**UNIVERSIDAD TECNICA FEDERICO SANTA MARIA**

**Repositorio Digital USM** **https://repositorio.usm.cl**

Tesis USM TESIS de Pregrado de acceso ABIERTO

2021-07

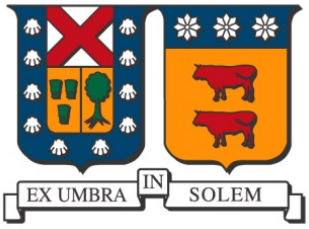
# CLASIFICACIÓN DE IMÁGENES CON SOBREEXPRESIÓN DE LA PROTEÍNA HER2 USANDO BAYESIAN DEEP LEARNING

BÓRQUEZ GONZÁLEZ, SEBASTIÁN IGNACIO

https://hdl.handle.net/11673/50623

*Repositorio Digital USM, UNIVERSIDAD TECNICA FEDERICO SANTA MARIA*

UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
DEPARTAMENTO DE INFORMÁTICA
VALPARAÍSO - CHILE




# “CLASIFICACIÓN DE IMÁGENES CON SOBREEXPRESIÓN DE LA PROTEÍNA HER2 USANDO BAYESIAN DEEP LEARNING”

SEBASTIÁN IGNACIO BÓRQUEZ GONZÁLEZ

MEMORIA PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL EN INFORMÁTICA

Profesor Guía: Dra. Raquel Pezoa Rivera
Profesor Correferente: Dr. Claudio Torres López

Julio - 2021

## DEDICATORIA

A mi madre, por su esfuerzo en velar por mi educación, y ser la constante fuente de motivación en mi crecimiento profesional. A mi padre, por su apoyo incondicional, el cual me ha acompañado a lograr cada una de mis metas. Porque les debo toda mi vida, les dedico este trabajo.

# AGRADECIMIENTOS

Agradezco a todos aquellos que me han acompañado durante estos años universitarios. A mi familia, cuyo cariño me ha acompañado desde la distancia de mi hogar. A los amigos y amigas que han llenado mi historia con sus recuerdos. A mis profesoras y profesores, quienes marcaron mi trayectoria profesional, en especial a Raquel Pezoa, por su confianza y apoyo académico, y con quien ha sido una grata experiencia trabajar.

Este trabajo fue parcialmente financiado por CCTVal ANID PIA/APOYO AFB180002, y FONDECYT Postdoc N° 3190740.

# RESUMEN

**Resumen—** La proteína *human epidermal growth factor receptor* 2 (HER2) es un biomarcador utilizado en muestras histológicas para el diagnóstico de distintos tipos de cáncer. *HER2 scoring*, es una tarea crucial realizada por los patólogos, que permite determinar si el paciente es apto para recibir la terapia anti-HER2. Sin embargo, este análisis es semi-cuantitativo y propenso a errores. Avances en la digitalización de muestras de histopatología, han permitido el desarrollo de herramientas computacionales para asistir en el proceso de evaluación. Tarea no trivial, debido al enorme tamaño de las imágenes, y a las diversas fuentes de ruido durante su digitalización. En esta memoria, se implementan algoritmos de análisis de imágenes, basado en el enfoque *bayesian deep learning*, para la asistencia en el diagnóstico histopatológico de la sobreepresión de la proteína HER2. Contribuyendo con los modelos de incertidumbre *stain separation MC-dropout model* y *stain separation aleatoric model*, y la desarrollo del *framework* `her2bdl`, una herramienta para acelerar el desarrollo de nuevos modelos.

