **56**
5.1. Trabajo futuro 58

Índice de figuras

1.1. Tipos de cáncer más comunes en mujeres (2020), tomado de [1] . . . 11

2.1. Imagen histológica de seno [2]. 17
2.2. Ejemplo de una red neuronal básica que consta de una neurona de entrada, tres neuronas en la capa oculta y una neurona de salida. . . 19
2.3. Ejemplo de una red neuronal profunda. 22
2.4. Visualización de una arquitectura típica de una CNN. [3] 23
2.5. Comportamiento de la perdida focal con diferentes γ. Tomado de [4]. 27

3.1. Proceso de crowdsourcing estructurado planteado en el artículo *Structured crowdsourcing enables convolutional segmentation of histology images* [2]. 29
3.2. Normalización de color de Reinhard de las imagenes 30
3.3. Estructura del conjunto de datos para el entrenamiento. 31
3.4. Flujo de extracción y utilización de características. 34
3.5. Arquitectura de la red neuronal base usada en los modelos del proyecto. Consta de una capa de entrada de 512 neuronas, dos capas ocultas de 128 y 64 neuronas, cada una con una capa de dropout de 0.25 y por último la capa de salida de 3 neuronas para las tres clases (1=cáncer, 2=estroma y 3=inflamación benigna). 35
3.6. Arquitectura de la red neuronal para *generalized categorical cross-entropy.* 40
3.7. Arquitectura de la red neuronal usada en los modelos *Crowdlayer.* . . 41

Índice de cuadros

3.1. Distribución de imágenes por clase. 32
3.2. Cantidad de anotaciones por anotador 32
3.3. Distribución de las anotaciones 33
3.4. Distribución de imágenes por clase con más de un anotador. 33
3.5. Distribución de imágenes por clase con *majority voting*. 38

4.1. Resultado del modelo gold standard con la función de costo cross-entropy 45
4.2. Resultados del modelo gold standard con la función de costo focal loss usando el parámetro α. 45
4.3. Resultado del modelo gold standard con la función de costo focal loss sin usar el parámetro α 46
4.4. Resultados del modelo con *majority voting* usando la función de costo *cross-entropy* 46
4.5. Resultado de modelos con *majority voting* con la función de costo *focal-loss* usando el parámetro α 47
4.6. Resultado de modelos con *majority voting* con la función de costo *focal-loss* sin usar el parámetro α 48
4.7. Resultados de modelos con *generalized categorical cross entropy*(GCCE) usando todo el conjunto de datos. 48
4.8. Resultados de modelos con *generalized categorical cross entropy*(GCCE) usando los datos que fueron anotados por dos o más personas. 49
4.9. Resultados de modelos con *Crowdlayer* usando todo el conjunto de datos. 50
4.10. Resultados de modelos con *Crowdlayer* usando los datos que fueron anotados por dos o más personas. 50

Resumen

El cáncer de seno es el tipo de cáncer más común en mujeres, por lo que la detección temprana de esta enfermedad es crucial para combatirla. Actualmente, una de las formas de detectar el cáncer de seno es mediante el análisis del tejido mamario a través de imágenes histológicas. Este análisis es un proceso tedioso que debe ser realizado por un experto. El aprendizaje automático puede ser útil para facilitar esta tarea; sin embargo, requiere una cantidad suficiente de información recolectada para su entrenamiento, lo cual es difícil de obtener debido a la escasez de expertos capaces de anotar las imágenes.

En este proyecto se proponen diferentes modelos de aprendizaje profundo para la clasificación de imágenes histológicas, aprovechando las anotaciones de múltiples anotadores con diversos grados de experiencia y conocimiento. Los resultados de estos modelos fueron comparados con varios modelos clásicos de aprendizaje automático que utilizan las etiquetas verdaderas para su entrenamiento.

Palabras clave: aprendizaje de máquina, aprendizaje profundo, múltiples anotadores, clasificación, cáncer de seno, imágenes, histología.

Summary

Breast cancer is the most common type of cancer in women, making early detection of this disease crucial for combating it. Currently, one of the ways to detect breast cancer is through the analysis of breast tissue via histological images. This analysis is a tedious process that must be performed by an expert. Machine learning can be useful to facilitate this task; however, it requires a sufficient amount of collected information for training, which is difficult to obtain due to the scarcity of experts capable of annotating the images.

In this project, different deep learning models are proposed for the classification of histological images, leveraging the annotations from multiple annotators with varying degrees of experience and knowledge. The results of these models were compared with several classical machine learning models that use true labels for their training.

Keywords: machine learning, deep learning, multiple annotators, classification, breast cancer, images, histology.

Introducción

El cáncer es una de las enfermedades con mayores tasas de mortalidad a nivel mundial[5]. En especial, el cáncer de seno, que es el más común entre las mujeres, es una de las variantes que más afecta a la población mundial[6]. La detección temprana del cáncer de seno es fundamental para aumentar las posibilidades de recuperación de las mujeres afectadas, por lo que es indispensable contar con técnicas apropiadas para su detección. Comúnmente, la detección temprana del cáncer de seno implica la realización de diversas pruebas diagnósticas. Sin embargo, estas pruebas pueden no ser lo suficientemente sensibles para detectar lesiones tempranas o pequeñas, lo que puede resultar en un diagnóstico tardío [7][8]. Es aquí donde la tecnología, utilizando métodos de aprendizaje profundo, ofrece una herramienta de gran ayuda para la detección temprana del cáncer de seno.

El aprendizaje profundo se presenta como una solución prometedora para el problema de la detección del cáncer de seno en imágenes histológicas. Estos métodos pueden automatizar y mejorar la precisión del análisis, permitiendo detectar características sutiles que podrían pasar desapercibidas en las evaluaciones tradicionales. Sin embargo, a veces es difícil extraer información relevante, ya que se requiere de expertos que hagan las anotaciones de las imágenes histológicas para entrenar los modelos de aprendizaje profundo. La escasez de estos expertos y la complejidad del proceso de anotación pueden limitar la cantidad de datos de alta calidad disponibles, lo que representa un desafío significativo para el desarrollo y la efectividad de estas tecnologías avanzadas.

Los datos utilizados en este proyecto provienen de un proceso de "crowdsourcing" estructurado, que permitió la colaboración de personas con diferentes niveles de conocimiento en patología para la anotación de imágenes histológicas. Las imágenes fueron tomadas de The Cancer Genome Atlas (TCGA)[9] y representan diversas muestras de casos confirmados de cáncer de seno. Con la colaboración de patólogos senior, se seleccionaron 151 regiones de interés para asegurar la representación de las principales clases de tejidos observados en el cáncer de seno. Luego de un proceso de preprocesamiento, estos datos se transformaron en 75,243 imágenes para el entrenamiento de los modelos.

En este proyecto se proponen diferentes modelos de aprendizaje automático que aprovechan las anotaciones de múltiples personas para recopilar información y generar predicciones. Estos modelos serán comparados con el "gold standard", un modelo tradicional que utiliza las etiquetas verdaderas. Los modelos desarrollados

en este proyecto utilizan las características de las imágenes histológicas extraídas mediante "transfer learning", utilizando una red VGG16. Este enfoque permite mejorar la precisión y eficiencia del análisis al transferir el conocimiento adquirido en grandes bases de datos a las imágenes histológicas específicas del cáncer de seno.

La estructura del proyecto se organiza en cinco capítulos. El capítulo 1 presenta la descripción del problema y los objetivos del proyecto. El capítulo 2 aborda el marco teórico, explicando los diferentes conceptos necesarios para el entendimiento del proyecto. En el capítulo 3 se detalla la metodología del proyecto, que abarca desde la obtención del conjunto de datos inicial y el preprocesamiento de los datos hasta el entrenamiento de los modelos. El capítulo 4 muestra los resultados obtenidos en el desarrollo del proyecto. Finalmente, el capítulo 5 ofrece las conclusiones y sugiere posibles trabajos futuros.

Capítulo 1

Descripción del problema

1.1. Planteamiento del Problema

El cáncer de seno es el tipo de cáncer más común en mujeres[5], junto con el cáncer de pulmón, colorrectal y de útero [6]. Es una de las principales causas de muerte a nivel mundial. En el año 2020, se registraron 1 948 321 casos de cáncer de seno, con una tasa de mortalidad del 26.35 %[1]. Debido a esto, se recomienda que las mujeres se realicen pruebas con relativa frecuencia, especialmente a partir de los 40 años, cada 1 a 3 años, para detectar la enfermedad a tiempo antes de que avancé a una etapa avanzada[10].

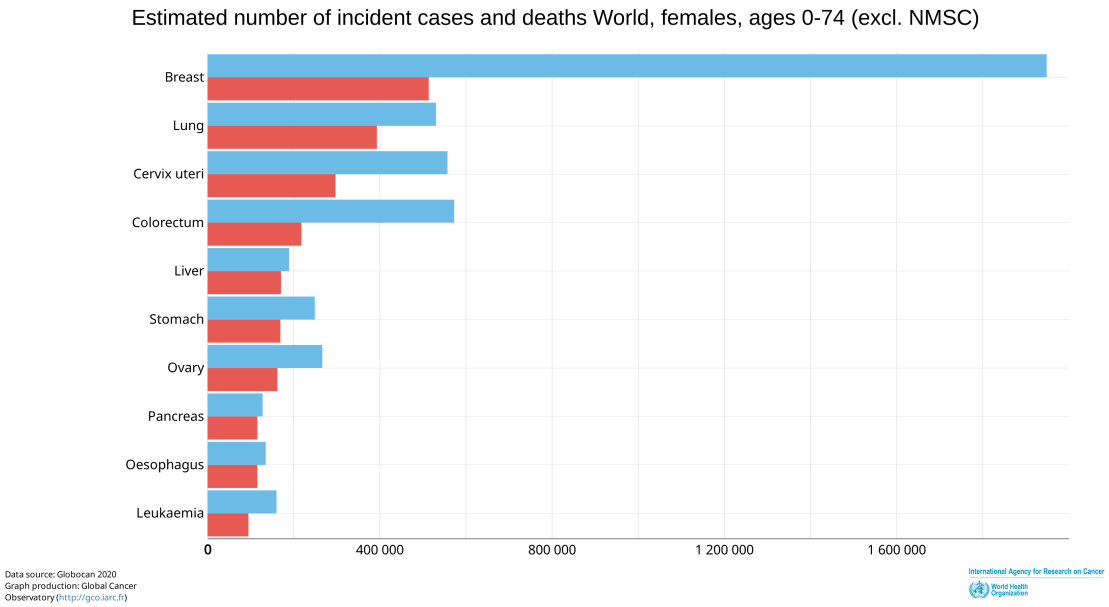


Figura 1.1: Tipos de cáncer más comunes en mujeres (2020), tomado de [1]

La detección temprana de esta enfermedad es crucial para su tratamiento y prevención, puesto que en etapas tempranas esta patología suele ser más pequeña y concentrada en ciertas zonas del cuerpo, lo que da acceso a tratamientos más efectivos y mejores probabilidades de éxito en la pelea contra la enfermedad[10]. Detectar cáncer en etapas avanzadas suele conllevar a tratamientos más extensivos que producen un aumento en la morbilidad del paciente, además de un aumento en el gasto económico.

Existen múltiples métodos para la detección del cáncer de seno, entre ellos está la mamografía, qué es una imagen de rayos X de los senos. Las mamografías son usualmente usadas como una prueba de cribado, es decir, que se realiza cuando el paciente no presenta síntomas con el fin de detectar la enfermedad en las etapas más tempranas[11]. Esta prueba puede presentar falsos positivos, por lo que es necesario que se acompañe de otras pruebas como ultrasonido, las cuales permiten diferenciar entre líquidos y masas sólidas[12]. Si los resultados de estas pruebas sugieren la presencia de cáncer de seno, puede que se necesite de una biopsia, la cual consiste de extraer pequeñas muestras de tejido de las áreas sospechosas del seno. De los tejidos extraídos se obtienen imágenes histológicas, las cuales son analizadas por un experto, el cual determina si hay presencia de células cancerígenas.

El análisis manual de imágenes histológicas ha sido históricamente un proceso basado en la experiencia, lo que puede dificultar su objetividad. La eficacia de estos análisis depende en gran medida de la experiencia y capacitación del patólogo o persona encargada, lo cual puede generar variaciones en los diagnósticos entre diferentes individuos. Estas variaciones conducen a una evaluación inexacta y pueden ocasionar retrasos en el tratamiento[13]. Es por lo anterior que el uso de la inteligencia artificial puede ayudar en el diagnóstico del cáncer.

El aprendizaje profundo ha demostrado ser una herramienta valiosa en la detección y clasificación de patologías [14]. Un ejemplo relevante sería [15] donde se hace uso de redes convoluciones para el procesamiento de las imágenes junto con un modelo de redes neuronales combinado con procesos gausianos para la detección de cáncer de seno en imágenes histológicas usando múltiples anotadores. Por otro lado, en el caso de un modelo que no usa aprendizaje profundo para resolver una tarea similar, está [16] que usa un modelo de clustering llamado K-means y segmentación de imágenes para detectar el cáncer de seno en imágenes histológicas. sin embargo, la implementación de un modelo de aprendizaje profundo presenta varios retos.

Una de las mayores complicaciones es la necesidad de un gran volumen de datos, donde cada uno de estos datos debe estar anotado. Además, el rendimiento del modelo entrenado depende en gran medida de las anotaciones utilizadas para entrenar el modelo, es decir, se requieren anotaciones precisas que permitan que el modelo identifique con mayor precisión el cáncer de seno, pero conseguir estas anotaciones y determinar que sean precisas es aún más complejo, puesto que incluso las anotaciones de expertos en el tema pueden tener cierto nivel de subjetividad y, por lo tanto, limitar la precisión del modelo.

Para ayudar a facilitar la implementación de modelos de aprendizaje profundo se ha usado la colaboración colectiva o múltiples anotadores, donde varias personas

con diferentes niveles de conocimiento ayudan a la anotación de los datos de los modelos [15]. Esto tiene la ventaja de agilizar el proceso de anotación y dar una mayor cantidad de opiniones en el proceso de clasificación, lo que ayuda a reducir la subjetividad del proceso de selección. No obstante, se presenta un nuevo problema, porque ahora es necesario manejar la opinión de múltiples personas, donde cada una tiene su propio prejuicio, en vez de confiar en una única etiqueta o anotación. Cómo manejar adecuadamente el ruido que introducen los nuevos anotadores al sistema es todavía un área activa de investigación.

Por lo tanto, el objetivo de este trabajo es entrenar un modelo de aprendizaje profundo para la detección de cáncer de seno utilizando múltiples anotadores, explorando diferentes maneras de manejar el ruido en las anotaciones y buscando mejorar la precisión del modelo y reducir la subjetividad en la detección de la enfermedad. Este enfoque busca ayudar en la detección del cáncer de seno y, por ende, mejorar la calidad de vida de las pacientes afectadas por esta enfermedad.

1.1.1. Formulación

¿Cómo construir un modelo de aprendizaje profundo basado en múltiples anotadores para la detección de cáncer de seno en imágenes histológicas?

1.1.2. Sistematización

- ¿Cómo implementar una etapa de preparación del conjunto de datos a partir de técnicas de procesamiento de datos con el fin de que pueda ser utilizado adecuadamente por los modelos de aprendizaje profundo?
- ¿Cómo implementar un modelo de aprendizaje profundo que tenga en cuenta múltiples anotadores para la clasificación de imágenes histológicas?
- ¿Cómo implementar un proceso de búsqueda de hiperparámetros para el modelo desarrollado?
- ¿Cómo validar los resultados del modelo de aprendizaje profundo con múltiples anotadores?

1.2. Objetivos

1.2.1. Objetivo General

- Construir un modelo de aprendizaje profundo basado en múltiples anotadores para la detección de cáncer de seno en imágenes histológicas.

1.2.2. Objetivos Específicos

- Implementar una etapa de preparación del conjunto de datos a partir de técnicas de procesamiento de datos, con el fin de que pueda ser utilizado adecuadamente por los modelos de aprendizaje profundo para la detección de cáncer de seno en imágenes histológicas.
- Implementar un modelo de aprendizaje profundo que tenga en cuenta múltiples anotadores para la clasificación de imágenes histológicas.
- Implementar un proceso de búsqueda de hiperparámetros para el modelo desarrollado.
- Validar los resultados del modelo de aprendizaje profundo con múltiples anotadores.

1.3. Justificación

La detección temprana del cáncer de seno es una tarea crucial que puede mejorar mucho la probabilidad de supervivencia de las personas que padecen esa enfermedad. Los métodos que existen en la actualidad para la detección, como el ultrasonido mamario y la biopsia, pueden carecer de eficiencia y precisión para detectar el cáncer en etapas tempranas[7].

En este contexto, el uso de técnicas de aprendizaje profundo y múltiples anotadores se presenta como una herramienta prometedora para mejorar la eficiencia y precisión del diagnóstico de cáncer de seno en imágenes histológicas. La combinación de las anotaciones de múltiples anotadores con diferentes grados de conocimiento y experiencia en la interpretación de estas imágenes puede mejorar la calidad y la fiabilidad de las anotaciones y, por lo tanto, aumentar la calidad de los modelos de aprendizaje profundo entrenados con ellas.

El uso de técnicas de aprendizaje profundo también puede permitir una mayor automatización y eficiencia en la detección temprana del cáncer de seno, lo que podría mejorar la disponibilidad y accesibilidad de los métodos de diagnóstico para las mujeres en todo el mundo.

Además de su importancia clínica, este proyecto puede tener un impacto positivo en el ámbito académico. Modelos de este tipo pueden servir como una herramienta de aprendizaje y entrenamiento para futuros profesionales de la salud, y puede ayudar a mejorar la calidad y eficiencia del diagnóstico de cáncer de seno en la práctica clínica.