**Palabras Clave—** aprendizaje profundo bayesiano; incertidumbre; clasificación de HER2; aprendizaje supervisado

# ABSTRACT

***Abstract—*** *Human epidermal growth factor receptor 2* (HER2) is a biomarker used in histological samples to diagnose different types of cancer. *HER2 scoring*, is a crucial task performed by pathologists, allowing to determine if the patient is suitable for receiving HER2-targeted therapy. However, this analysis is semi-quantitative and prone to errors. Advances in the digitization of histopathology samples have allowed the development of computational tools to assist in the evaluation process. This is a non-trivial task due to the enormous size of the images and the various noise sources during their digitization. In this work, image analysis algorithms are implemented, based on the *bayesian deep learning* approach, to assist in the histopathological diagnosis of HER2 protein overexpression. This work's contributions are the uncertainty models *stain separation MC-dropout model* and *stain separation aleatoric model*, and the development of the `her2bdl` framework, a tool to accelerate the development of new models.

***Keywords—*** bayesian deep learning; uncertainty; HER2 classification; supervised learning

# GLOSARIO

ANN: Redes Neuronales Artificiales - *Artificial Neural Networks*.
API: *Application Programming Interface*.
BDL: *Bayesian Deep Learning*.
BP: Back-Propagation.
CNN: Redes Neuronales Convolucionales - *Convolutional Neural Networks*.
CSV: *Comma Separated Values*
CV: *Computer Vision*
DAB: Diaminobencidina.
DI: Departamento de Informática.
DL: *Deep learning*.
FN: Falso Negativo - *False Negative*.
FP: Falso Positivo - *False Positive*.
FWP: *Forward-Propagation*.
GT: *Ground Truth*.
HED: Hematoxilina-Eosin-DAB.
HER2: *Human epidermal growth factor receptor 2*.
HDX: Hematoxilina-DAB.
HPC: *High-Performance Computing*.
IHC: *Immunohistochemistry*.
MC: Monte-Carlo
mpp:micrómetros por píxel
ROI: *Region of Interest*.
TN: Verdadero Negativo - *True Negative*.
TP: Verdadero Positivo - *True Positive*.
UTFSM: Universidad Técnica Federico Santa María.
WSI: *Whole Slide Image*.
YAML: *YAML Ain't Markup Language*

# ÍNDICE DE CONTENIDOS

**RESUMEN** . . . . . IV

**ABSTRACT** . . . . . IV

**GLOSARIO** . . . . . V

**ÍNDICE DE FIGURAS** . . . . . VIII

**ÍNDICE DE TABLAS** . . . . . IX

**INTRODUCCIÓN** . . . . . 1

**CAPÍTULO 1: DEFINICIÓN DEL PROBLEMA** . . . . . 3
1.1 Contexto Médico . . . . . 3
1.2 Clasificación de la Sobreexpresión . . . . . 3
1.2.1 Análisis Computacional . . . . . 5
1.2.2 La Importancia de Medir la Incertidumbre . . . . . 6
1.3 Objetivos . . . . . 7
1.3.1 Objetivo General . . . . . 7
1.3.2 Objetivos Específicos . . . . . 7
1.4 Contribuciones . . . . . 7

**CAPÍTULO 2: MARCO CONCEPTUAL** . . . . . 9
2.1 Whole Slide Imaging . . . . . 9
2.1.1 Digital Slides . . . . . 9
2.2 Procesamiento de Imágenes Digitales . . . . . 10
2.2.1 OpenSlide . . . . . 11
2.3 Redes Neuronales Artificiales . . . . . 12
2.3.1 Representación . . . . . 12
2.3.2 Evaluación . . . . . 14
2.3.3 Optimizacón . . . . . 15
2.4 Deep Learning . . . . . 15
2.4.1 Redes Neuronales Convolucionales . . . . . 15
2.4.2 Clasificación de Imágenes . . . . . 17
2.5 Bayesian Deep Learning . . . . . 20
2.5.1 Cuantificar la Incertidumbre . . . . . 20
2.5.2 Modelo Probabilístico . . . . . 21
2.5.3 Estimación de la Incertidumbre Epistémica . . . . . 22
2.5.4 Medidas de Incertidumbre Epistémica . . . . . 23
2.5.5 Estimación de la Incertidumbre Aleatoria . . . . . 25
2.6 Estado del Arte . . . . . 26