En cuanto a la viabilidad del proyecto propuesto, existen variedad de técnicas y modelos usados en la detección del cáncer de seno. En (agregar nombre artículo?) [16] se encuentra un ejemplo de un modelo clásico de aprendizaje no supervisado, con la técnica de clustering y el modelo de K-means junto con segmentación de imágenes para detección automática de cáncer de seno en imágenes histológicas. Por

otra parte, y más orientado al ámbito de los múltiples anotadores y de aprender de distintos niveles de conocimiento, existen diferentes trabajos interesantes que usan modelos de aprendizaje supervisado con datos etiquetados por múltiples personas para la búsqueda y detección de cáncer de seno en imágenes histológicas como son [15] [17] [2]. Adicional a esto, existen distintos repositorios de información donde se almacena información relevante a los diferentes tipos de cáncer, entre ellos el cáncer de seno, llamado The Cancer Genome Atlas (TCGA) disponible en Genomic Data Commons, Data Portal[9].

En resumen, este proyecto se propone abordar un problema crítico en el diagnóstico de cáncer de seno y explorar el potencial de las técnicas de aprendizaje profundo y múltiples anotadores para mejorar la precisión y eficiencia del diagnóstico de esta enfermedad. Los resultados de este proyecto pueden tener un impacto significativo en el diagnóstico del cáncer de seno y ayudar tanto a profesionales como estudiantes en entrenamiento a identificar esta enfermedad en imágenes histológicas con mayor facilidad.

Capítulo 2

Marco Teórico

2.1. Cáncer de seno

El cáncer de seno, como su nombre sugiere, es un tipo de cáncer que se origina en los senos. Aunque es más común en las mujeres, también puede afectar a los hombres[18]. El cáncer de seno es uno de los tipos más frecuentes en mujeres, junto con el cáncer de pulmón, colorrectal y de útero[6], y constituye una de las principales causas de mortalidad a nivel mundial. Según estimaciones de la Sociedad Americana del Cáncer, en los Estados Unidos se diagnosticaron aproximadamente 297,790 nuevos casos de cáncer de seno invasivo en mujeres para el año 2023, con un triste pronóstico de alrededor de 47,300 fallecimientos debido a esta enfermedad durante este año[19].

Teniendo en cuenta lo mencionado previamente, la detección temprana del cáncer y la pronta iniciación del tratamiento representan dos de las estrategias más cruciales para prevenir fallecimientos a causa de esta enfermedad[11]. En las etapas iniciales, el cáncer tiende a ser de menor tamaño y focalizado en áreas específicas, lo que contribuye a que el tratamiento sea más efectivo y menos complejo[10]. Por el contrario, la detección del cáncer en etapas más avanzadas conlleva tratamientos más invasivos y extensos, lo que incrementa considerablemente la tasa de mortalidad de los pacientes.

Existen varios métodos para realizar un diagnóstico temprano del cáncer de seno, y uno de los más utilizados es la mamografía, que proporciona una imagen en rayos X del seno. Esta prueba se realiza típicamente como parte de un enfoque preventivo, y no es necesario que el paciente presente síntomas de cáncer de seno para someterse a ella[11]. Es importante mencionar que la mamografía puede arrojar resultados falsos positivos, por lo que generalmente se combina con otras pruebas, como el ultrasonido, que ayuda a distinguir entre líquidos y masas sólidas en el seno[12]. Si se detecta alguna anomalía en estas pruebas, es posible que se requiera una biopsia, un procedimiento que implica la extracción de una pequeña muestra de tejido para su análisis bajo un microscopio. Este análisis genera imágenes histológicas que son evaluadas por un especialista, quien determinará si hay presencia de células cancerosas.

2.2. Imágenes histológicas

Cuando nos referimos a la histología en el contexto del cuerpo humano, estamos hablando de la investigación de la estructura microscópica de los diversos componentes biológicos que lo componen. También exploramos cómo estas estructuras se relacionan entre sí para asegurar que el cuerpo funcione correctamente [20].

Las imágenes histológicas son representaciones visuales de secciones finas de tejidos biológicos observados a través de un microscopio. Estos tejidos suelen obtenerse de biopsias, autopsias u otras fuentes. En las imágenes histológicas, es posible identificar alteraciones en la estructura o función de los tejidos, que pueden incluir la presencia de células anómalas, inflamación, daño o muerte celular, entre otros aspectos. Estos exámenes son evaluados por un patólogo, quien interpreta la imagen y emite un diagnóstico, siendo cruciales para la detección y diagnóstico de diversas enfermedades, como el cáncer, enfermedades infecciosas y muchas otras condiciones médicas [21].

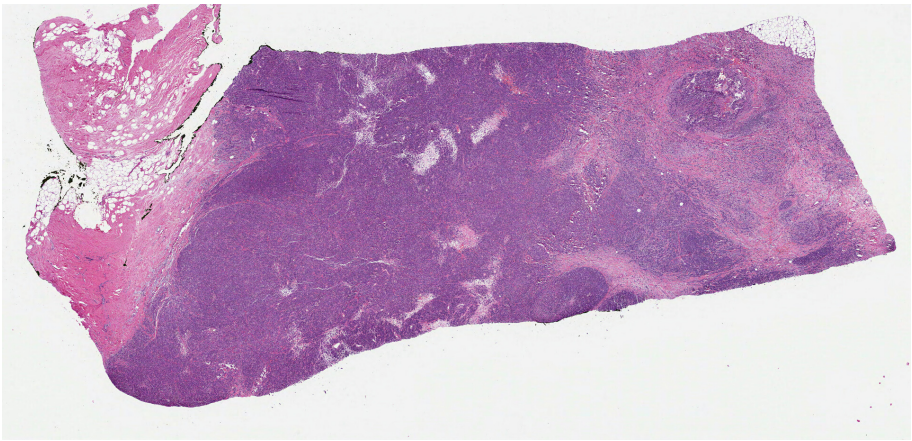


Figura 2.1: Imagen histológica de seno [2].

2.3. Aprendizaje supervisado

El aprendizaje supervisado es un tipo de aprendizaje automático que busca estimar una función $f : \mathcal{X} \to \mathcal{Y}$ que modele la relación entre los datos de entrada $\mathcal{X}$ y las correspondientes salidas o etiquetas $\mathcal{Y}$. Esta estimación se lleva a cabo utilizando un conjunto de entrenamiento $\mathcal{D} = \{(X_1, Y_1), (X_2, Y_2), ..., (X_N, Y_N)\}$, donde a partir de este se busca minimizar una función de costo o perdida $L(f(X), Y)$, donde L mide la discrepancia entre las predicciones $f(X)$ y las etiquetas Y [22].

La función f tiene un conjunto de parámetros θ, los cuales son inicializados aleatoriamente en un principio, y se ajustan mediante algoritmos de optimización de parámetros como gradiente descendente. Estos algoritmos buscan reducir el costo de la función de perdida L a través de ajustes al conjunto de parámetros θ. El objetivo

final es encontrar un conjunto de parámetros θ que minimicé el costo L y que de buenas predicciones para nuevos datos, es decir, entradas que no hayan aparecido en el conjunto de entrenamiento $\mathcal{D}$.

Sin una o múltiples personas que asignen las etiquetas Y, no sería posible contar con una función de costo L que mida dicha discrepancia. Es esta función de costo, en conjunto con un algoritmo de optimización y el conjunto de entrenamiento D, lo que actúa como un profesor o supervisor que le enseña al modelo matemático f la relación entre los datos, de ahí viene el nombre "aprendizaje supervisado".

2.3.1. Aprendizaje con un único anotador

Los problemas de aprendizaje supervisado de imágenes con un único anotador suelen consistir de un conjunto de entrenamiento $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$, donde N es la cantidad de registros en el conjunto, $\mathbf{x}_i \in \mathbb{N}^{M \times P \times C}$ es una matriz de M píxeles de altura, P píxeles de ancho, y $C \in \{1, 3\}$ canales de la imagen (1 para escala de grises y 3 para RGB). Para problemas de clasificación multiclase $y_i \in \{1, ..., K\}$ es la etiqueta conocida, representada como un número que indica la clase correspondiente. Por ejemplo, 1 podría representar "tumor", 2 podría representar "estroma", etc.

La etiqueta y_i debe ser idealmente dada por una persona que sea capaz de dar una clasificación lo más objetiva posible, es decir, un experto en el área de conocimiento que se está trabajando. A esta única etiqueta se le suele llamar "ground truth" o "gold standard". En el área de la medicina y las imágenes histológicas, esto es un proceso laborioso que consume mucho tiempo, sobre todo para un solo anotador.[22]

2.3.2. Aprendizaje con múltiples anotadores

Los problemas de aprendizaje supervisado con múltiples anotadores son muy similares a la sección 2.3.1, con la principal diferencia en el conjunto de entrenamiento, específicamente en el conjunto de etiquetas. En vez de contar con una única etiqueta "gold standard" y_i por entrada de datos $\mathbf{x}_i$, se cuenta con múltiples etiquetas para cada entrada de datos que provienen de distintos anotadores con diferentes niveles de conocimiento.

Se tiene entonces un conjunto de entrenamiento $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{N}$, donde $\mathbf{y}_i \in \{\varnothing, 1, ..., K\}^R$ es un vector de R etiquetas, siendo y_i^r la etiqueta del anotador r para la instancia $\mathbf{x}_i$. Es posible que un anotador no etiquete todas las instancias del conjunto de entrenamiento, esto se ve indicado por el símbolo $\varnothing$.

Este enfoque agiliza el proceso de recolectar etiquetas, puesto que puedes contar con múltiples personas para anotar las entradas de datos, pero introduce nuevos problemas que hay que tener en cuenta. Los anotadores, al tener diferentes niveles de conocimiento y sesgos, pueden diferir en las anotaciones que proporcionan,

introduciendo ruido en el conjunto de datos. Además, es posible contar con anotadores maliciosos que intencionadamente etiqueten erróneamente distintas entradas de datos. Es necesario tener en cuenta estos problemas a la hora de entrenar nuevos modelos, en particular es importante realizar cambios sobre las funciones de costo para que tengan en cuenta el ruido introducido por múltiples anotadores. [22, 23]

2.4. Redes neuronales

Las redes neuronales son un modelo computacional inspirado en la estructura y el funcionamiento del cerebro humano, utilizado principalmente en el campo del aprendizaje automático y la inteligencia artificial. Compuestas por unidades básicas llamadas neuronas, estas redes están organizadas en capas que permiten la transformación y el procesamiento de datos. A través del entrenamiento, donde se ajustan los pesos y sesgos de las conexiones entre neuronas, las redes neuronales pueden aprender patrones complejos a partir de grandes conjuntos de datos. Este enfoque ha demostrado ser especialmente eficaz en tareas como el reconocimiento de imágenes, el procesamiento del lenguaje natural, etc. Una red neuronal básica consiste principalmente de tres capas, una capa de entrada, una capa oculta y una capa de salida [24]:

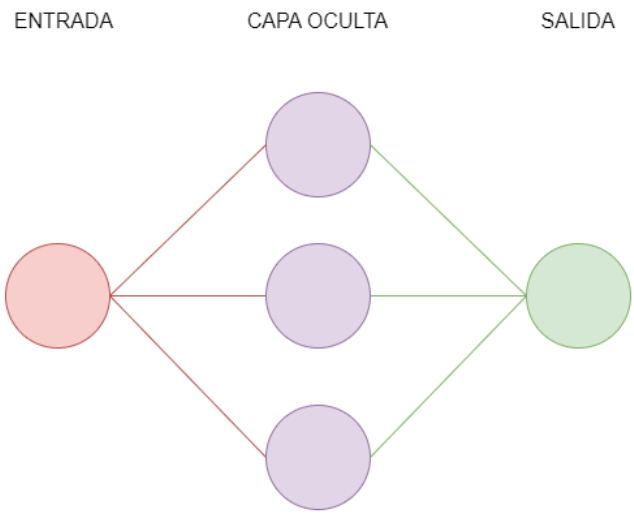


Figura 2.2: Ejemplo de una red neuronal básica que consta de una neurona de entrada, tres neuronas en la capa oculta y una neurona de salida.

- **Capa de entrada:** La capa de entrada es la más sencilla, representa el conjunto de datos de entrada, y tendrá la misma cantidad de nodos o neuronas en esta capa, que dimensionalidad de los datos de entrada $\mathbf{x}$.
- **Capa oculta:** La capa oculta recibe $\mathbf{x}$ de la capa de entrada y procede a calcular tantas funciones lineales como neuronas en la capa oculta haya. Si se

toma como ejemplo una red neuronal con una neurona en la capa de entrada, tres neuronas en su capa oculta, una neurona de salida y un vector de pesos $\phi = \{\theta_{10}, \theta_{11}, \theta_{20}, \theta_{21}, \theta_{30}, \theta_{31}\}$, la función lineal z_j se calcula de la siguiente manera:

$$z_1 = \theta_{10} + \theta_{11}x$$
$$z_2 = \theta_{20} + \theta_{21}x$$
$$z_3 = \theta_{30} + \theta_{31}x$$

Luego, a cada una de las funciones lineales z_j se le aplica una función de activación a para generar la salida h_j. El objetivo de una función de activación en las redes neuronales es introducir no linealidad en el modelo. Esta no linealidad es crucial porque permite a la red neuronal modelar relaciones complejas en los datos. Algunas funciones de activación son: tangente hiperbólica, ReLU, sigmoide, softmax, etc. Entonces, el cálculo de h_j termina siendo:

$$h_1 = a(z_1) = a(\theta_{10} + \theta_{11}x)$$
$$h_2 = a(z_2) = a(\theta_{20} + \theta_{21}x)$$
$$h_3 = a(z_3) = a(\theta_{30} + \theta_{31}x)$$

Para un caso general, el cálculo de la salida h_j es de la siguiente manera:

$$h_j = a(z_j) = a(\theta_{j0} + \sum_{i=1}^{n} \theta_{ji}\mathbf{x}_i)$$

Siendo n la cantidad de neuronas en la capa de entrada y $\mathbf{x}_i$ la salida de cada una de estas neuronas. θ_{ji} es el peso aplicado a cada valor $\mathbf{x}_i$ de la neurona j de la capa oculta y θ_{j0} corresponde al sesgo de la neurona j [24].

- **Capa de salida**: La capa de salida es la encargada de generar las predicciones de la red neuronal, para esto se combinan las salidas h_j de la capa oculta mediante una función lineal. Siguiendo con el ejemplo anterior y teniendo un vector de pesos $\phi = \{\phi_0, \phi_1, \phi_2, \phi_3\}$, esta función lineal c tiene la siguiente forma:

$$c = \phi_0 + \phi_1 h_1 + \phi_2 h_2 + \phi_3 h_3$$

Luego de hacer la combinación de las salidas de la capa oculta h_j, se debe aplicar una función de activación a para poder generar las predicciones. Esta función de activación, por ejemplo, debe ser sigmoide en caso de tratarse de un problema de clasificación binaria o *softmax* para un problema de clasificación con K clases.

Dado que el ejemplo usado en esta sección consiste de un problema de clasificación binaria, se debe aplicar la función sigmoide a la combinación de las salidas h_j:

$$a(x) = \frac{1}{1 + e^{-x}}$$
$$y = a(c) = a(\phi_0 + \phi_1 h_1 + \phi_2 h_2 + \phi_3 h_3)$$

Para un caso general, donde hay múltiples neuronas de salida y_k de la red neuronal, se calcula de la siguiente forma:

$$y_k = a(c_k) = a(\phi_{k0} + \sum_{j=1}^{m} h_j \phi_{kj})$$

Donde ϕ_{kj} es el peso asignado a cada salida h_j para la neurona k en la capa de salida, y ϕ_{k0} es el sesgo para la neurona k.

Utilizando aprendizaje automático, se busca optimizar los parámetros θ y ϕ de las neuronas, de tal forma que se puedan realizar predicciones generales para el problema que se desea representar. Además, a medida que se agregan más neuronas en la capa oculta, se pueden representar problemas más complejos, sin embargo, esto también aumenta la cantidad de parámetros que se deben optimizar, lo que aumenta el tiempo de entrenamiento[24].

2.5. Aprendizaje profundo

El aprendizaje profundo es una rama del aprendizaje automático que, a diferencia de los modelos tradicionales, cuenta con múltiples capas ocultas, lo que les permite aprender estructuras más complejas de datos, como imágenes o reconocimiento de voz. Los modelos de aprendizaje profundo tienen la capacidad de mejorar su rendimiento y aprovechar una mayor cantidad de datos para mejorar su aprendizaje. Cabe resaltar que, mientras mayor sea la complejidad o profundidad del modelo, se requiere una mayor cantidad de datos para que el modelo haga predicciones más precisas[25]. Además, estos modelos utilizan técnicas avanzadas como la retropropagación para ajustar los parámetros internos, optimizando así la función de pérdida y mejorando el desempeño del modelo. Las arquitecturas comunes incluyen redes neuronales convolucionales (CNNs) para el procesamiento de imágenes y redes neuronales recurrentes (RNNs) para datos secuenciales, así como transformadores para el procesamiento de lenguaje natural.

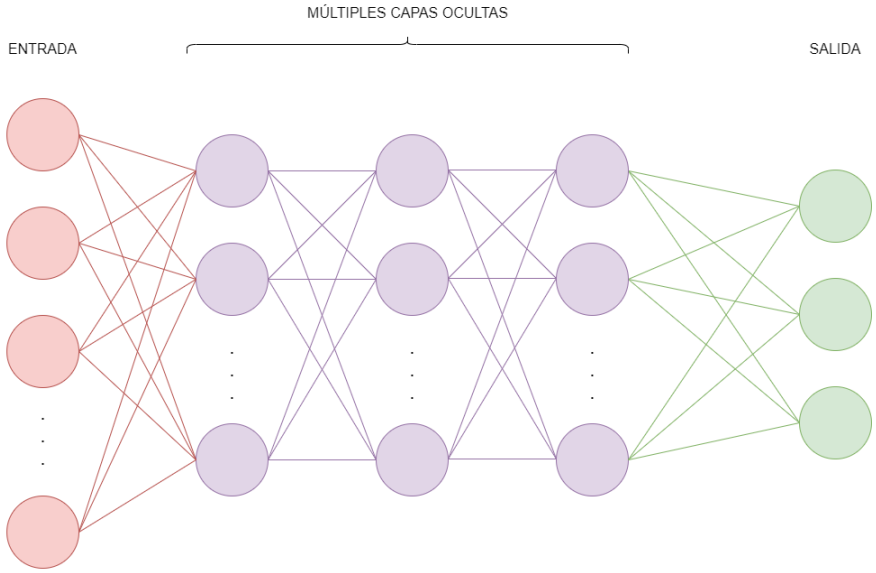


Figura 2.3: Ejemplo de una red neuronal profunda.