**CAPÍTULO 3: PROPUESTA DE SOLUCIÓN** . . . . . 28
3.1 her2bdl: Framework para Clasificar Tejidos en Imágenes de Alta Resolución . . 28
3.2 Análisis de Imágenes de Alta Resolución . . . . . 30
3.2.1 Preprocesamiento Asistido de *Digital Slide* . . . . . 31
3.2.2 Generación de Conjunto de Datos . . . . . 33
3.2.3 Definición de Modelo de Clasificación . . . . . 35
3.2.4 Evaluación y Estimación de Incertidumbre . . . . . 35
3.3 Modelos de Incertidumbre . . . . . 35
3.3.1 Incertidumbre Epistémica: Monte-Carlo Classification Model . . . . . 36
3.3.2 *Predictive Distribution*: Estimación Eficiente . . . . . 39
3.3.3 Incertidumbre Aleatoria: Aletoric Classification Model . . . . . 40
3.4 Stain Separation Layer: Deconvolución de Color . . . . . 42
3.5 Métodos de Agregación . . . . . 44
3.5.1 Average Aggregator . . . . . 45
3.5.2 Threshold Aggregator . . . . . 45
3.5.3 Mixture Aggregator . . . . . 46
3.6 Conjunto de datos . . . . . 46
3.6.1 Cats vs. Dogs . . . . . 47
3.6.2 HER2 Scoring in Breast Cancer Histology Images . . . . . 48

**CAPÍTULO 4: VALIDACIÓN DE LA SOLUCIÓN** . . . . . 52
4.1 Clasificación Binaria . . . . . 52
4.1.1 Métricas de Clasificación . . . . . 53
4.1.2 Incertidumbre Epistémica . . . . . 54
4.1.3 Incertidumbre Aleatoria . . . . . 57
4.2 Evaluación de *Stain Separation Layer* . . . . . 58
4.3 Evaluación de Clasificación de *Patches* . . . . . 60
4.3.1 Métricas de Clasificación . . . . . 61
4.3.2 Incertidumbre Epistémica . . . . . 63
4.3.3 Incertidumbre Aleatoria . . . . . 65
4.4 Evaluación de la Sobreexpresión . . . . . 67
4.4.1 Criterios de Evaluación . . . . . 68
4.4.2 Visualización de Mapas de Incertidumbre . . . . . 69

**CAPÍTULO 5: CONCLUSIONES** . . . . . 71
5.1 Discusión . . . . . 72
5.2 Trabajo Futuro . . . . . 73

**ANEXOS** . . . . . 74

**REFERENCIAS BIBLIOGRÁFICAS** . . . . . 76

# ÍNDICE DE FIGURAS


1 Tejidos de dos pacientes tratados con IHC . . . . . 4

2 Estructura piramidal de una *digital slide* . . . . . 10

3 Imagen digital . . . . . 10

4 Red Neuronal Artificial con una capa oculta . . . . . 13

5 Operación de una capa de convolución . . . . . 16

6 Operación de una capa de *pooling* . . . . . 17

7 Arquitectura de una red neuronal convolucional. . . . . 17

8 Curva ROC . . . . . 19

9 Deep Learning vs. Bayesian Deep Learning . . . . . 20

10 Preprocesamiento asistido de *digital slide* . . . . . 32

11 Patch Generators . . . . . 34

12 Evaluación y estimación de incertidumbre . . . . . 36

13 Arquitectura de EfficientNet B0 MC-Dropout . . . . . 37

14 Arquitectura de EfficientNet B0 Aleatoric . . . . . 42

15 Descomposición de colores HED . . . . . 43

16 Stain separation layer . . . . . 44

17 Ejemplos de *Cats vs. Dogs* . . . . . 47

18 Distribución de clases de *Cats vs. Dogs* . . . . . 48

19 Comparación de tamaño de *digital slide* y *patch* . . . . . 48

20 Distribución de tamaños de *digital slides* en el conjunto de entrenamiento . . . 49

21 Distribución de clases en el conjunto de entrenamiento y evaluación . . . . . 50

22 Entrenamiento de modelo epistémico para *cats vs. dogs* . . . . . 54

23 Matriz de Confusión y curva ROC para *cats vs. dogs* . . . . . 54

24 Incertidumbre epistémica en *cats vs. dogs* . . . . . 55
25 Ejemplos con alta incertidumbre epistémica . . . . . 56
26 Incertidumbre aleatoria en *cats vs. dogs* . . . . . 57
27 Ejemplos con alta incertidumbre aleatoria . . . . . 58
28 Instancia de modelo stain separation MC-dropout . . . . . 59
29 Comparación de desempaño de *stain separation layer* . . . . . 60
30 Medidas de desempeño para *crossvalidation* . . . . . 60
31 Entrenamiento de modelos epistémicos para clasificación de *patches* . . . . . 62
32 Evaluación de modelos clasificadores de *patches* por clase . . . . . 63
33 Incertidumbre epistémica en clasificación de *patches* . . . . . 64
34 Predicciones destacadas de incertidumbre epistémica de *patches* . . . . . 65
35 Instancia de modelo stain separation aleatoric . . . . . 66
36 Distribución de incertidumbre aleatoria de *patch* por clase . . . . . 66
37 Ejemplos destacados de aleatoria en clasificación de *patch* . . . . . 67
38 Evaluación de método de agregación . . . . . 68
39 Predicciones agregadas destacadas . . . . . 70

# ÍNDICE DE TABLAS

1 Criterios de la ASCO/CAP para HER2 scoring . . . . . 5
2 Métricas de desempeño para la clasificación de la clase $c_k$. . . . . 18
3 Medidas para la clasificación multi-clase . . . . . 19
4 Ejemplo de medidas de incertidumbre epistémica . . . . . 24
5 Tiempo de ejecución para la estimación de *Predictive Distribution* . . . . . 40
6 Puntos de acuerdo para *score* predichos . . . . . 50

7 EfficientNet vs. Humanos . . . . . 57

8 Evaluación de modelos clasificadores de *patches* . . . . . 61

9 Evaluación de modelos clasificadores de *patches* por clase . . . . . 62

10 Medias de incertidumbre epistémica para modelos de clasificación de *patches* . 64

11 Configuraciones de métodos de agregación . . . . . 67

12 Resultados de evaluación a nivel de muestra . . . . . 69

13 Arquitecturas EfficientNetB0 . . . . . 74

14 Arquitecturas Stain Separation . . . . . 74

15 Arquitecturas Stain Separation Encoder Model . . . . . 74

16 Arquitecturas C876 . . . . . 74

17 Resultados de evaluación de conjunto de datos *test* . . . . . 75

# INTRODUCCIÓN

En la última década, el campo de la inteligencia artificial (AI) ha tenido un crecimiento exponencial. El gran volumen de datos generados cada segundo por nuestra sociedad cada vez más digitalizada; el poder de cómputo accesible cada vez mayor, a través del desarrollo en hardware de procesamiento dedicado como GPUs y TPUs; y la constante innovación de nuevos algoritmos, como el *deep learning* (DL), han permitido a la AI irrumpir, cada vez más, en distintos aspectos de nuestras vidas.

La proliferación del *deep learning* se debe a su capacidad de llevar a cabo tareas específicas y repetitivas, con una capacidad sobrehumana, siempre que se disponga de una gran cantidad de datos etiquetados. Es una tecnología con facultad de potenciar diversas industrias que cumplen con esta condición, por ejemplo: las redes sociales, agricultura, finanzas, y medicina.