2.6. Red neuronal convolucional (CNN)

Una red neuronal convolucional, también conocida como Convolutional Neural Network (CNN) en inglés, es un tipo especial de red neuronal diseñada para trabajar con datos no estructurados como imágenes, vídeos, audio y lenguaje natural. Las CNN son especialmente útiles para reconocer imágenes, texto y dígitos escritos a mano, detección de anomalías, descubrimiento de medicamentos, evaluación de riesgos de salud, pronóstico de series temporales, entre otros usos frecuentes[3, 26].

En ocasiones, para garantizar el desempeño óptimo de una Convolutional Neural Network (CNN), es esencial disponer de una cantidad adecuada de datos variados, tales como imágenes, audios, texto, entre otros. Estos datos deben ofrecer información suficiente para permitir que la red neuronal extraiga características relevantes de manera efectiva durante el proceso de entrenamiento. De no contar con un robusto conjunto de entrenamiento o los recursos computacionales para entrenar un modelo convolucional desde cero, es posible usar CNNs ya previamente entrenadas. Estas redes ya han aprendido características útiles de los datos, lo que reduce significativamente el tiempo y los recursos necesarios para entrenar modelos en nuevas tareas. A este proceso se le llama transfer learning o aprendizaje por transferencia[3, 27].

Generalmente, una CNN se compone de diferentes capas que permiten la extracción de características para llegar a una predicción final. Estas capas son:

- **Capa de Convolución**: Esta capa se encarga de la extracción de características significativas de la imagen de entrada mediante filtros(kernels)[3] que se mueven sobre la imagen y realizan operaciones de convolución para detectar características como bordes, texturas y patrones[28, 29].

- **Capa de Pooling (Agrupación)**: La capa de pooling reduce la dimensionalidad de las características extraídas por la capa anterior. Esto se logra al usar operaciones como Maxpooling, Average pooling o Minpooling[3] que agrupan estas características reduciendo así la dimensión de las mismas.[28, 29]
- **Capa Totalmente Conectada (Fully Connected)**: Esta capa consta de una red neuronal tradicional en la que todas las neuronas están conectadas entre sí. Su función es tomar las características extraídas por las capas anteriores y las utiliza para realizar la clasificación[3, 28].
- **Capa de Salida**: Esta capa se encarga de generar la predicción final utilizando una función de activación, como la función softmax, para obtener la predicción deseada[3].

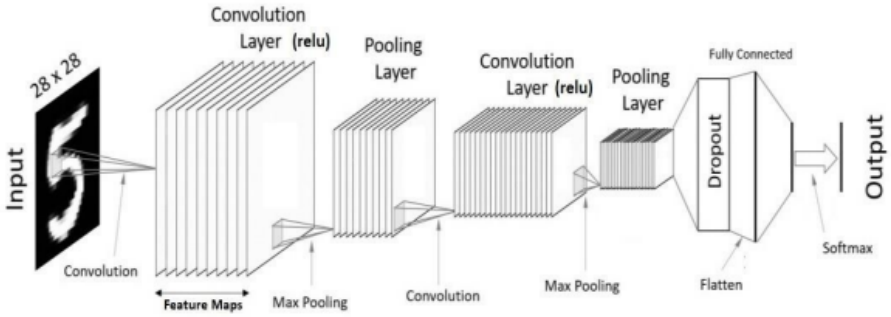


Figura 2.4: Visualización de una arquitectura típica de una CNN. [3]

2.6.1. VGG16

La VGG16 es una red neuronal convolucional creada por los investigadores Karen Simonyan y Andrew Zisserman del grupo Visual Geometry Group (VGG) de la Universidad de Oxford en 2014.[30] Esta CNN es comúnmente conocida por su simplicidad y su alto rendimiento en una variedad de conjuntos de datos, especialmente en las tareas de clasificación de imágenes, gracias a que fue entrenada principalmente en el conjunto de datos ImageNet, que es uno de los conjuntos de datos más grandes y ampliamente utilizados para la clasificación de imágenes. ImageNet contiene millones de imágenes etiquetadas que abarcan miles de categorías diferentes.

La arquitectura de la VGG16 es una red neuronal convolucional que consta de un total de 16 capas, incluyendo 13 capas convolucionales y 3 capas completamente conectadas. Las capas convolucionales están compuestas por bloques de convoluciones 3x3, seguidos de capas de agrupación máxima (max-pooling) para reducir la dimensionalidad.[30] La red comienza con dos capas convolucionales de 64 filtros, seguidas de dos capas de 128 filtros, tres capas de 256 filtros, y finalmente, tres capas de 512 filtros. Después de cada par de capas convolucionales, se aplica una capa de agrupación máxima para reducir la resolución espacial. Posteriormente, la red tiene tres capas completamente conectadas, con las dos primeras capas conteniendo 4096

neuronas y activación ReLU. La última capa completamente conectada tiene 1000 neuronas con activación softmax, correspondientes a las clases de salida de la base de datos ImageNet.

2.7. Funciones de costo

Una función de costo o perdida, es aquella que mide la discrepancia entre las predicciones $f(\mathbf{x}_i)$ de un modelo y las etiquetas "ground truth" y_i del conjunto de entrenamiento. La discrepancia se mide como un número, cuyo valor depende de la función de costo utilizada. Durante el entrenamiento del modelo, se busca encontrar un conjunto de parámetros θ que minimice el valor dado por la función de costo, de tal forma que la discordancia entre las predicciones y las etiquetas sea lo menor posible.

Existen diversas funciones de costo, cada una con sus aplicaciones dependiendo del contexto del problema. A continuación, se explicarán las funciones de costo que fueron usadas para el entrenamiento de modelos en este trabajo:

2.7.1. Cross Entropy

La entropía cruzada (*cross-entropy*), también conocida como entropía cruzada categórica (*categorical cross-entropy*) para problemas de clasificación multiclase, es una función de costo comúnmente usada en tareas de clasificación, que penaliza fuertemente aquellas muestras que un modelo clasifique incorrectamente. La entropía cruzada para una muestra se calcula como:

$$CCE = \sum_{i=1}^{N} \left(- \sum_{k=1}^{K} (y_{ik} \cdot \log(\hat{y}_{ik})) \right)$$

$y_{ik} \in \{0, 1\}$ se refiere a la etiqueta con codificación one-hot correspondiente a la entrada $\mathbf{x}_i$ para la clase k. $\hat{\mathbf{y}}_i$ se refiere a las probabilidades predichas para cada una de las K clases dadas por el modelo para la entrada $\mathbf{x}_i$ [24]. Esta función de costo es relativamente sencilla, pero es posible simplificarla.

Teniendo en cuenta que $\mathbf{y}_i$ es un vector con codificación one-hot, $\mathbf{y}_i$ es un vector con un elemento igual a 1 y $K - 1$ elementos iguales a cero, en consecuencia, $K - 1$ multiplicaciones $\mathbf{y}_{ik} \cdot \log(\hat{y}_{ik})$ resultaran en cero, dando como resultado:

$$CCE = \sum_{i=1}^{N} - \log(\hat{y}_{i\hat{k}})$$

Siendo $\hat{k}$ la clase verdadera para la muestra $\mathbf{x}_i$, es decir, aquella muestra donde $y_{ik} = 1$. Esta función de costo es ampliamente usada, pero puede presentar problemas cuando se tiene un conjunto de datos con muestras ruidosas, puesto que da valores muy altos de costo para muestras mal clasificadas, lo cual puede crear sesgos en el modelo hacia estas muestras ruidosas [31].

2.7.2. Generalized Categorical Cross Entropy

En el contexto de clasificación de imágenes con múltiples anotadores, la entropía cruzada presenta dos problemas, en primer lugar, en casos donde el conjunto de datos es ruidoso, es posible que el rendimiento del modelo se vea afectado [31], y en segundo lugar, la función espera recibir una única etiqueta, pero se cuentan con múltiples etiquetas, posiblemente ruidosas, de personas con diferentes grados de conocimiento y experiencia.

Para resolver el primer problema, existen funciones de costo robustas, como el error absoluto medio (*Mean Absolute Error* o MAE), la cuales son resistentes a datos ruidosos, pero en consecuencia tienden a no poder capturar las relaciones entre los datos, resultando en modelos menos precisos [31]. El error absoluto medio (MAE) puede ser usado en tareas de clasificación al calcularse de la siguiente forma:

$$MAE = \sum_{i=1}^{N} \sum_{k=1}^{K} y_{ik} \cdot (1 - \hat{y}_{ik})$$

Se desea una función que tenga cierta resistencia al ruido y que al mismo tiempo pueda capturar las relaciones de entre los datos de entrada y de salida, de tal forma que tenga un buen rendimiento. Combinando MAE y CCE, se puede obtener la siguiente función de costo:

$$GCCE = \sum_{i=1}^{N} \sum_{k=1}^{K} \mathbf{y}_{ik} \frac{1 - \hat{\mathbf{y}}_{ik}^{q}}{q}$$

donde $q \in (0, 1]$. Esta función de costo, llamada *Generalized Categorical Cross-entropy* [32], cambia de comportamiento dependiendo del valor del parametro q. Cuando $q = 1$, la función se comporta como el error absoluto medio, lo que le da más resistencia al ruido, y a medida que q toma valores más cercanos a 0, la función se comporta de manera similar a la entropía cruzada, por lo que dependiendo del ruido presente en el conjunto de datos, puede ser de interes buscar un valor de q que de un mejor entrenamiento para el modelo.

Sin embargo, esta función todavía no acepta múltiples anotadores. Además, es importante tener en cuenta que cada anotador puede tener sesgos o equivocarse, por lo que es de interés incluir en la función alguna forma de representar la confianza que

se tiene para cada anotador en el conjunto de datos. Se puede modificar la función previa de siguiente manera:

$$GCCE = \sum_{i=1}^{N} \sum_{r \in \Psi_i} \left[\hat{\lambda}_i^r \left(\sum_{k=1}^{K} y_{ik}^r \frac{1 - \hat{\mathbf{y}}_{ik}^q}{q} \right) + (1 - \hat{\lambda}_i^r) \left(\frac{1 - (\frac{1}{K})^q}{q} \right) \right]$$

donde Ψ_i representa el conjunto de anotadores que etiquetaron la muestra i, $\lambda_i^r \in [0, 1]$ es un valor que representa la confiabilidad que se tiene para el anotador r, y y_{ik}^r es la etiqueta del anotador r para la clase k en la muestra i [32].

La confiabilidad λ_i^r es un valor que sé entrena con el modelo, y se va adaptando a medida que el modelo se entrena. El valor λ_i^r se puede ver como un porcentaje de confianza, dándole un peso al voto del anotador. El resto del porcentaje $1 - \lambda_i^r$, sé obtiene como la probabilidad uniforme de que la muestra pertenezca a cualquier clase, es decir, $1/K$.

2.7.3. Focal Loss

Previamente, en la sección 2.7.1, se explicó sobre como la entropía cruzada penaliza a aquellas muestras que el modelo clasifica incorrectamente. La perdida focal, o *focal loss* (FL), le da aún más énfasis a este aspecto de la entropía al agregar a la entropía cruzada por un factor adicional:

$$FL = \sum_{i=1}^{N} \left(- \sum_{k=1}^{K} y_{ik} (1 - \hat{y}_{ik})^{\gamma} \log(\hat{y}_{ik}) \right)$$

Esta modificación a la entropía cruzada proviene de la investigación "Focal Loss for Dense Object Detection"[33], y agrega un factor $(1 - \hat{y}_{ik})$, el cual será más bajo mientras mayor sea la probabilidad $\hat{y}_{ik}$ que le dé el modelo a la clase k. Además, se agrega un parámetro γ, el cual modifica la importancia de este factor agregado.

Para poder interpretar el impacto de este cambio, se puede observar la figura 2.5, en donde a medida que aumenta γ, el costo o *loss*, de las muestras bien clasificadas se reduce, lo que le da un mayor enfoque durante el entrenamiento a las muestras "dificiles", es decir, aquellas muestras en las que el modelo le asigna una baja probabilidad a la clase verdadera.

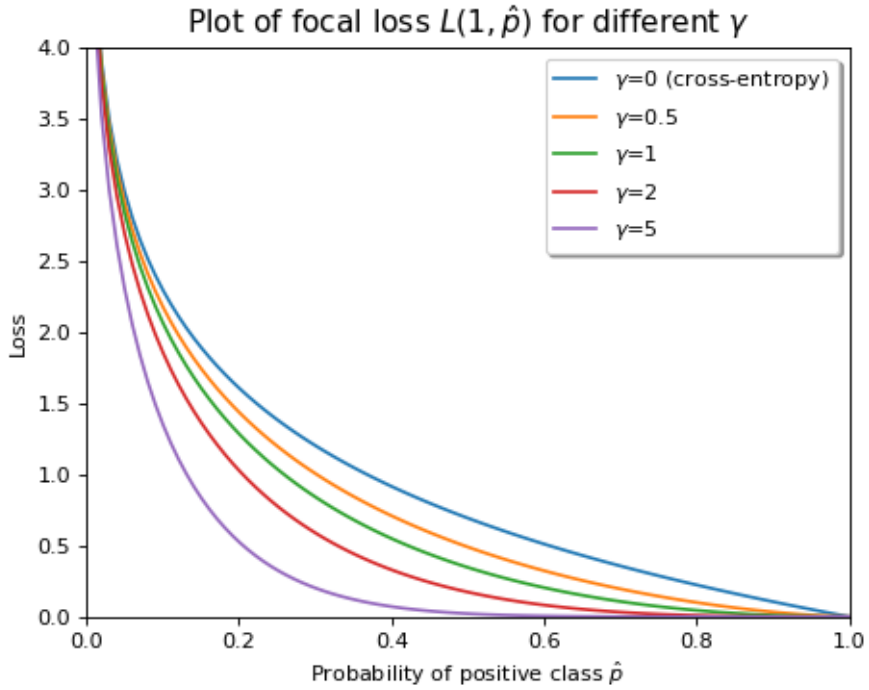


Figura 2.5: Comportamiento de la perdida focal con diferentes γ. Tomado de [4].

Otro problema común en los conjuntos de datos, es el desbalance de clases, en donde hay cantidades drásticamente diferentes en las muestras disponibles para cada clase. Es posible agregar un factor adicional a la perdida focal de tal forma que la función de costo le dé distintos pesos a cada clase según se desee:

$$FL = \sum_{i=1}^{N} \left(-\sum_{k=1}^{K} y_{ik}(\alpha_k)(1 - \hat{y}_{ik})^{\gamma} \log(\hat{y}_{ik}) \right)$$

donde $\alpha \in \mathbb{R}^K$ es un vector que le da un peso a cada clase k. De esta forma es posible asignar pesos a α de tal forma que ciertas clases tengan más prioridad que otras durante el entrenamiento.

Capítulo 3

Metodología

3.1. Obtención del conjunto de datos

Los datos obtenidos en este proyecto provienen del estudio "Structured crowdsourcing enables convolutional segmentation of histology images"[2]. Esta investigación propone un método de crowdsourcing estructurado que facilita la colaboración de diferentes participantes con diversos niveles de conocimiento para generar anotaciones en imágenes histológicas.

El experimento se inicia con un conjunto de datos de imágenes histológicas tomadas de The Cancer Genome Atlas (TCGA)[9]. Específicamente, se seleccionaron 151 imágenes de diapositivas completas teñidas con hematoxilina y eosina (H&E) de tejidos de cáncer de seno. Estas imágenes representan varias muestras de casos confirmados de cáncer de seno y, con la colaboración de patólogos senior, se seleccionaron 151 regiones de interés para asegurar la representación de las principales clases de tejidos observados en el cáncer de seno.

El proceso de crowdsourcing estructurado involucra la colaboración de estudiantes de medicina (NP), patólogos junior (JP) y patólogos senior (SP), asignándoles tareas según su nivel de conocimiento. Los participantes debían realizar las anotaciones sobre las imágenes en la plataforma DSA[34], donde organizaron las anotaciones por clases de región (por ejemplo, tumor, necrosis) con estilos predefinidos para mejorar la consistencia. Los patólogos senior revisaron las anotaciones realizadas por los participantes no expertos, proporcionando retroalimentación y realizando las correcciones necesarias.

Las regiones anotadas en el conjunto de datos son almacenadas como imágenes en escala de grises, donde cada pixel tiene un valor que corresponde a la clase asignada, a estas imágenes se les llama "máscara". Las clases que pueden tomar cada uno de los píxeles consisten de 22 clases específicas, las cuales pueden ser simplificadas en 4 conjuntos de clases: tumor, estroma, infiltración linfocítica y otros. En este proyecto se trabajarán con los primeros 3 conjuntos de clases.

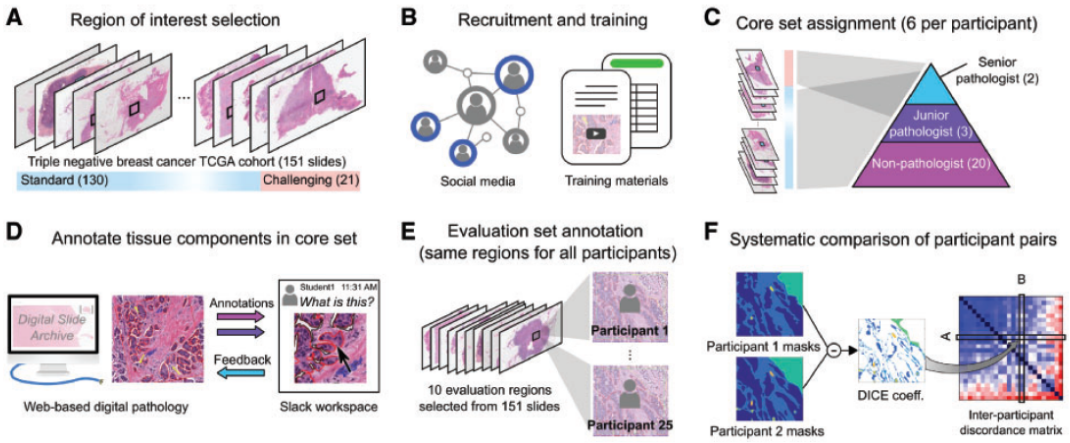


Figura 3.1: Proceso de crowdsourcing estructurado planteado en el artículo *Structured crowdsourcing enables convolutional segmentation of histology images* [2].