La disponibilidad de datos y el acceso al conocimiento médico puede dar paso al desarrollo de herramienta analíticas potenciadas por la AI, brindando un apoyo a las decisiones tomadas por los médicos. En particular, en el análisis de imágenes médicas, un modelo de DL puede ser entrenado con miles de casos anteriores, encontrando rápidamente correlaciones indetectables por el ojo del experto. De esta forma, el proceso de diagnóstico médico se vuelve más eficiente y eficaz, haciendo un mejor uso del tiempo del médico.

Sin embargo, la mayoría de las soluciones de DL son tratadas como “cajas negras”, que dificulta analizar su razonamiento. Más aún, sus predicciones suelen ser tomadas con una confianza absoluta en ellas. Esto puede causar problemas cuando estas soluciones son desplegadas en escenarios reales, donde las decisiones tomadas por estos sistemas autónomos afectan directamente la vida de las personas, como lo es el diagnóstico médico. El enfoque *bayesian deep learning* (BDL) permite mitigar estas dificultades, facilitando la medición de la incertidumbre de los modelos, y de esta forma, permite cuantificar el grado de confianza de sus predicciones. Este enfoque puede ser utilizado en escenarios de alto riesgo, donde las decisiones tomadas por el modelo de DL, pone en riesgo a las personas.

Uno de estos escenarios es el diagnóstico del cáncer de mama. La proteína human epidermal growth factor receptor 2 (HER2) es un biomarcador utilizado en muestras histológicas para el diagnóstico de distintos tipos de cáncer. La clasificación de los tejidos con sobreexpresión de la proteína HER2, es una tarea crucial realizada por los patólogos, que permite determinar si el paciente es apto para recibir la terapia anti-HER2. Sin embargo, este análisis es semicuantitativo y propenso a errores. Utilizando las imágenes digitales de muestras de tejido, el desarrollo de modelos de BDL puede contribuir a la evaluación cuantitativa y objetiva de la sobreexpresión, y junto a una evaluación de incertidumbre, mitigar el riesgo de la evaluación.

En esta memoria, se abordará el problema de la clasificación de la sobreexpresión de la proteína HER2, también conocido como HER2 *scoring*. Desarrollando un algoritmo de análisis de imágenes, basado en el enfoque BDL para la estimación de incertidumbre, como una herra-

mienta que permita asistir a patólogos en el diagnóstico de la sobreexpresión de la proteína HER2. En este contexto, se enmarca *her2bdl*: *bayesian deep learning for HER2 tissue image classification*. Un *framework* desarrollado para investigadores, una herramienta que permita acelerar el desarrollo de nuevos modelos para el problema HER2 *scoring*.

El documento está estructurado de la siguiente manera, el Capítulo 1 contextualiza la importancia del diagnóstico de HER2 y se define el alcance de esta memoria. Para el desarrollo de algoritmos de análisis de imágenes de histopatología, basados en BDL, en el Capítulo 2 se introducen los distintos conceptos y herramientas utilizadas para su formulación. Seguido de una discusión bibliográfica sobre el problema de HER2 *scoring*. En el Capítulo 3, se realiza una descripción del *framework* `her2bdl`, seguido de la formalización de los modelos de incertidumbre implementados. El capítulo finaliza con una descripción de los datos a utilizar en la validación. En el Capítulo 4, se muestra los resultados en un problema de clasificación binaria, estos son utilizados para formular una interpretación de las medidas de incertidumbre. Luego, la solución propuesta es evaluada en el conjunto de datos de HER2. Finalmente, en Capítulo 5, se dan las conclusiones, seguidas de comentarios finales sobre las perspectivas para un trabajo futuro.

# CAPÍTULO 1
# DEFINICIÓN DEL PROBLEMA

En este capítulo se realizará una descripción de los hechos y elementos que configuran el problema respecto a la clasificación de la sobreexpresión de la proteína HER2. Dando a conocer el contexto médico en que se desenvuelve. A continuación se definirá la problemática a tratar y finalmente, se precisarán los objetivos de esta memoria.