Para más información acerca del proceso de crowdsourcing estructurado se puede consultar el artículo: "Structured crowdsourcing enables convolutional segmentation of histology images"[2], a su vez, la información necesaria para poder descargar y utilizar los datos de esta investigación se encuentra en el siguiente repositorio de GitHub: `https://github.com/PathologyDataScience/BCSS`.

3.2. Preprocesamiento de las imágenes histológicas

El preprocesamiento de imágenes, especialmente en tareas de análisis y reconocimiento, es de vital importancia para poder mantener una consistencia en el proceso de entrenamiento y clasificación de los modelos. En este caso, se parte de las 151 regiones de interés anotadas, provenientes del proceso descrito en la sección 3.1, por lo que se pueden tener ciertas diferencias en los colores y tamaños entre las imágenes, lo que afectará el entrenamiento de los modelos.

Una técnica comúnmente empleada con este propósito es la normalización de color, que busca alinear las distribuciones de color de varias imágenes para que sean comparables y coherentes[35]. Asimismo, debido a que las imágenes pueden tener diferentes tamaños, es necesario normalizar el tamaño de las mismas para asegurar datos de entrada uniformes a los modelos durante el entrenamiento. Esta normalización de tamaño facilita la manipulación de las imágenes y mejora la precisión y consistencia del modelo al evitar variaciones que puedan introducirse debido a diferencias en las dimensiones de las imágenes.

3.2.1. Normalización de color

Existen diversos métodos para realizar la normalización de color de las imágenes histológicas. En este caso, se usa la normalización de color de Reinhard[36]. Esta técnica se basa en el ajuste de la media y la desviación estándar de los canales de color de las imágenes, lo que permite alinearlas en un espacio de color común y mitigar las diferencias de iluminación y calibración entre las diferentes fuentes de imágenes. Lo que resulta en una mejora de la coherencia visual y la comparabilidad entre las imágenes. Además, la normalización de color de Reinhard también puede contribuir a la robustez y la predicción de los modelos, al reducir el impacto de las variaciones de color que pueden surgir debido a diferencias en las condiciones de captura o en las fuentes de datos.

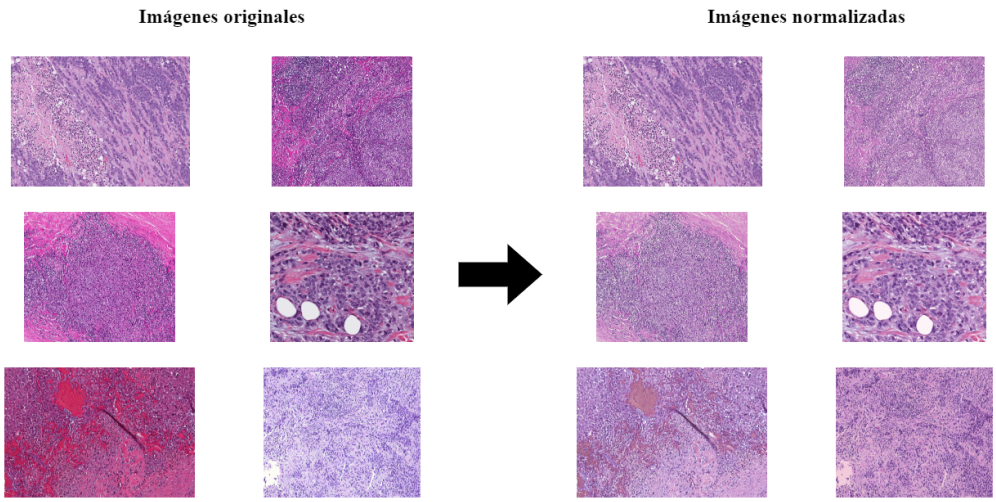


Figura 3.2: Normalización de color de Reinhard de las imagenes

3.2.2. Segmentación de las imágenes

Las imágenes disponibles en el conjunto de datos no solo tienen variaciones en color, sino que también tienen dimensiones considerablemente diferentes, que pueden ir desde los 1222×1036 a 9335×9248 píxeles. Los modelos de aprendizaje automático típicamente necesitan que los datos de entrada tengan dimensiones uniformes, por lo que es necesario realizar una transformación a las imágenes para que estas puedan ser utilizadas durante el entrenamiento.

Para abordar esta necesidad, se implementa un proceso de segmentación de imágenes junto con sus correspondientes máscaras, asegurando que los segmentos seleccionados cumplan con el requerimiento de que al menos el 50 % de los píxeles del segmento pertenezcan a alguna clase específica en la imagen original. Esto se consigue mediante una ventana deslizante de tamaño fijo de 224x224, que se va desplazando por la imagen y su máscara asociada, extrayendo los segmentos y verifi-

cando si el segmento correspondiente cumple con el criterio de asignación de píxeles mínimo. En caso afirmativo, se guarda el segmento de imagen junto con su máscara y clase para su posterior uso en el análisis o procesamiento de imágenes.

Esta segmentación de imágenes tiene como finalidad, primero, extraer más información de las imágenes, aumentando la cantidad de muestras disponibles para el entrenamiento y, segundo, garantizar que cada segmento tenga información valiosa sobre las clases existentes en la imagen original, lo que facilita su utilización en tareas como clasificación, segmentación o detección de objetos. Además, al limitar la inclusión de segmentos que no cumplen con este criterio, se mejora la calidad y la utilidad de los datos extraídos para su análisis y aplicación posterior.

3.3. Análisis de los datos

Como resultado del proceso de preprocesamiento de las imágenes histológicas, se obtuvo un dataset final que consta de 75,243 imágenes para el conjunto de entrenamiento y 4,364 para el conjunto de prueba. Estas imágenes, tanto del conjunto de entrenamiento como del conjunto de prueba, se almacenan en un sistema de carpetas en el que cada sub-carpeta representa la etiqueta de las imágenes que contiene. Se tienen tres diferentes etiquetas, 1=Tumor, 2=Estroma y 3=Inflamación benigna. Por ejemplo, si una imagen se encuentra en la carpeta denominada "1", significa que esta imagen pertenece a la clase "1", es decir, tumor.

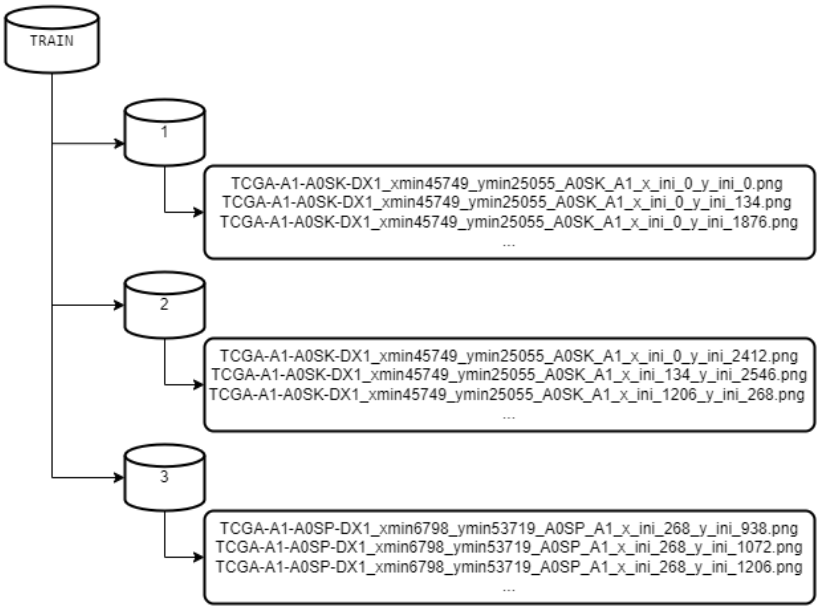


Figura 3.3: Estructura del conjunto de datos para el entrenamiento.

El conjunto de entrenamiento posee un ligero desbalance, siendo casi el 50 % de las muestras pertenecientes a la clase 1, mientras que la clase 3 es la minoría con un

13.71 %. Esto podría generar cierto sesgo a la hora del entrenamiento del modelo, por lo que valdría la pena explorar algunas técnicas que manejen el balanceo de datos.

Clase	Cantidad de imágenes	Porcentaje
1	37260	49.52 %
2	27668	36.77 %
3	10315	13.71 %

Cuadro 3.1: Distribución de imágenes por clase.

Además, se cuenta con las múltiples anotaciones obtenidas del proceso descrito en la sección 3.1. De estas anotaciones, se puede observar que se cuenta con 20 anotadores, que en promedio han anotado 5424 imágenes cada uno.

ID Anotador	Cantidad de anotaciones
5	5036
6	5728
7	6896
8	6084
9	5891
10	7930
11	4092
12	8581
13	5269
14	8223
15	6236
16	5518
17	2100
18	748
19	5739
20	3980
21	4816
22	2285
23	5481
24	7862

Cuadro 3.2: Cantidad de anotaciones por anotador

También es importante resaltar que no todos los anotadores han anotado todas las imágenes, y de hecho, gran parte de las imágenes han sido anotadas por un único anotador, como puede ser visto en la tabla 3.3.

Número de anotaciones	Cantidad de imágenes
1	73103
19	615
18	345
17	297
16	254
11	188
20	110
15	105
14	60
10	51
13	35
9	30
12	21
8	15
7	10
6	4

Cuadro 3.3: Distribución de las anotaciones

De un total de 75243 imágenes, solo 2140 tienen más de una anotación, lo que corresponde al 2.84 % de todo el conjunto de datos. Esto es normal, puesto que si se desea anotar la mayor cantidad de muestras, es mejor dividir el trabajo entre todos los anotadores de tal forma de que no repitan muestras. Sin embargo, esto también quiere decir que en la mayoría de muestras solo se cuenta con una única opinión, lo que puede hacer que modelos entrenados bajo estos datos sean más susceptibles a sesgos individuales de los anotadores.

Clase	Cantidad de imágenes	Porcentaje
1	963	45.00 %
2	822	38.41 %
3	355	16.59 %

Cuadro 3.4: Distribución de imágenes por clase con más de un anotador.

Este conjunto de datos reducido, donde solo se toman en cuenta las muestras que tienen al menos dos anotaciones, será usado en el entrenamiento de algunos modelos para observar el rendimiento de los modelos con un conjunto de datos más pequeño.

3.4. Extracción de características

Para el proceso de extracción de características, se utilizó una VGG16 preentrenada. Después de pre-procesar las imágenes histológicas, cada una de ellas se pasa a través de la VGG16 para extraer sus características. Este procedimiento permite extraer las características de todas las imágenes del conjunto de datos una sola vez, almacenándolas para su uso posterior en el entrenamiento de diferentes modelos. De esta manera, se evita la necesidad de procesar cada imagen a través de la VGG16 en cada ciclo de entrenamiento, lo que reduce significativamente el tiempo y los recursos computacionales necesarios. La estructura de la CNN empleada en este proceso incluye la VGG16 sin la capa "fully connected", seguida de una capa "average pooling" y una capa "flatten", lo que transforma cada imagen en un vector de características que se utiliza posteriormente como datos de entrada en el entrenamiento de los modelos.

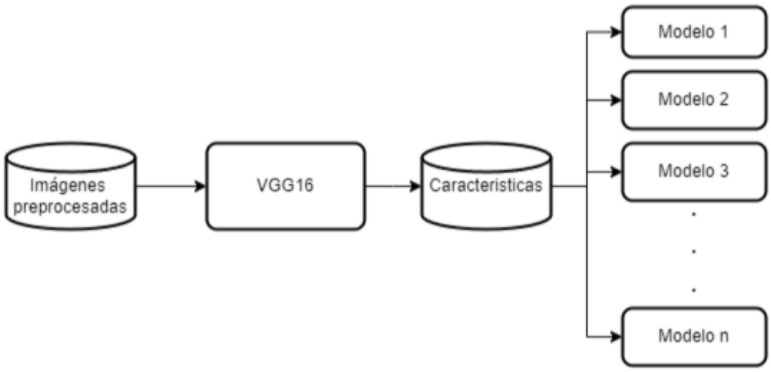


Figura 3.4: Flujo de extracción y utilización de características.

3.5. Entrenamiento de los modelos

En este proyecto se entrenan múltiples modelos de aprendizaje automático, cada uno con ciertas diferencias con respecto al anterior. En esta sección, se explicará la metodología usada para construir y entrenar dichos modelos, así como también las decisiones tomadas con respecto al diseño de los entrenamientos.

El enfoque de esta investigación está en como manejar los múltiples anotadores al momento de entrenar un modelo de aprendizaje profundo para la detección de cáncer de seno en imágenes histológicas. Para poder determinar si las estrategias usadas para manejar los múltiples anotadores dan resultados competentes, es útil tener un punto de referencia. Para esto se dividirán los modelos en 2 categorías: "gold standard" y múltiples anotadores.

Los modelos "gold standard" serán aquellos entrenados con las etiquetas verdaderas del conjunto de datos, es decir, el caso ideal de entrenamiento, y servirán como punto de referencia para analizar los resultados de entrenar modelos de aprendizaje profundo usando múltiples anotadores con diferentes grados de experiencia y conocimiento. Para manejar los múltiples anotadores, se usarán 3 estrategias diferentes: *majority voting*, *generalized categorical cross-entropy* y *crowdlayer*.

Con el fin de limitar la cantidad de variables e hiperparámetros que puedan afectar el análisis y comparación entren los modelos, se decidió usar la misma arquitectura de red neuronal para el entrenamiento de todos los modelos. Esta red se muestra en la figura 3.5, y tiene un total 74115 parámetros entrenables (alrededor de 290KB). Esta red neuronal tendrá algunas variaciones dependiendo de cada modelo, las cuales serán explicadas cuando sea necesario.

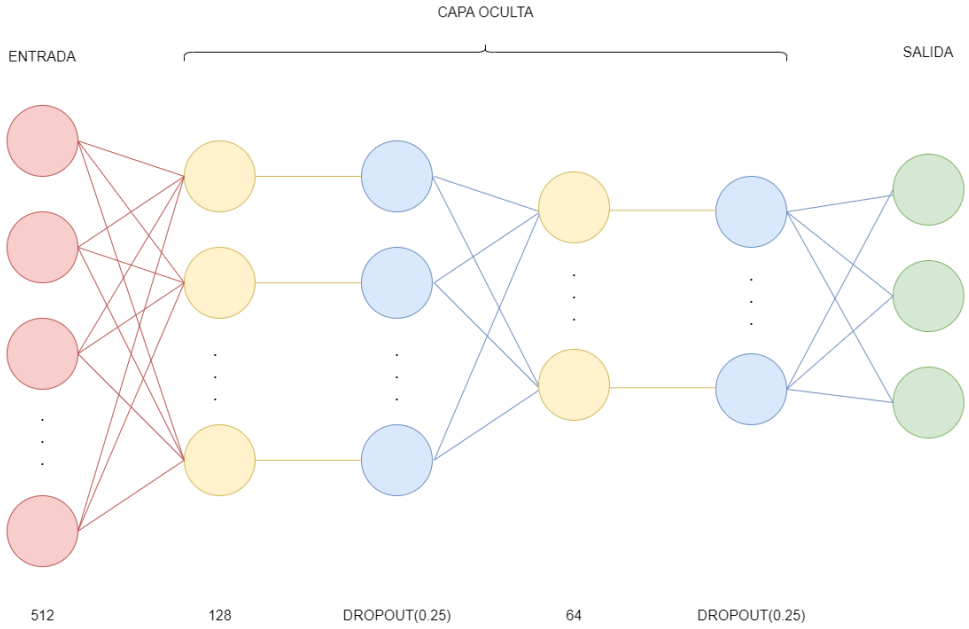


Figura 3.5: Arquitectura de la red neuronal base usada en los modelos del proyecto. Consta de una capa de entrada de 512 neuronas, dos capas ocultas de 128 y 64 neuronas, cada una con una capa de dropout de 0.25 y por último la capa de salida de 3 neuronas para las tres clases (1=cáncer, 2=estroma y 3=inflamación benigna).

Esta arquitectura se obtuvo durante las etapas de exploración del proyecto, donde se probaron modelos con distintas cantidades de capas y neuronas. Se determinó que una arquitectura de 2 capas densas y 2 capas de *dropout* con una poca cantidad de neuronas puede lograr resultados competentes en un tiempo de entrenamiento reducido, lo cual facilita realizar una mayor cantidad de entrenamientos en el menor tiempo posible. Esto permite enfocar el análisis del proyecto en la eficacia de las estrategias para manejar múltiples anotadores, y hace viable el uso de recursos gratuitos de servicios disponibles como Google Colab, que limitan el tiempo de uso continuo del servicio.

Adicionalmente, durante el entrenamiento de los modelos, se observó que la inicialización aleatoria de los parámetros podía influir considerablemente en el desempeño del modelo en algunos casos. Ejemplos de estos casos se observarán en mayor detalle en el capítulo 4. Para mitigar el impacto de la variabilidad que pueda introducir la aleatoriedad, cada modelo se entrenará 10 veces bajo las mismas condiciones.

3.5.1. Gold standard

Este modelo se refiere a una situación ideal en la que se conocen con exactitud las etiquetas verdaderas para cada imagen, lo que proporciona una base comparativa con la cual se mide la eficacia y precisión de los modelos desarrollados. Para este modelo, sé usarán dos versiones con diferentes funciones de costo. La primera versión utiliza "Categorical Cross Entropy" (entropía cruzada categórica), mientras que la segunda emplea "Focal Loss" (pérdida focal).

- **Categorical Cross Entropy:** Esta función de costo mide la diferencia entre las predicciones del modelo y las etiquetas verdaderas. El costo total se toma como la suma de los costos o perdidas entre la clase verdadera y las predicciones del modelo para cada una de las muestras. La entropía cruzada categórica es ideal para este escenario, ya que cada instancia pertenece a una única clase y la función penaliza fuertemente las predicciones incorrectas, incentivando al modelo a asignar una alta probabilidad a la clase correcta.

- **Focal loss:** Esta función de costo es una extensión de la función de entropía cruzada, permitiendo que el entrenamiento del modelo le dé más importancia a las muestras "difíciles", es decir, aquellas que el modelo no logra identificar correctamente. Podemos regular la importancia de estas muestras difíciles a través del hiperparámetro γ. En el artículo *Focal Loss for Dense Object Detection*[33], donde se introduce la función de perdida focal, encontraron que un valor de $\gamma = 2$ solía dar mejores resultados. Para el entrenamiento de los modelos en este proyecto, se probarán 9 valores de γ para poder analizar el impacto de este hiperparámetro en el entrenamiento. En particular se usarán los valores: 1.0, 1.5, 2.0, 2.5, 3.0, 3.5, 4.0, 4.5 y 5.0.

  A esta función se le puede adicionar un hiperparámetro α para regular el desbalance de clases. En la tabla 3.1 se puede observar la distribución de las imágenes que se tienen por cada clase.

  Aunque no es muy significativo, existe un ligero desbalance en la cantidad de imágenes por clase. En este caso se tienen 3 clases, por lo que α será un vector de 3 valores, donde cada valor α_i le da un peso a cada clase respectivamente. Para calcular los pesos, podemos usar la inversa del porcentaje de muestras de cada clase:

$$w = \begin{bmatrix} \frac{1}{p_1} \\ \frac{1}{p_2} \\ \frac{1}{p_3} \end{bmatrix} = \begin{bmatrix} \frac{1}{0.4952} \\ \frac{1}{0.3677} \\ \frac{1}{0.1371} \end{bmatrix} = \begin{bmatrix} 2.0194 \\ 2.7194 \\ 7.2945 \end{bmatrix}$$

Esto permite darle una mayor importancia a aquellas clases que tienen una menor cantidad de muestras. Adicionalmente, podemos convertir estos pesos a sus respectivos porcentajes, donde la suma de los mismos será igual 1, lo cual nos dará el valor final de α:

$$w_{sum} = \sum_{i=1}^{K} w_i$$

$$\alpha = \begin{bmatrix} \frac{w_1}{w_{sum}} \\ \frac{w_2}{w_{sum}} \\ \frac{w_3}{w_{sum}} \end{bmatrix} = \begin{bmatrix} 0.1678 \\ 0.2259 \\ 0.6061 \end{bmatrix}$$

Este α le da un mayor peso a la clase 3, puesto que se dispone de una menor cantidad de muestras, lo que aumenta el costo de las muestras mal clasificadas para dicha clase. Cabe resaltar que esta es solo una de las infinitas formas de asignar valores a α, pero se decidió limitar la exploración de este hiperparámetro a estos valores.

La inclusión de α en la función de costo para el balanceo de clases puede que sea beneficioso para rendimiento del modelo, pero también se realizarán entrenamientos de modelos con pérdida focal sin hacer uso del hiperparámetro α para observar su impacto en los resultados.

3.5.2. Múltiples anotadores

Al entrenar un modelo de aprendizaje automático, es fundamental disponer de etiquetas o anotaciones para cada muestra del conjunto de datos, de tal forma que la función de costo pueda medir el error de las predicciones del modelo. La función de costo permite cuantificar la discrepancia entre las predicciones del modelo y las etiquetas verdaderas, lo cual es esencial para el proceso de optimización de parámetros del modelo.

Cuando se trabaja con múltiples anotaciones para cada muestra del conjunto de datos de entrada, se requiere aplicar métodos o estrategias específicas que permitan manejar las múltiples anotaciones, de tal forma que la función de costo pueda medir el error de las predicciones del modelo. Por ejemplo, la función de costo de entropía cruzada usa una única etiqueta para medir el error del modelo, por lo que si se usa

esta función de costo con múltiples etiquetas, se requiere consolidar las múltiples etiquetas en una única etiqueta, o modificar la entropía cruzada para que pueda hacer uso de las múltiples etiquetas.

Además, es importante tener en cuenta que no todos los anotadores cuentan con los mismos conocimientos y experiencias, cada persona puede tener sesgos y criterios de evaluación diferentes, lo que lleva a variaciones en las etiquetas asignadas.

En este proyecto se considerarán 3 principales estrategias diferentes que abordan las múltiples anotaciones. Para cada una de estas estrategias, se realizarán diferentes variaciones y búsqueda de hiperparámetros para explorar y analizar como dichas variaciones afectan el rendimiento de los modelos.

Majority voting

Esta estrategia es una de las más sencillas, puesto que como su nombre indica, toma el voto mayoritario entre las anotaciones disponibles como la única etiqueta. Existen variaciones de *majority voting* que buscan tomar en cuenta la confiabilidad y sesgo de cada anotador. Por ejemplo, es posible asignarle pesos a cada anotador basado en su experiencia, conocimiento o rendimiento previo, lo que le daría un mayor peso al voto de aquellos anotadores que deberían ser, en teoría, más precisos en sus anotaciones. El conjunto de datos disponible usado en este proyecto no cuenta con información detallada de los anotadores o su experiencia, por lo que estas variaciones no son posibles.

Otro aspecto a tener en cuenta es posibilidad de empates en entre diferentes clases en las anotaciones. Hay múltiples formas de manejar empates: escoger un ganador aleatoriamente, pedir ayuda a un experto para que haga el desempate manualmente, buscar más anotaciones para esa muestra, escoger una clase por defecto, etc. En el caso particular del conjunto de datos de este proyecto hay 25 empates de un total de 75243 muestras.

Debido a la baja cantidad de empates no se le dio mucha importancia en esta investigación, y se optó por usar el comportamiento por defecto de la función "argmax" de la librería numpy, donde la clase con menor índice es la seleccionada. Por ejemplo, si hay un empate entre la clase 1 y 2, la clase 1 es la seleccionada por tener menor valor que la clase 2. El resultado final del *majority voting* se puede observar en la tabla 3.5.

Clase	Cantidad de imágenes	Porcentaje
1	36217	48.13 %
2	28955	38.48 %
3	10071	13.38 %

Cuadro 3.5: Distribución de imágenes por clase con *majority voting*.

El resultado del *majority voting* es bastante cercano a las etiquetas *gold stan-*

dard. De las 75243 imágenes del conjunto de datos, 69718 (92.65 %) coinciden con la clase verdadera.

Con esta estrategia se planea realizar los mismos 2 modelos utilizados en *gold standard* explicados en la sección 3.5.1, es decir, se realizará un modelo con *cross-entropy* y otro modelo con *focal loss*. A diferencia de *gold standard*, se usará como etiqueta única el resultado de hacer votación mayoritaria.

Para la búsqueda de hiperparámetros en *focal loss* del hiperparámetro γ, se utilizará el mismo rango de valores con el fin de comparar con los resultados de los modelos *gold standard*. Esta búsqueda de hiperparámetros se hará dos veces: una vez sin el hiperparámetro α y otra búsqueda usando α. Se utilizará la misma metodología de la sección 3.5.1 para obtener el vector α.

$$w = \begin{bmatrix} \frac{1}{p_1} \\ \frac{1}{p_2} \\ \frac{1}{p_3} \end{bmatrix} = \begin{bmatrix} \frac{1}{0.4813} \\ \frac{1}{0.3848} \\ \frac{1}{0.1338} \end{bmatrix} = \begin{bmatrix} 2.0775 \\ 2.5986 \\ 7.4712 \end{bmatrix}$$

$$\alpha = \begin{bmatrix} \frac{w_1}{w_{sum}} \\ \frac{w_2}{w_{sum}} \\ \frac{w_3}{w_{sum}} \end{bmatrix} = \begin{bmatrix} 0.1710 \\ 0.2139 \\ 0.6150 \end{bmatrix}$$

Generalized categorical cross-entropy

Cómo se mencionó previamente al inicio de esta sección, una de las formas de utilizar múltiples anotaciones es utilizar una función de costo que tenga en cuenta estas condiciones. *Generalized categorical cross-entropy* (GCCE) es una función de costo que permite tomar en cuenta múltiples anotaciones, en conjunto con la "confiabilidad" que el modelo le da a cada anotador.

La confiabilidad es una métrica entre 0 y 1, que permite darle más o menos peso al voto de cada anotador. En GCCE, la confiabilidad se obtiene a partir de R neuronas de salida adicionales a las habituales K neuronas de salida que tiene un modelo para realizar la tarea de clasificación. Esto quiere decir que la confiabilidad de cada anotador es entrenada en conjunto con el resto del modelo, mejorando así la precisión de las predicciones al considerar la variabilidad y calidad de las anotaciones proporcionadas por diferentes fuentes.

Como se puede observar en la figura 3.7, se adicionan las R neuronas en la capa de salida del modelo base de entrenamiento, dando un total de $K + R$ valores de salida en el modelo. Tomando como referencia la tabla 3.2, se conoce que hay $R = 20$ anotadores en el conjunto de datos, y también se conoce que se tienen $K = 3$ clases, por lo que hay $K + R = 23$ valores de salida en total.

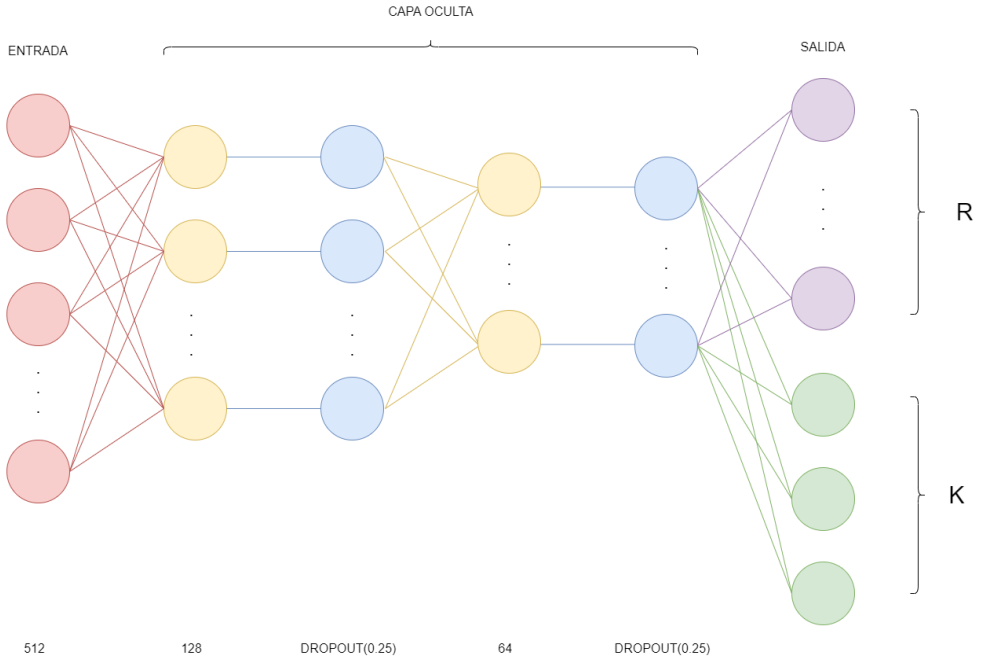


Figura 3.6: Arquitectura de la red neuronal para *generalized categorical cross-entropy*.

Con respecto a la búsqueda de hiperparámetros, se tiene el hiperparámetro $q \in (0, 1]$. Cuando $q = 1$, la función GCCE se comporta como el error absoluto medio (*Mean Absolute Error* o MAE), lo que lo hace muy resistente a muestras ruidosas, puesto que le da la misma importancia a todos los valores. Mientras más bajo es el valor de q, más se comporta como la función de entropía cruzada, la cual penaliza fuertemente malas predicciones. Por lo tanto, es de interés encontrar un q que optimice el entrenamiento del modelo, por lo que se realizará una búsqueda de diez valores de q: 0.01, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9.

Adicionalmente, durante el análisis del conjunto de datos en la sección 3.3, se observó que solo una pequeña parte del conjunto de datos está anotada por más de dos personas. Con el fin de observar el rendimiento de modelos bajo un conjunto de datos reducido, se repetirá la búsqueda de hiperparámetros de q usando un conjunto de datos donde solo se usan aquellas muestras que son anotadas por más de una persona.

Crowdlayer

Crowdlayer es una capa propuesta en el artículo "*Deep Learning from Crowds*"[23], la cual, de manera similar a GCCE, busca aprender directamente de las etiquetas de múltiples anotadores usando diferentes estrategias para calcular los sesgos y pesos de cada anotador por cada clase.

Antes de realizar el entrenamiento de un modelo, la capa *Crowdlayer* se agrega al final del modelo, recibiendo como entrada la usual capa de salida de K neuronas. En cuanto la función de costo, se usará entropía cruzada, pero es necesario aplicar modificaciones para manejar aquellas etiquetas faltantes de cada anotador, de tal forma de que no afecten el resultado de la función de costo. Esto es sencillo de solucionar, ya que solo es necesario asignar un valor de 0 al resultado de la entropía cruzada si el anotador no proporcionó una etiqueta para una muestra.

Al finalizar el entrenamiento de un modelo, la capa *Crowdlayer* se extrae del modelo, dejando expuesta la capa de salida para poder evaluar el rendimiento del modelo. Esta estructura de trabajo permite adaptar modelos a un conjunto de datos con múltiples anotadores sin mayor dificultad.

Crowdlayer propone múltiples formas de manejar el sesgo y pesos de los anotadores. Para tareas de clasificación se proponen cinco configuraciones:

- MW: Una matriz de pesos $\mathbf{W}^r \in \mathbb{R}^{K \times K}$ para cada anotador r.
- VW: Un vector de pesos por clase $\mathbf{w}^r \in \mathbb{R}^K$ para cada anotador r.
- VB: Un vector de sesgos por clase $\mathbf{b}^r \in \mathbb{R}^K$ para cada anotador r
- VW+B: Una combinación entre VW y VB, donde se tiene un vector de pesos y un vector de sesgos.
- SW: Un único peso w para cada anotador.

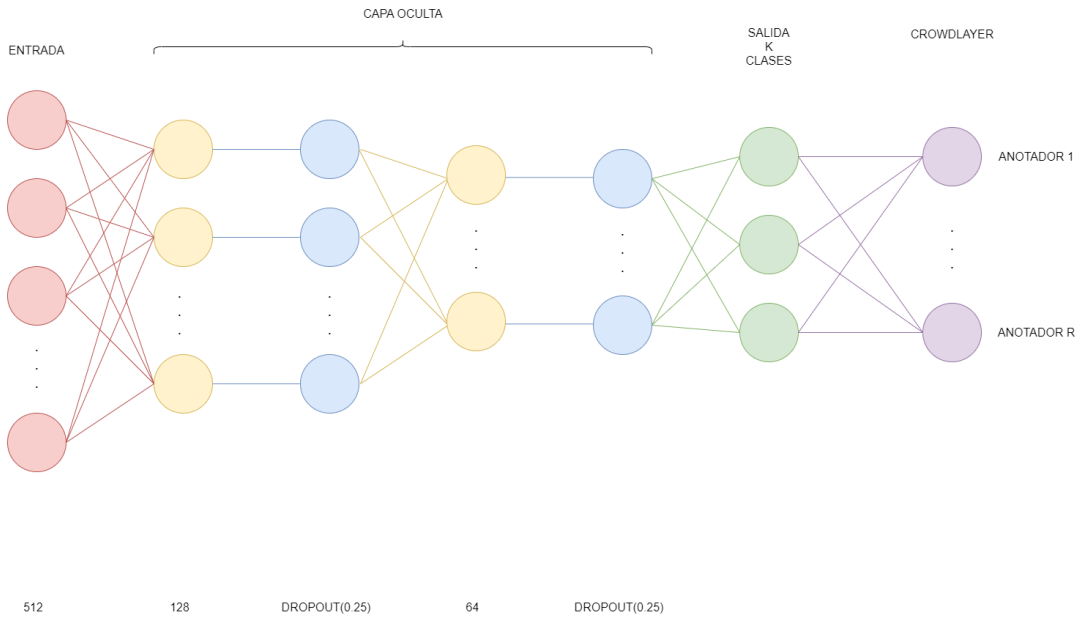


Figura 3.7: Arquitectura de la red neuronal usada en los modelos *Crowdlayer*.

En este proyecto, se tomarán estas cinco configuraciones como hiperparámetros, y se realizarán entrenamientos con los cinco valores para analizar las diferencias en

rendimiento con diferentes pesos y sesgos para los anotadores.

3.6. Criterios de evaluación

Existen múltiples criterios y métricas para medir el desempeño de un modelo de aprendizaje automático. En el área de medicina, es particularmente deseable tener modelos precisos, puesto que se espera que aporten información útil o valiosa. Tener un modelo que siempre prediga con absoluta certeza es una tarea difícil, pero usando criterios de evaluación se espera encontrar que configuraciones y modelos funcionan mejor a comparación de otros para la tarea de clasificación de imágenes histológicas.

Una métrica para evaluar el desempeño del modelo es el costo, también llamado perdida o "loss", que provee la función de costo para el entrenamiento del modelo. Esta métrica permite comparar el rendimiento de diferentes modelos que usen la misma función de costo. Sin embargo, en este proyecto los modelos usan funciones de costo diferentes, e incluso en modelos que usan la misma función de costo, se cambiaron hiperparámetros que afectan la función de costo, lo que proporciona valores de perdida considerablemente diferentes. En consecuencia, usar la perdida como métrica de comparación es inviable.

La exactitud o "accuracy" por otra parte, es una métrica que indica el porcentaje de muestras correctamente predichas por el modelo. Esto nos permite comparar fácilmente el rendimiento entre diferentes modelos basado en la exactitud de sus predicciones después de ser entrenados. Sin embargo, la exactitud no siempre es la mejor forma de comparar, por ejemplo, cuando el conjunto de datos está desbalanceado, lo que significa que algunas clases son significativamente más frecuentes que otras, la precisión puede ser engañosa. Un modelo que predice la clase mayoritaria la mayor parte del tiempo aún puede lograr una alta exactitud, pero funcionar mal en las clases minoritarias.