## 1.1. Contexto Médico

La evaluación de la sobreexpresión de la proteína *human epidermal growth factor receptor 2* (HER2) presente en muestras histológicas de pacientes con cáncer, es un procedimiento clínico crucial realizado por los patólogos. Permite determinar el pronóstico de la enfermedad, junto con el tratamiento clínico [Gutierrez y Schiff, 2011].

La proteína HER2 se expresa[1] normalmente en la superficie de la membrana celular del epitelio de distintos órganos, como pulmón, vejiga, páncreas, mama, estómago y próstata. Cuando la proteína HER2 está presente en cantidades normales, esta juega un papel importante en el desarrollo normal del crecimiento celular. Sin embargo, cuando hay una producción excesiva de la proteína HER2, es decir, existe sobreexpresión, las células se multiplican y crecen más rápidamente de lo normal, promoviendo la carcinogénesis en diversos tipos de cáncer [Moasser, 2007].

En el caso del cáncer de mama, el análisis de la sobreexpresión de HER2 es fundamental para determinar cuando el paciente es apto para recibir la terapia anti-HER2, lo que incluye tratamientos con anticuerpos anti-HER2. Estudios indican que alrededor del 20 % de los pacientes con cáncer de mama son HER2 positivos, es decir, presentan sobreexpresión de la proteína HER2 [Gutierrez y Schiff, 2011, Sánchez *et al.*, 2018]. Un mal pronóstico, se asocia a una enfermedad más agresiva y un mayor riesgo de recurrencia de la enfermedad. El uso de la terapia anti-HER2 junto con quimioterapia aumentan la supervivencia general solo de las personas que padecen cáncer de mama HER2 positivo. [Wang y Xu, 2019]

## 1.2. Clasificación de la Sobreexpresión

La inmunohistoquímica (IHC, de sus siglas en inglés *immunohistochemistry*) es el procedimiento histológico comúnmente usado para analizar la sobreexpresión del biomarcador HER2.

[1] La expresión de una proteína se refiere a la cantidad de proteínas que elabora una célula.

En forma general, la IHC es una técnica que permite detectar e identificar anticuerpos específicos en células o muestras de tejido, basándose en el principio de unión existente entre antígenos y anticuerpos objetivos. En el caso de la evaluación de sobreexpresión de HER2, la proteína HER2 corresponde al antígeno objetivo [Pezoa, 2017]. La hematoxilina y la diaminobencidina (DAB) son compuestos químicos usados durante el procedimiento de IHC para visualizar los núcleos y las membranas celulares. Los efectos de aplicar estos compuestos a los tejidos mediante la técnica IHC puede apreciarse en la Figura 1. La hematoxilina se usa como contratinción que produce tinción azul-violeta de los elementos nucleares. DAB produce una tinción marrón oscuro en la membrana celular sólo cuando hay sobreexpresión de HER2.