Para acompañar la exactitud, se usará el puntaje F1 o "F1 Score", el cual se describe como la media armónica entre la precisión y la exhaustividad:

- **Precisión:** es una proporción que indica cuantas de las muestras que el modelo clasificó como parte de la clase k, realmente pertenecen a la clase k.
- **Exhaustividad:** es una proporción que mide cuántas de las muestras pertenecientes a la clase k, fueron correctamente clasificadas por el modelo.

El puntaje F1 permite observar el rendimiento del modelo por cada clase, al balancear la precisión y la exhaustividad en una sola métrica, lo cual es especialmente útil en casos de desbalance de clases para confirmar que el modelo no se esté sesgando a una clase en específico.

Para comparar el desempeño entre los modelos, también se pueden realizar pruebas estadísticas, las cuales permiten determinar si los resultados observados en los datos son significativos o si podrían haber ocurrido al azar. Esto nos permite

analizar si los resultados de las diferentes configuraciones de los modelos tienen diferencias estadísticamente significativas.

En la sección de resultados se usará la prueba de Friedman, la cual es una prueba estadística no paramétrica, utilizada frecuentemente para evaluar la consistencia entre muestras obtenidas de diferentes maneras [37]. Para realizar una prueba de Friedman, es necesario contar con una hipótesis nula y una hipótesis alternativa. Además, se escoge un valor alfa (α) o valor de significancia, el cual es un umbral elegido antes de realizar la prueba para determinar si se rechaza la hipótesis nula. Este valor representa la probabilidad de rechazar la hipótesis nula cuando en realidad es verdadera (error tipo I o falso positivo).

La prueba de Friedman dará como resultado un valor llamado *p-value*, el cual se interpreta de la siguiente manera: si *p-value* está por debajo del nivel de significancia α, entonces se rechaza la hipótesis nula. Cuanto menor sea el *p-value*, más significativo será el resultado.

Un valor común para *alpha* es 0.05 (5 %), sin embargo, se pueden tomar valores más bajos como 0.01 (1 %) o 0.001 (0.1 %) cuando se desea que la probabilidad de rechazar la hipótesis nula incorrectamente sea menor. En el análisis de resultados se usará $\alpha = 0.01$.

Capítulo 4

Resultados

El objetivo principal de esta investigación fue desarrollar y evaluar modelos de aprendizaje profundo para la detección de cáncer de seno a partir de imágenes histológicas. El enfoque estuvo en determinar la efectividad de diferentes conjuntos de anotaciones en el conjunto de datos y estrategias para manejar el ruido de los anotadores. Para lograr esto, se entrenaron y probaron varios modelos de aprendizaje profundo.

En la sección de metodología anterior, se detalló el paso a paso para preparar los datos, la arquitectura de las redes neuronales, el proceso de entrenamiento y los criterios de evaluación. Los modelos fueron evaluados utilizando una variedad de métricas de rendimiento, incluyendo exactitud u "accuracy" y puntaje F1 para cada una de las etiquetas de las imágenes (1=tumor, 2=estroma, 3=inflamación benigna).

Esta sección de resultados está organizada de la siguiente manera:

- Rendimiento de los modelos: Se presentan las métricas de rendimiento de cada modelo entrenado, destacando sus fortalezas y debilidades.
- Análisis comparativo: Se realiza un análisis comparativo de los modelos utilizando la prueba de Friedman.

Cómo se mencionó en la sección 3.5, se repitió el entrenamiento de cada modelo 10 veces para reducir el impacto de la inicialización aleatoria de los parámetros de cada modelo. Con el fin de reducir la cantidad de información en los resultados, sé presentarán las métricas de cada modelo como el promedio de los 10 entrenamientos, seguido del símbolo "±" y la desviación estándar.

Los resultados de los modelos fueron guardados en formato JSON y se pueden encontrar en esta carpeta de Google Drive, en conjunto con un Jupyter Notebook que puede ser abierto con Google Colab, donde se muestra como utilizar los archivos en conjunto con un resumen de los resultados de cada modelo.

Adicionalmente, se creó un repositorio en GitHub que contiene el resto de Notebooks utilizados para el entrenamiento de múltiples modelos e información adicional sobre como obtener los datos utilizados para el entrenamiento de los modelos: `https://github.com/Buitragox/Crowdsourcing-Thesis`.

4.1. Rendimiento de los modelos

4.1.1. Gold standard

Gold standard con función de costo cross-entropy

Accuracy	F1-Score class 1	F1-Score class 2	F1-Score class 3
0.8165 ± 0.0171	0.8679 ± 0.0153	0.7382 ± 0.0173	0.7359 ± 0.0294

Cuadro 4.1: Resultado del modelo gold standard con la función de costo cross-entropy

Los resultados de modelos con una única anotación, calificada como "ground truth" o "gold standard", son un caso ideal, puesto que no siempre es posible contar con la etiqueta verdadera al momento de entrenar un modelo. En este caso, con la arquitectura del modelo implementada y el conjunto de datos utilizado, se obtuvo una exactitud promedio de 0.8165 y con puntajes F1 0.867, 0.738 y 0.735 para las clases 1, 2 y 3 respectivamente. Estos resultados se tomarán como un punto de referencia, en conjunto con los resultados de perdida focal, para medir el desempeño de los modelos con múltiples anotadores.

Gold standard con función de costo focal-loss

γ	Accuracy	F1-Score class 1	F1-Score class 2	F1-Score class 3
1.0	0.8155 ± 0.0102	0.8687 ± 0.0088	0.7462 ± 0.0122	0.7148 ± 0.0307
1.5	0.7942 ± 0.0331	0.8501 ± 0.0376	0.7229 ± 0.0277	0.6905 ± 0.0242
2.0	0.7886 ± 0.0192	0.8478 ± 0.0212	0.7179 ± 0.0193	0.6738 ± 0.0271
2.5	0.8106 ± 0.0118	0.8640 ± 0.0113	0.7372 ± 0.0174	0.7176 ± 0.0251
3.0	0.7949 ± 0.0248	0.8505 ± 0.0308	0.7215 ± 0.0263	0.6973 ± 0.0271
3.5	0.7982 ± 0.0091	0.8571 ± 0.0093	0.7214 ± 0.0224	0.6900 ± 0.0259
4.0	0.7615 ± 0.0608	0.8127 ± 0.0762	0.6921 ± 0.0421	0.6763 ± 0.0599
4.5	0.7842 ± 0.0328	0.8409 ± 0.0369	0.7087 ± 0.0337	0.6852 ± 0.0263
5.0	0.7924 ± 0.0176	0.8553 ± 0.0125	0.6998 ± 0.0506	0.6840 ± 0.0358

Cuadro 4.2: Resultados del modelo gold standard con la función de costo focal loss usando el parámetro α.

La función de costo de *focal loss* busca darle una mayor importancia a las muestras difíciles, reduciendo la importancia de las muestras fáciles de clasificar. Además, se añade el hiperparámetro α, el cual tiene como objetivo darle un peso al costo de las muestras dependiendo de la clase a la que pertenece dicha muestra.

En este proyecto se buscó asignar un mayor peso α_k a las clases que tienen menor representación en el conjunto de entrenamiento, como lo es la clase 3 (inflamación benigna). Para esto, se siguió el procedimiento indicado en la sección 3.5.1, en la

cual primero se obtuvo la inversa del porcentaje de muestras de cada clase, y luego sé dividió cada valor resultante, entre la suma total de los valores, creando un vector de pesos donde las clases con menor cantidad de muestras, tienen un mayor valor en α.

Al observar el rendimiento de los modelos que usan el hiperparámetro α, no se obtuvo ninguna mejora en comparación a usar *cross-entropy*, puesto que ninguna versión de los modelos obtuvo una mejora significativa, ni en los puntajes F1, ni en la exactitud de los modelos.

γ	Accuracy	F1-Score class 1	F1-Score class 2	F1-Score class 3
1.0	0.8250 ± 0.0122	0.8753 ± 0.0106	0.7434 ± 0.0149	0.7451 ± 0.0268
1.5	0.8154 ± 0.0127	0.8661 ± 0.0129	0.7380 ± 0.0144	0.7369 ± 0.0317
2.0	0.8262 ± 0.0087	0.8766 ± 0.0097	0.7457 ± 0.0080	0.7383 ± 0.0271
2.5	0.8213 ± 0.0103	0.8705 ± 0.0098	0.7412 ± 0.0116	0.7481 ± 0.0154
3.0	0.8138 ± 0.0174	0.8682 ± 0.0176	0.7295 ± 0.0143	0.7079 ± 0.0707
3.5	0.8199 ± 0.0102	0.8766 ± 0.0059	0.7260 ± 0.0175	0.7058 ± 0.0544
4.0	0.8121 ± 0.0198	0.8683 ± 0.0162	0.7305 ± 0.0260	0.6923 ± 0.0582
4.5	0.8175 ± 0.0157	0.8704 ± 0.0145	0.7329 ± 0.0178	0.7248 ± 0.0430
5.0	0.8039 ± 0.0213	0.8586 ± 0.0200	0.7139 ± 0.0450	0.7178 ± 0.0391

Cuadro 4.3: Resultado del modelo gold standard con la función de costo focal loss sin usar el parámetro α

Por otro lado, los modelos que usan la versión de *focal loss* sin el hiperparámetro α (comparable a tener $\alpha_k = 1$ para todas las clases), tuvo un rendimiento más consistente, con exactitudes superiores a 0.8 en todos los resultados. Además, tuvo resultados ligeramente superiores a *cross-entropy* en términos de exactitud en algunos de los casos, para hiperparámetros γ iguales a 1.0, 2.0 y 2.5, 4.5. El puntaje F1 también es ligeramente superior en algunos de los casos.

Esto se puede deber a una variedad de causas, entre ellas, el proceso de selección de valores de α puede no ser el mejor para el conjunto de datos con la arquitectura implementada. Una búsqueda de hiperparámetros de α exhaustiva puede llegar a conseguir mejores resultados.

4.1.2. Múltiples anotadores

Majority voting

Accuracy	F1-Score class 1	F1-Score class 2	F1-Score class 3
0.8206 ± 0.0106	0.8706 ± 0.0096	0.7425 ± 0.0137	0.7422 ± 0.0253

Cuadro 4.4: Resultados del modelo con *majority voting* usando la función de costo *cross-entropy*

El *majority voting* (MV) busca agregar las anotaciones de múltiples personas en una única anotación, lo que permita entrenar modelos de manera idéntica al *gold standard*. En este caso, se lograron resultados ligeramente superiores al *gold standard*, con una exactitud de 0.82 y puntajes F1 un poco mayores.

Normalmente, se esperaría que *gold standard* obtuviera un resultado superior en todos los casos, puesto que entrenar modelos con la etiqueta verdadera es en teoría la situación ideal, sin embargo, esto no siempre es el caso. Por ejemplo, el entrenamiento del modelo con votación mayoritaria podría capturar mejor la relación entre los datos del conjunto de evaluación utilizado, dándole una mejor métrica de desempeño.

A pesar del buen desempeño de modelos con MV en este proyecto, es importante recalcar que la técnica de MV usada en este proyecto le da la misma importancia al voto de todos los anotadores, lo cual puede ser perjudicial en presencia de anotadores maliciosos o con fuertes sesgos. Los datos obtenidos en este proyecto fueron obtenidos de manera rigurosa con la supervisión de patólogos expertos [2], pero puede que este no sea el caso en otras fuentes de datos.

γ	Accuracy	F1-Score class 1	F1-Score class 2	F1-Score class 3
1.0	0.7818 ± 0.0287	0.8412 ± 0.0308	0.7082 ± 0.0205	0.6797 ± 0.0295
1.5	0.7715 ± 0.0393	0.8340 ± 0.0393	0.6891 ± 0.0369	0.6758 ± 0.0399
2.0	0.7982 ± 0.0221	0.8578 ± 0.0219	0.7110 ± 0.0233	0.7039 ± 0.0218
2.5	0.7649 ± 0.0344	0.8216 ± 0.0379	0.6976 ± 0.0306	0.6723 ± 0.0408
3.0	0.7843 ± 0.0148	0.8474 ± 0.0117	0.6973 ± 0.0250	0.6836 ± 0.0306
3.5	0.7682 ± 0.0640	0.8258 ± 0.0823	0.6799 ± 0.0543	0.6844 ± 0.0270
4.0	0.7538 ± 0.0787	0.8168 ± 0.1087	0.6478 ± 0.0938	0.6420 ± 0.0510
4.5	0.7686 ± 0.0353	0.8271 ± 0.0316	0.6878 ± 0.0439	0.6889 ± 0.0435
5.0	0.7469 ± 0.0726	0.7991 ± 0.0981	0.6559 ± 0.0547	0.6997 ± 0.0327

Cuadro 4.5: Resultado de modelos con *majority voting* con la función de costo *focal-loss* usando el parámetro α

El proceso realizado en MV con *focal-loss*, fue muy similar al procedimiento con *gold standard*, con la única diferencia siendo las etiquetas utilizadas. Se utilizó el procedimiento indicado en la sección 3.5.2 para obtener los valores α_k de cada clase, con el objetivo de darle más peso a aquellas clases con menor representación en el conjunto de datos.

El rendimiento con *focal-loss* en MV usando el hiperparámetro α estuvo por debajo de la media a comparación de *gold standard*. La mayoría de modelos dieron resultados peores, en algunos casos con una diferencia en la exactitud de hasta 0.05 (como es el caso de $\gamma = 5.0$).

γ	Accuracy	F1-Score class 1	F1-Score class 2	F1-Score class 3
1.0	0.8190 ± 0.0110	0.8720 ± 0.0106	0.7350 ± 0.0113	0.7242 ± 0.0532
1.5	0.8060 ± 0.0175	0.8607 ± 0.0168	0.7221 ± 0.0156	0.7115 ± 0.0413
2.0	0.7968 ± 0.0559	0.8491 ± 0.0651	0.7142 ± 0.0433	0.7093 ± 0.0332
2.5	0.8103 ± 0.0127	0.8697 ± 0.0102	0.7093 ± 0.0394	0.7017 ± 0.0386
3.0	0.7996 ± 0.0261	0.8575 ± 0.0274	0.7077 ± 0.0307	0.6901 ± 0.0578
3.5	0.8113 ± 0.0080	0.8678 ± 0.0084	0.7199 ± 0.0171	0.7070 ± 0.0310
4.0	0.7961 ± 0.0248	0.8559 ± 0.0255	0.6977 ± 0.0453	0.6866 ± 0.0736
4.5	0.8000 ± 0.0136	0.8594 ± 0.0112	0.7047 ± 0.0246	0.6874 ± 0.0639
5.0	0.7889 ± 0.0333	0.8540 ± 0.0256	0.6899 ± 0.0338	0.6611 ± 0.1052

Cuadro 4.6: Resultado de modelos con *majority voting* con la función de costo *focal-loss* sin usar el parámetro α

Al usar la versión simplificada de *focal-loss* sin α, los resultados son mucho más comparables con la versión de *gold standard*, aunque siguen siendo generalmente menores por 0.01 o 0.02 en la exactitud.

Generalized categorical cross-entropy

q	Accuracy	F1-Score class 1	F1-Score class 2	F1-Score class 3
0.01	0.7610 ± 0.1627	0.7814 ± 0.2608	0.7028 ± 0.0915	0.6496 ± 0.2220
0.1	0.8148 ± 0.0070	0.8654 ± 0.0071	0.7372 ± 0.0094	0.7289 ± 0.0329
0.2	0.8123 ± 0.0112	0.8634 ± 0.0107	0.7355 ± 0.0082	0.7271 ± 0.0364
0.3	0.8076 ± 0.0080	0.8593 ± 0.0072	0.7350 ± 0.0098	0.7058 ± 0.0381
0.4	0.8100 ± 0.0121	0.8629 ± 0.0093	0.7385 ± 0.0121	0.7043 ± 0.0600
0.5	0.7623 ± 0.0619	0.8294 ± 0.0662	0.6706 ± 0.0846	0.5230 ± 0.0958
0.6	0.7027 ± 0.1511	0.7452 ± 0.2532	0.6315 ± 0.1054	0.3473 ± 0.2878
0.7	0.6734 ± 0.2043	0.6689 ± 0.3377	0.6347 ± 0.1181	0.4325 ± 0.2189
0.8	0.7392 ± 0.1193	0.7979 ± 0.1361	0.6881 ± 0.0921	0.4032 ± 0.2666
0.9	0.7436 ± 0.1586	0.7733 ± 0.2586	0.6652 ± 0.1045	0.5058 ± 0.1882

Cuadro 4.7: Resultados de modelos con *generalized categorical cross entropy*(GCCE) usando todo el conjunto de datos.

Para el caso de *generalized categorical cross entropy* (GCCE), los resultados variaron significativamente dependiendo del valor del hiperparámetro q. En particular, valores de q mayores a 0.4 y menores a 0.1, obtuvieron resultados promedios considerablemente peores en términos de exactitud y puntaje F1. Como se mencionó en la sección 2.7.2, el hiperparámetro q regula la resistencia al ruido de la función GCCE, por lo que se podría interpretar, que valores de $q > 0.4$ proporcionan mucha robustez y no permiten que el modelo capturé las relaciones entre los datos, mientras

que valores de $q < 0.1$ no son lo suficientemente resistentes al ruido presente en el conjunto de datos.

q	Accuracy	F1-Score class 1	F1-Score class 2	F1-Score class 3
0.01	0.7926 ± 0.0193	0.8464 ± 0.0182	0.7287 ± 0.0180	0.6221 ± 0.0190
0.1	0.8050 ± 0.0093	0.8608 ± 0.0083	0.7357 ± 0.0101	0.6109 ± 0.0119
0.2	0.8004 ± 0.0095	0.8563 ± 0.0093	0.7270 ± 0.0149	0.6260 ± 0.0139
0.3	0.7980 ± 0.0139	0.8523 ± 0.0150	0.7285 ± 0.0093	0.6295 ± 0.0128
0.4	0.7972 ± 0.0131	0.8515 ± 0.0126	0.7300 ± 0.0114	0.6184 ± 0.0169
0.5	0.8001 ± 0.0113	0.8557 ± 0.0120	0.7240 ± 0.0094	0.6218 ± 0.0097
0.6	0.7978 ± 0.0143	0.8515 ± 0.0149	0.7319 ± 0.0121	0.6188 ± 0.0159
0.7	0.8014 ± 0.0241	0.8595 ± 0.0130	0.7377 ± 0.0251	0.5463 ± 0.1829
0.8	0.8007 ± 0.0171	0.8565 ± 0.0178	0.7365 ± 0.0143	0.5837 ± 0.0489
0.9	0.7994 ± 0.0177	0.8580 ± 0.0107	0.7330 ± 0.0305	0.5514 ± 0.1843

Cuadro 4.8: Resultados de modelos con *generalized categorical cross entropy*(GCCE) usando los datos que fueron anotados por dos o más personas.