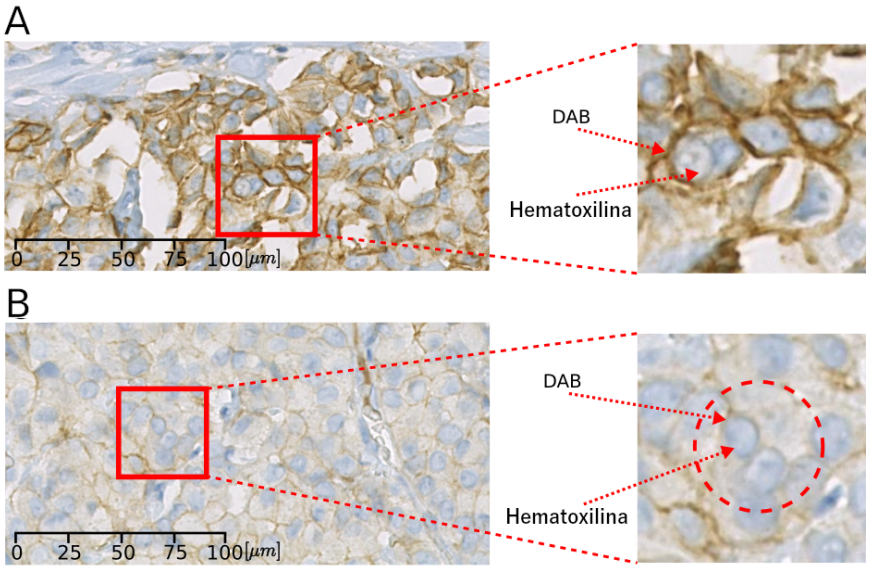



Figura 1: Tejidos de dos pacientes tratados con IHC - La hematoxilina resalta los núcleos celulares con una tinción azul-violeta. DAB resalta membrana celular con una tinción marrón oscuro, cuando estas presentan sobreexpresión de HER2. El paciente (A) presenta un tejido con mayor sobreexpresión que el paciente (B), esto se aprecia en la intensidad del color marrón de las membranas celulares. En (A) se puede identificar fácilmente la cada célula y su membrana, mientras que en (B) esta tarea requiere un mayor esfuerzo.
Fuente: [Pezoa, 2017].

A la muestra se le asigna una clasificación [Wolff *et al.*, 2018], también definido como *HER2 scoring*, correspondiente a una de las siguientes cuatro categorías (ver Tabla 1): **0**, **1+ (negativos HER2)**, **2+ (equívocos)** o **3+ (positivos HER2)**. Solo los pacientes con HER2 positivo pueden recibir el tratamiento anti-HER2.

La clasificación de muestras de histopatología con sobreexpresión de la proteína HER2, es un procedimiento clínico que debe ser realizado por un patólogo con experiencia, lo que requiere de tiempo del especialista, asociado a costo económico, que claramente aumenta a medida que crece el volumen de muestras que deben analizarse. Junto a esto, la evaluación realizada es un análisis semi-cuantitativo, ya que corresponde al resultado

| Score | Características de la tinción | Evaluación |
|---|---|---|
| 0 | No se observó tinción de la membrana o se identifica en menos del 10 % de las células tumorales. | Negativo |
| 1+ | Tinción de membrana débil y tenue en alguna proporción de las células tumorales. | Negativo |
| 2+ | Tinción de membrana débil y no uniforme en más del 10 % de las células tumorales. | Equivoco (Débilmente positivo) |
| 3+ | Se observa una tinción de membrana vigorosa e intensa en más del 10 % de las células tumorales. | Positivo |

Tabla 1: Criterios de la Asco/Cap para HER2 scoring.
Fuente: [Wolff *et al.*, 2018]

de la inspección visual —basado en los conocimientos y experiencia médica— que puede estar sesgado por la experiencia del patólogo, por el kit de IHC utilizado, por las características del equipo de digitalización utilizado, entre otros. Esto conlleva a que distintos profesionales reporten distintos resultados para el mismo paciente. Como se reporta en [Qaiser *et al.*, 2018, Perez *et al.*, 2006, Rakha *et al.*, 2015, Cregger *et al.*, 2006], existiendo un alto grado de discordancia, de un $20$ %, entre diferentes laboratorios.

Por lo tanto, introducir herramientas de asistencia, que permitan cuantificar los parámetros analizados por los patólogos, es un apoyo significativo en la generación de resultados cuantitativos y objetivos, generando un impacto en la reducción de la discordancia, y por ende, aumentando la reproducibilidad de los resultados.