Al reducir el tamaño del conjunto de datos a aquellas muestras que tienen más de una anotación, la métrica de exactitud es bastante más consistente. Además, la métrica de puntaje F1 para las clases 1 y 2 es bastante consistente en todos los hiperparámetros. No obstante, al observar el puntaje F1 de clase 3, se nota que es considerablemente peor al comparar con el puntaje F1 de *gold standard* (0.7359), teniendo una diferencia aproximada de 0.1 hasta casi 0.2 en el peor de los casos. Esto es indicativo de que el conjunto de datos reducido, no contiene suficientes muestras para representar las características de la clase 3 (inflamación benigna).

Sin tomar en cuenta los resultados de la clase 3, un conjunto de datos reducido (2140 muestras) es capaz de obtener resultados comparables al conjunto original (75243 muestras), además de ser entrenado considerablemente más rápido. Durante el entrenamiento de GCCE, utilizando todo el conjunto de datos, se observaron tiempos aproximados de 32 segundos por época, mientras que usando el conjunto reducido, se observaron entrenamientos en promedio de 2 segundos por época, esta es una diferencia importante de 30 segundos por época.

Esto quiere decir, que siguiendo la metodología usada en el proyecto de 20 épocas por modelo, 10 repeticiones del entrenamiento de cada modelo, y un espacio de búsqueda de hiperparámetros de 10, se gastó un total de $32 * 20 * 10 * 10 = 64000$ segundos, o aproximadamente 18 horas de entrenamiento. A diferencia de $2*20*10*10 = 4000$ segundos, o poco más de una hora. Estos tiempos corresponden a entrenar en Google Colab (sin GPU) con un Intel(R) Xeon(R) CPU @ 2.20GHz de un núcleo.

Este tipo de diferencia de tiempos en el entrenamiento son altamente importantes, sobre todo cuando se trabaja con recursos gratuitos como Google Colab, que limitan el tiempo de uso continuo, lo que resulta en realizar los entrenamientos a lo largo de múltiples días.

Si bien es útil poder entrenar modelos rápidamente, es posible que los datos no sean suficientes para representar el problema que se desea predecir, lo que ocasione un subajuste en los modelos resultantes, como lo fue el resultado de la clase 3.

Crowdlayer

Conn type	Accuracy	F1-Score class 1	F1-Score class 2	F1-Score class 3
MW	0.7514 ± 0.0346	0.8282 ± 0.0251	0.6754 ± 0.0398	0.4990 ± 0.1613
SW	0.8184 ± 0.0090	0.8671 ± 0.0093	0.7430 ± 0.0104	0.7428 ± 0.0188
VB	0.8176 ± 0.0183	0.8662 ± 0.0154	0.7420 ± 0.0214	0.7533 ± 0.0289
VW	0.7654 ± 0.0347	0.8087 ± 0.0401	0.7005 ± 0.0313	0.7342 ± 0.0235
VW+B	0.8022 ± 0.0238	0.8523 ± 0.0238	0.7343 ± 0.0205	0.7215 ± 0.0253

Cuadro 4.9: Resultados de modelos con *Crowdlayer* usando todo el conjunto de datos.

En el caso de *Crowdlayer*, el hiperparámetro "conn type" representa el tipo de representación de los pesos y sesgos de cada anotador. Se puede observar que, en promedio, los valores de MW (matriz de pesos) y VW (vector de pesos), obtuvieron resultados significativamente peores en comparación al resto. En particular, SW (peso único por anotador) y VB (vector de sesgos), obtuvieron resultados promedios bastante similares a *gold standard*, tanto en la exactitud de las predicciones como en los puntajes F1 de cada clase.

Conn type	Accuracy	F1-Score class 1	F1-Score class 2	F1-Score class 3
MW	0.7965 ± 0.0178	0.8498 ± 0.0189	0.7299 ± 0.0162	0.6259 ± 0.0140
SW	0.8023 ± 0.0154	0.8563 ± 0.0153	0.7339 ± 0.0142	0.6311 ± 0.0122
VB	0.8051 ± 0.0075	0.8595 ± 0.0078	0.7337 ± 0.0090	0.6372 ± 0.0144
VW	0.7964 ± 0.0175	0.8511 ± 0.0173	0.7305 ± 0.0180	0.6192 ± 0.0092
VW+B	0.7968 ± 0.0098	0.8525 ± 0.0099	0.7228 ± 0.0125	0.6223 ± 0.0157

Cuadro 4.10: Resultados de modelos con *Crowdlayer* usando los datos que fueron anotados por dos o más personas.

Los resultados con el conjunto de datos reducido, obtuvieron un comportamiento similar a GCCE, donde los resultados son más consistentes en todos los hiperparametros, sin embargo, se repite el mismo rendimiento deficiente de la clase 3 al comparar el puntaje F1 con *gold standard*, teniendo una diferencia promedio aproximada de 0.1.

4.2. Comparación entre modelos

En la sección previa, se observaron distintos comportamientos y patrones en los resultados de rendimiento de cada modelo. Si bien es cierto que se realizó el entrenamiento y evaluación de cada modelo diez veces para evitar la influencia de la aleatoriedad, todavía es posible que haya cierto nivel de aleatoriedad en los resultados.

Las pruebas estadísticas, como la prueba de Friedman, permite identificar si hay diferencias estadísticamente significativas en el rendimiento de los modelos evaluados. Esto es útil para evitar decisiones basadas en variaciones aleatorias del rendimiento. Además, al utilizar un método estadístico robusto, se asegura una mayor objetividad en la selección del modelo más adecuado, minimizando el sesgo que podría introducirse al comparar los modelos de manera informal o intuitiva. En esta sección se realizará la prueba de Friedman para comparar el rendimiento de los múltiples modelos entrenados en este proyecto.

Esta prueba requiere de al menos 3 grupos, cada uno con la misma cantidad de muestras. En el caso de este proyecto, un grupo estará representado por un modelo específico, y las exactitudes de las evaluaciones de cada uno de los 10 entrenamientos representan las muestras presentes de ese grupo. Por ejemplo, el modelo que usa GCCE con hiperparámetro $q = 0.1$, representa un grupo, y las exactitudes de cada uno de los 10 entrenamientos, son las 10 muestras que representan este grupo.

Majority Voting

En esta prueba, se comparan todos los modelos que utilizaron la votación por mayoría. Es decir, el único modelo con entropía cruzada, nueve modelos que utilizan *focal-loss* con el hiperparámetro α, y nueve modelos con *focal-loss* sin el hiperparámetro α.

Hipótesis nula: No hay una diferencia estadísticamente significativa entre las exactitudes de los modelos que utilizaron la votación por mayoría.

Resultado: El *p-value* (0.0000) está por debajo del valor de significancia del 1 %, por lo que rechazamos la hipótesis nula. Se concluye que hay una diferencia significativa entre las exactitudes de los modelos que utilizaron la votación por mayoría. Esto indica que hay diferencias entre las diferentes configuraciones de *majority voting* realizadas, es decir, entropía cruzada, perdida focal con α y sin α.

El modelo que usó la técnica de *majority voting* con la función de costo *cross-entropy* obtuvo los mejores resultados en términos de exactitud (0.8206 ± 0.0106) y puntajes F1 para todas las clases (0.8706 ± 0.0096 para la clase 1, 0.7425 ± 0.0137 para la clase 2, y 0.7422 ± 0.0253 para la clase 3). En comparación, los modelos que emplearon *focal-loss* tanto con el parámetro α como sin él mostraron un rendimiento inferior en todas las métricas. Esto sugiere que la función de costo *cross-entropy* es más efectiva para el esquema de *majority voting* en este contexto

particular, proporcionando un equilibrio más robusto entre la exactitud general y la capacidad de capturar correctamente cada clase individual.

Si bien los resultados de *cross-entropy* fueron superiores, el rendimiento de *focal-loss* no estuvo significativamente por detrás en algunos casos. Por ejemplo, con $\gamma = 1$ se obtuvo una exactitud de 0.8190 ± 0.0110, y puntajes F1 de 0.8720 ± 0.0106, 0.7350 ± 0.0113 y 0.7242 ± 0.0532 para las clases 1, 2 y 3 respectivamente. Es posible que utilizando una arquitectura de red neuronal o conjuntos de datos diferentes, la función de *focal-loss* proporciones resultados superiores a *cross-entropy*.

Crowdlayer (CL)

En esta prueba, se compararán los modelos de *Crowdlayer* (CL) con el conjunto de datos original, y el conjunto de datos reducido que usa al menos 2 anotadores por muestra (CLM2). Esto resulta en un total de 10 modelos para comparar.

Hipótesis nula: No hay una diferencia estadísticamente significativa entre las exactitudes de los modelos basados en *Crowdlayer*.

Resultado: El *p-value* (0.0000) es menor que el nivel de significancia ($\alpha = 0.01$), por lo que rechazamos la hipótesis nula. Se concluye que existe una diferencia significativa entre las exactitudes de los modelos de CL y CLM2. Además, hay algunos puntos importantes a destacar:

- Aunque los modelos *Crowdlayer* (CL) y *Crowdlayer* M2 (CLM2) muestran exactitudes generales similares, hay una diferencia notable en la exactitud con los mismos hiperparámetros. Por ejemplo, el modelo CL con el hiperparámetro "VW+B" tuvo la mayor exactitud media entre todos los demás modelos CL. En contraste, con CLM2, el modelo con el hiperparámetro "VW+B" tuvo la menor exactitud media entre todos los demás modelos *Crowdlayer* M2.
- CL y CLM2 representan los mismos modelos, con la única diferencia de que el conjunto de datos utilizado para entrenar estos modelos es distinto. Los modelos M2 se entrenaron en un subconjunto del conjunto de datos original, específicamente, solo en las entradas de datos que fueron anotadas por al menos dos personas. Esto sugiere que el número de entradas en el conjunto de datos puede causar diferencias notables en el rendimiento de los modelos que usan *Crowdlayer*.

Comparando los resultados, se observa que el modelo entrenado con Crowdlayer usando todo el conjunto de datos generalmente tuvo un mejor desempeño en términos de exactitud (accuracy) y puntajes F1 para las clases 1 y 2.

El hiperparámetro SW mostró la mayor exactitud ("accuracy") de 0.8184 ± 0.0090 y los mejores puntajes F1 para la clase 1 (0.8671 ± 0.0093) y clase 2 (0.7430 ± 0.0104) usando todo el conjunto de datos. Sin embargo, para la clase 3, el hiperparámetro VB obtuvo el mejor puntaje F1 (0.7533 ± 0.0289) usando todo el conjunto de datos.

Cuando se usan los datos anotados por dos o más personas, el hiperparámetro VB mostró una exactitud decente (0.8051 ± 0.0075) y el mejor puntaje F1 para la clase 1 (0.8595 ± 0.0078) y la clase 3 (0.6372 ± 0.0144).

En resumen, el hiperparámetro SW usando todo el conjunto de datos ofrece el mejor equilibrio entre exactitud y puntajes F1 para las clases. Esto sugiere que SW es un hiperparámetro robusto para modelos generales, aunque esto puede variar dependiendo del conjunto de datos y la arquitectura del modelo utilizado.

Generalized categorical cross-entropy (GCCE)

En esta prueba, se comparan los modelos que usan la función de costo "Generalized categorical cross-entropy", específicamente, se compararan modelos que usan únicamente los datos anotados por dos o más personas (GCCEM2) contra los modelos que usan todo el conjunto de datos(GCCE). Esto resulta en un total de 20 grupos, cada uno con 10 exactitudes.

Hipótesis nula: No hay una diferencias estadísticamente significativas entre las exactitudes de los modelos basados en GCCE.

Resultado: El valor *p-value* (0.0142) es mayor que el nivel de significancia ($\alpha = 0.01$), por lo que no rechazamos la hipótesis nula. Estos valores son cercanos y hay algunas cosas que vale la pena señalar sobre estos modelos:

- Al igual que en los modelos Crowdlayer, la única diferencia entre GCCE y GCCEM2 es el conjunto de datos utilizado para entrenar los modelos. Este cambio muestra algunas diferencias interesantes en los resultados de ambos modelos.
- La versión GCCEM2 obtuvo resultados significativamente más consistentes en todos los hiperparámetros, con una exactitud media de 0.7992 y una desviación estándar de 0.0158. En contraste, al usar el conjunto de datos completo, la exactitud media fue de 0.7626 con una desviación estándar de 0.1251.
- El hiperparámetro q tuvo un impacto mucho mayor al usar el conjunto de datos completo. En particular, los valores de q mayores a 0.4 y menores a 0.1 tuvieron un rendimiento peor.
- El rendimiento del modelo GCCEM2 tuvo un rendimiento notablemente menor en el puntaje F1 de la clase 3.

Comparando los resultados, se observa que el modelo entrenado con GCCE usando todo el conjunto de datos generalmente tuvo un mejor desempeño. Específicamente, para $q = 0.1$, se obtuvo la mayor exactitud ("accuracy") de 0.8148 ± 0.0070 y los mejores puntajes F1 para la clase 1 (0.8654 ± 0.0071) y clase 3 (0.7289 ± 0.0329). Aunque para la clase 2, el valor $q = 0.4$ obtuvo un puntaje F1 ligeramente superior (0.7385 ± 0.0121), la diferencia no es significativa en comparación con los otros valores q.

En resumen, el modelo que utiliza $q = 0.1$ entrenado con todo el conjunto de datos ofrece el mejor equilibrio entre exactitud y puntajes F1 para las diferentes clases. Este resultado sugiere que un valor $q = 0.1$ en la función de costo GCCE es el más adecuado para este conjunto de datos, y el entrenamiento con todos los datos proporciona un modelo más robusto, a costo de un mayor tiempo de entrenamiento.

Generalized categorical cross-entropy y Crowdlayer

En esta prueba, se comparan los modelos que utilizan múltiples anotadores sin votación por mayoría, "Generalized Categorical Cross-entropy" (GCCE) y Crowdlayer (CL).

Hipótesis nula: No hay una diferencia estadísticamente significativa entre las exactitudes de los modelos que utilizan múltiples anotadores, exceptuando votación por mayoría.

Resultado: El *p-value* (0.0000) es menor que α 0.01, por lo que rechazamos la hipótesis nula. Se concluye que existe una diferencia significativa entre las exactitudes de los modelos. Esto indica que hay evidencias de que las estrategias de GCCE y CL presentan diferencias significativas.

Comparando los resultados de todas las tablas, se observa que Crowdlayer usando todo el conjunto de datos (tabla 4.9) tiene la mejor exactitud (accuracy) general con 0.8184 ± 0.0090. Este modelo también muestra los mejores puntajes F1 para la clase 1 con 0.8671 ± 0.0093 y para la clase 2 con 0.7430 ± 0.0104. Sin embargo, para la clase 3, el mejor puntaje F1 se observa en el modelo de Crowdlayer con conexión VB usando todo el conjunto de datos (Tabla 3), con un F1-score de 0.7533 ± 0.0289.

En general, el modelo de Crowdlayer con hiperparámetro SW usando todo el conjunto de datos (tabla 4.9) es el mejor en términos de exactitud y equilibrio entre las clases. Este modelo ofrece el mejor rendimiento general, especialmente para las clases 1 y 2.

Modelos con múltiples anotadores

En esta prueba, se comparan las 10 exactitudes de cada uno de los modelos que usan múltiples anotadores (MV, GCCE y *Crowdlayer*). Esto resulta en un total de 49 grupos por comparar.

Hipótesis nula: No hay una diferencia estadísticamente significativa entre las exactitudes de todos los modelos que utilizan múltiples anotadores, incluyendo la votación por mayoría.

Resultado: El *p-value* (0.0000) está por debajo del valor de significancia del 1 %, por lo que se rechaza la hipótesis nula. Se concluye que hay una diferencia significativa entre las exactitudes de los modelos que usan múltiples anotadores. Esto es indicativo de que los distintos métodos utilizados presentan diferencias entre sí, y es de interés revisar a fondo los resultados obtenidos.

Entre todos los modelos evaluados en el desarrollo de este proyecto, el modelo de *majority voting* utilizando la función de costo *cross-entropy* (tabla 4.4) muestra el mejor rendimiento global con una exactitud de 0.8206 ± 0.0106 y puntajes F1 equilibrados. Aunque algunos modelos específicos con diferentes configuraciones de GCCE y Crowdlayer se aproximan a estos resultados, ninguno logra superarlo.

Sin embargo, el *majority voting* puede no ser el método más confiable para manejar múltiples anotadores, puesto que asume que todos los anotadores son igualmente confiables y no poseen sesgos significativos. Como se explicó en la sección 3.5.2, existen modificaciones de esta técnica que buscan tomar en cuenta la experiencia, conocimiento y sesgos de cada anotador, pero determinar estos factores es un reto por sí mismo. Las estrategias de *generalized categorical cross-entropy* y *crowdlayer* solucionan esta dificultad al calcular los pesos y sesgos de cada anotador durante el entrenamiento.

En conclusión, es importante tomar en cuenta las características del conjunto de datos cuando se trabaja con múltiples anotadores. Si bien es posible que **majority voting** proporcione resultados competentes, es importante explorar otros métodos que tomen en cuenta la posibilidad de presencia de ruido o sesgos en las anotaciones.

Capítulo 5

Conclusiones

El objetivo final de este proyecto fue realizar diferentes modelos de aprendizaje automático que permitieran la detección de cáncer de seno en imágenes histológicas. Para esto, se buscó que algunos de estos modelos pudieran aprender de las etiquetas propuestas por múltiples personas con diferentes grados de conocimiento, aprovechando el saber colectivo para generar las distintas clasificaciones de estas imágenes.

Para lograr un buen aprendizaje de los modelos mediante la extracción de características relevantes de las imágenes histológicas, se requirió un conjunto de datos adecuado para la investigación propuesta. En este sentido, estudios como el descrito en la sección 3.1 y el artículo "Deep Gaussian Processes for Classification With Multiple Noisy Annotators. Application to Breast Cancer Tissue Classification"[15] fueron de gran utilidad, ya que proporcionaron un punto de partida sólido para el desarrollo de los modelos estudiados en este proyecto.

Es importante destacar que el proyecto se enmarca en un problema de clasificación de imágenes, lo que requiere la utilización de herramientas especializadas para llevar a cabo esta tarea. En consecuencia, los modelos empleados en este proyecto se basan en características relevantes extraídas previamente por una red neuronal convolucional (CNN), específicamente una VGG16 preentrenada, mediante la técnica de aprendizaje por transferencia (transfer learning).

En las primeras etapas del desarrollo, se evidenció que la combinación de la red neuronal propuesta (figura 3.5) con la VGG16 generaba un aumento considerable en los tiempos de entrenamiento. Esto se debe a que en cada iteración, la red convolucional debía realizar el proceso de extracción de características para cada imagen, impactando negativamente tanto en el tiempo de ejecución como en la cantidad de recursos computacionales requeridos. Para abordar esta problemática, se optó por realizar la extracción de características de todo el conjunto de datos de forma previa y luego utilizar estas características preprocesadas para alimentar a los diferentes modelos durante el entrenamiento. Esta estrategia permitió optimizar significativamente el tiempo de entrenamiento de los modelos.

Un aspecto fundamental en el desarrollo de los modelos propuestos en este proyecto fue el diseño de funciones de costo personalizadas para abordar diversos

desafíos. Estas funciones de costo se centraron en la solución de dos problemáticas principales:

- **Manejo de muestras difíciles de clasificar:** La función de costo "Focal Loss" (Sección 2.7.3) se diseñó para abordar el problema de las muestras difíciles de clasificar, donde algunos ejemplos presentan un desequilibrio significativo en las clases. Esta función asigna mayor peso a las muestras mal clasificadas, dirigiendo el proceso de aprendizaje hacia una mejor discriminación entre clases.
- **Manejo de múltiples anotadores:** La función de costo "Generalized Categorical Cross Entropy" (Sección 2.7.2) se desarrolló para permitir el aprendizaje a partir de anotaciones proporcionadas por múltiples individuos con diferentes niveles de conocimiento. Esta función incorpora un parámetro de ponderación que modula la contribución de cada anotador, permitiendo que el modelo aprenda de manera más efectiva de las anotaciones confiables y minimice el impacto de las anotaciones erróneas.

El diseño y la implementación de estas funciones de costo personalizadas presentaron diversos desafíos que requirieron un esfuerzo considerable para su resolución. Un caso particular fue el desarrollo de la función "Focal Loss". Al implementarla utilizando la librería TensorFlow, se observó un comportamiento diferente del esperado, a pesar de la correcta implementación matemática. Para solucionar este problema, se optó por utilizar la función categorical cross-entropy de TensorFlow y posteriormente realizar las modificaciones necesarias para transformarla en "Focal Loss".

Las funciones de costo personalizadas desempeñaron un papel crucial en el éxito de los modelos propuestos. Al abordar los desafíos específicos del problema de clasificación de imágenes histológicas para la detección de cáncer de seno, estas funciones permitieron el desarrollo de modelos con rendimientos similares o comparables al punto de referencia que es el modelo "gold standard".

Tal como se menciona al inicio de esta sección, el proyecto se enfocó en el desarrollo de modelos de aprendizaje automático para la detección de cáncer de seno en imágenes histológicas. Estos modelos tenían la particularidad de poder aprovechar la opinión de múltiples individuos con distintos niveles de conocimiento. Para evaluar su desempeño, se compararon con un modelo entrenado en condiciones ideales, es decir, utilizando las etiquetas verdaderas de las imágenes como referencia. Gracias a esta comparación hay evidencia de que es posible crear modelos de aprendizaje automático que cumplan con las características antes mencionadas.

Los resultados del proyecto indican que los modelos que integran anotaciones de múltiples individuos alcanzaron un rendimiento similar al del modelo entrenado en condiciones ideales, donde se utiliza un conjunto de datos etiquetado por un solo experto. En algunos casos, incluso superaron ligeramente al punto de referencia ("gold standard"), como es el caso de modelos con "Crowdlayer" dependiendo del hiperparámetro usado. Esto sugiere que la diversidad de perspectivas en las anotaciones puede contribuir a mejorar la capacidad del modelo para identificar correctamente

el cáncer de seno.

Si bien los resultados obtenidos en este proyecto son prometedores, existen multitud de estrategias que se pueden explorar para optimizar o mejorar los modelos desarrollados en esta investigación. En la siguiente sección se propone el trabajo futuro donde se pueden evidenciar algunas de las estrategias que, a pesar de no ser abarcadas en el desarrollo de este proyecto, pueden ser tenidas en cuenta para la optimización y mejora de los modelos propuestos.

5.1. Trabajo futuro

- **Optimización adicional de hiperparámetros**
  En el presente proyecto, se ha implementado una búsqueda de hiperparámetros enfocada en ajustar los parámetros específicos de las funciones de costo, como γ para la focal loss y q para la Generalized Categorical Cross Entropy (GCCE). Sin embargo, existen múltiples hiperparámetros que pueden ser optimizados para mejorar el rendimiento de los modelos presentados, tales como la tasa de aprendizaje, el tamaño de lote, el número de capas y la cantidad de neuronas por capa. Además, parámetros de regularización como el dropout y la penalización L2 también pueden ser ajustados para evitar el sobreajuste. Por último, se pueden probar diferentes funciones de activación, que influyen en cómo se propagan las señales a través de la red.

- **Mejora de la complejidad de la red neuronal**
  Como se puede evidenciar en la figura 3.5 que representa la arquitectura de la red neuronal usada en los modelos del proyecto, se trata de una red neuronal simple que puede hacerse más compleja. Para aumentar la complejidad de la red, se pueden añadir más capas ocultas, lo que permite a la red aprender representaciones más profundas y abstractas de los datos. Además, incrementar el número de neuronas por capa puede ayudar a capturar más características del conjunto de datos.

- **Segmentación semántica**
  Se puede probar otra forma de clasificación de las imágenes mediante la segmentación semántica de las imágenes histológicas, que permita identificar y mostrar las áreas correspondientes a las clases del proyecto (tumor, estroma, inflamación benigna). La segmentación semántica es una técnica que asigna una etiqueta a cada píxel de una imagen, diferenciando las distintas regiones presentes. Esto proporciona una comprensión más detallada y precisa de la estructura y composición de las imágenes. Para este propósito, se puede implementar una arquitectura como UNET, que es particularmente efectiva en tareas de segmentación semántica. UNET utiliza una estructura de encoder-decoder que permite capturar tanto las características globales como los detalles finos de las imágenes, proporcionando segmentaciones precisas y detalladas de las diferentes áreas de interés.

- **Prevención del sobreajuste y subajuste**

En este proyecto, no se utilizó ninguna técnica, salvo el uso de dropout en la red neuronal, para evitar el sobreajuste o subajuste. Sin embargo, existen diversas técnicas que pueden implementarse para mitigar estos problemas. Para prevenir el sobreajuste, además del dropout, se pueden utilizar técnicas como la regularización L2 y el aumento de datos (data augmentation), que incrementa la diversidad del conjunto de datos mediante la aplicación de transformaciones aleatorias a las imágenes de entrenamiento. Por otro lado, para prevenir el subajuste, se pueden usar técnicas como aumentar la complejidad de los modelos, añadiendo más capas y neuronas, o buscar optimizar los parámetros de la red para que el modelo pueda aprender correctamente de los datos.

- **Técnicas de balanceo de muestras**
  Implementar técnicas de balanceo de muestras, como el sobremuestreo (oversampling) o el submuestreo (subsampling), puede ser crucial para mejorar el rendimiento del modelo en conjuntos de datos desbalanceados. El sobremuestreo aumenta el número de ejemplos de las clases minoritarias replicando las muestras existentes o generando nuevas muestras sintéticas. Por otro lado, el submuestreo reduce el número de ejemplos de las clases mayoritarias, equilibrando así el conjunto de datos y permitiendo que el modelo trate por igual todas las clases.

- **Implementar técnicas para mejorar el majority voting**
  Como se mencionó en la sección 3.5.2, los empates en el majority voting (MV) se solucionaron de una manera estándar, eligiendo simplemente la siguiente clase en orden. Sin embargo, para mejorar la robustez y precisión del proceso de votación en la presencia de múltiples anotadores, se pueden proponer nuevas formas de manejar estos empates. Una estrategia podría ser ponderar los votos de los anotadores en función de su confiabilidad o experiencia previa, asignando mayor peso a los votos de anotadores con un historial más consistente o experticia en el dominio específico. Otra posibilidad sería implementar un sistema de votación donde los empates se resuelvan mediante la consulta de un grupo adicional de anotadores o mediante técnicas de voto ponderado.

Bibliografía

[1] "GLOBOCAN 2020: Global cancer observatory," https://gco.iarc.fr/.

[2] M. Amgad, H. Elfandy, H. Hussein, L. A. Atteya, M. A. T. Elsebaie, L. S. Abo Elnasr, R. A. Sakr, H. S. E. Salem, A. F. Ismail, A. M. Saad, J. Ahmed, M. A. T. Elsebaie, M. Rahman, I. A. Ruhban, N. M. Elgazar, Y. Alagha, M. H. Osman, A. M. Alhusseiny, M. M. Khalaf, A.-A. F. Younes, A. Abdulkarim, D. M. Younes, A. M. Gadallah, A. M. Elkashash, S. Y. Fala, B. M. Zaki, J. Beezley, D. R. Chittajallu, D. Manthey, D. A. Gutman, and L. A. D. Cooper, "Structured crowdsourcing enables convolutional segmentation of histology images," *Bioinformatics*, vol. 35, no. 18, pp. 3461–3467, 02 2019. [Online]. Available: https://doi.org/10.1093/bioinformatics/btz083

[3] R. Shyam, "Convolutional neural network and its architectures," vol. 12, p. 2021, 10 2021.

[4] A. Mavrin, "Focal loss for dense object detection (github repository)." [Online]. Available: https://github.com/artemmavrin/focal-loss

[5] Centers for Disease Control and Prevention, "Leading causes of death in females - all races and origins, United States, 2017," Available online: https://www.cdc.gov/women/lcod/2017/all-races-origins/index.htm, 2017, [Online]. Accessed on: Mar. 20, 2024.

[6] W. C. R. F. International, "Worldwide cancer data," Available online: https://www.wcrf.org/cancer-trends/worldwide-cancer-data/, 2023, accessed on: Mar. 20, 2024.

[7] A. C. Society, "Breast cancer early detection and diagnosis," https://www.cancer.org/content/dam/CRC/PDF/Public/8579.00.pdf, 2020.

[8] R. B. Marilys Corbex and H. Sancho-Garnier, "Breast cancer early detection methods for low and middle income countries, a review of the evidence," *The Breast*, vol. 21, no. 4, pp. 428–434, 2012. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0960977612000033

[9] N. C. Institute, "Using tcga data," https://www.cancer.gov/ccg/research/genome-sequencing/tcga/using-tcga-data, Accessed 2023.

[10] W. H. Organization, "Cancer screening and early detection of cancer," accessed: May 30, 2024. [Online]. Available: https://www.who.int/europe/news-room/fact-sheets/item/cancer-screening-and-early-detection-of-cancer

[11] A. C. Society, "American cancer society recommendations for the early detection of breast cancer," accessed: Jan 17, 2024. [Online]. Available: https://www.cancer.org/cancer/types/breast-cancer/screening-tests-and-early-detection/american-cancer-society-recommendations-for-the-early-detection-of-breast-cancer.html

[12] ——, "Breast ultrasound," accessed: May 30, 2024. [Online]. Available: https://www.cancer.org/cancer/types/breast-cancer/screening-tests-and-early-detection/breast-ultrasound.html

[13] J.-M. Chen, Y. Li, J. Xu, L. Gong, L.-W. Wang, W.-L. Liu, and J. Liu, "Computer-aided prognosis on breast cancer with hematoxylin and eosin histopathology images: A review," *Journal of International Medical Research*, vol. 45, no. 3, pp. 718–738, 2017. [Online]. Available: https://journals.sagepub.com/doi/pdf/10.1177/1010428317694550

[14] A. Janowczyk and A. Madabhushi, "Deep learning for digital pathology image analysis: A comprehensive tutorial with selected use cases," *Journal of Pathology Informatics*, vol. 7, no. 1, p. 29, 2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2153353922005478

[15] M. López-Pérez, P. Morales-Álvarez, L. A. D. Cooper, R. Molina, and A. K. Katsaggelos, "Deep gaussian processes for classification with multiple noisy annotators. application to breast cancer tissue classification," *IEEE Access*, vol. 11, pp. 6922–6934, 2023.

[16] Q. B. Baker, T. A. Zaitoun, S. Banat, E. Eaydat, and M. Alsmirat, "Automated detection of benign and malignant in breast histopathology images," in *2018 IEEE/ACS 15th International Conference on Computer Systems and Applications (AICCSA)*, 2018, pp. 1–5.

[17] M. López-Pérez, M. Amgad, P. Morales-Álvarez, P. Ruiz, L. A. D. Cooper, R. Molina, and A. K. Katsaggelos, "Learning from crowds in digital pathology using scalable variational gaussian processes," Jun 2021. [Online]. Available: https://www.nature.com/articles/s41598-021-90821-3

[18] A. C. Society, "What is breast cancer?" 2023. [Online]. Available: https://www.cancer.org/cancer/types/breast-cancer/about/what-is-breast-cancer.html

[19] ——, "How common is breast cancer?" 2023. [Online]. Available: https://www.cancer.org/cancer/types/breast-cancer/about/how-common-is-breast-cancer.html

[20] A. Stevens and J. Lowe, *Histología humana*, 5th ed. Elsevier, 2023.

[21] G. Musumeci, "Past, present and future: overview on histology and histopathology," *Journal of Histology and Histopathology*, 2014.

[22] V. C. Raykar, S. Yu, L. H. Zhao, G. H. Valadez, C. Florin, L. Bogoni, and L. Moy, "Learning from crowds," *Journal of Machine Learning Research*, vol. 11, no. 43, pp. 1297–1322, 2010. [Online]. Available: http://jmlr.org/papers/v11/raykar10a.html

[23] F. Rodrigues and F. Pereira, "Deep learning from crowds," 2017.

[24] S. J. Prince, *Understanding Deep Learning*. MIT press, 2023.

[25] NetApp, "What is deep learning?" https://www.netapp.com/es/artificial-intelligence/what-is-deep-learning/, Accessed 2023.

[26] F. C. Zegarra, J. Vargas-Machuca, and A. M. Coronado, "Comparison of cnn and cnn-lstm architectures for tool wear estimation," in *2021 IEEE Engineering International Research Conference (EIRCON)*, 2021, pp. 1–4.

[27] J. Hao, "Deep learning-based medical image analysis with explainable transfer learning," in *2023 International Conference on Computer Engineering and Distance Learning (CEDL)*, 2023, pp. 106–109.

[28] K. O'Shea and R. Nash, "An introduction to convolutional neural networks," 2015.

[29] N. Ketkar and J. Moolayil, *Convolutional Neural Networks*. Berkeley, CA: Apress, 2021, pp. 197–242. [Online]. Available: https://doi.org/10.1007/978-1-4842-5364-9_6

[30] H. Wang, "Garbage recognition and classification system based on convolutional neural network vgg16," in *2020 3rd International Conference on Advanced Electronic Materials, Computers and Software Engineering (AEMCSE)*, 2020, pp. 252–255.

[31] D.-B. Wang, Y. Wen, L. Pan, and M.-L. Zhang, "Learning from noisy labels with complementary loss functions," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 11, pp. 10 111–10 119, May 2021. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/17213

[32] J. C. Triana-Martinez, J. Gil-González, J. A. Fernandez-Gallego, A. M. Álvarez Meza, and C. G. Castellanos-Dominguez, "Chained deep learning using generalized cross-entropy for multiple annotators classification," *Sensors*, vol. 23, no. 7, 2023. [Online]. Available: https://www.mdpi.com/1424-8220/23/7/3518

[33] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," 2018.

[34] Kitware, "Histomicstk demo: Interactive web-based annotation tool for histopathology," https://demo.kitware.com/histomicstk/histomicstk#?image=5bbdee62e629140048d01b0d&bounds=-57297%2C291%2C142747%2C84059%2C0, 2024, accessed: 2024-05-28.

[35] H. E. Z. Reyes, H. P. Barreto, J. A. C. Ramos, and G. Del Carmen López Armas, "Comparative analysis of stained normalization in he histopathological images

of breast cancer for nuclei segmentation improvement," in *2023 IEEE International Autumn Meeting on Power, Electronics and Computing (ROPEC)*, vol. 7, 2023, pp. 1–6.

[36] E. Reinhard, M. Adhikhmin, B. Gooch, and P. Shirley, "Color transfer between images," *IEEE Computer Graphics and Applications*, vol. 21, no. 5, pp. 34–41, 2001.

[37] M. Friedman, "The use of ranks to avoid the assumption of normality implicit in the analysis of variance," *Journal of the American Statistical Association*, vol. 32, no. 200, pp. 675–701, 1937. [Online]. Available: https://www.tandfonline.com/doi/abs/10.1080/01621459.1937.10503522