### 1.2.1. Análisis Computacional

Gracias al avance en los escáneres digitales de alta resolución, es posible digitalizar las muestras obteniendo una *digital slide* (DS), imágenes de gran aumento y alta resolución, han permitido el desarrollo de *software* para el análisis computacional de estos tejidos. Sin embargo, desarrollar algoritmos para analizar estas imágenes es una tarea desafiante.

Existen distintos factores que generan variaciones, o ruido, en las distintas imágenes obtenidas. El proceso de tinción depende de la selección del anticuerpo y tiempo de fijación, estos provocan variaciones de color de las estructuras de interés o tinción no deseada. De igual forma, es desafiante la morfología irregular de la célula, la discontinuidad y las estructuras finas de las membranas, y la existencia de artefactos en la imagen.

Es necesario tener algoritmos que permitan automatizar el análisis y asistir la labor de los patólogos. La construcción de estos algoritmos deben ser robustos, considerando las distintas fuentes de ruido y heterogeneidad de las imágenes, para así obtener resultados cuantitativos

replicables.

### 1.2.2. La Importancia de Medir la Incertidumbre

En los últimos años, algoritmos de *machine learning* (ML), y más específicamente del *deep learning* (DL), han demostrado su gran potencial de resolver problemas complejos en diversos escenarios, donde los algoritmos clásicos anteriores han fallado los algoritmos de DL han logrado resultados satisfactorios, incluso superando a los humanos. Como se demostró en el año 2017 con *CheXNet* [Rajpurkar *et al.*, 2017], los autores desarrollaron un modelo de DL entrenado con en un conjunto de datos de imágenes de rayos X de tórax, el algoritmo logró superar el desempeño promedio del radiólogo en la detección de 14 enfermedades respiratorias.

Sin embargo, al hablar de DL frecuentemente se suele comparar con una "caja negra". El concepto de "caja negra" se refiere a la incapacidad de ver el funcionamiento interno de un sistema, donde solo es posible observar lo que entra y sale de este. Esta metáfora de los modelos de DL destaca la baja interpretabilidad que sufren en cuanto a su representaciones internas, es decir, al momento de obtener una respuesta particular del algoritmo, no se sabe realmente cuáles son las razones o reglas que utiliza para llegar a tal conclusión.

Por otro lado, se encuentra el problema de la confianza que se le suele dar a los resultados, permite que se den situaciones incómodas para los investigadores como la ocurrida en 2015 [BBC, 2015], cuando el algoritmo de reconocimiento de imágenes de *Google*, erróneamente identifica como gorilas a una foto de una pareja afroamericana. Si bien nadie resultó gravemente afectado en la situación anterior, el asunto toma una mayor importancia cuando estos algoritmos son implementados en otros tipos de escenarios, donde las decisiones tomadas son potencialmente perjudiciales para el bienestar de la ciudadanía. Escenarios donde existe un alto nivel de riesgo asociado a la toma de decisión como la conducción autónoma, el mercado de acciones o el tema que trata esta memoria, el diagnóstico médico.

Es posible mitigar este riesgo que surge de desplegar modelos de DL en el mundo real. Una medición de incertidumbre permite cuantificar el grado de confianza que se podría tener en el modelo, identificando cuando este no es capaz de determinar una respuesta certera, delegando la responsabilidad al humano para tomar la decisión.

Determinar cuando el modelo "no sabe" la respuesta para el diagnóstico de un paciente puede ser vital para evitar casos de diagnósticos erróneos por parte del algoritmo. Al tener disponible tanto la respuesta junto a la incertidumbre, el patólogo puede utilizar esta información extra para tomar la decisión final de su análisis. Es así como el desarrollo de técnicas que permitan medir el nivel de incertidumbre, es decir, el grado de confianza que tiene el algoritmo sobre su respuesta puede tener un gran impacto en el desarrollo e implementación de soluciones de DL en escenarios reales.

## 1.3. Objetivos

Esta memoria aborda el problema de la clasificación de imágenes y su aplicación práctica en escenarios médicos. Desarrollando una herramienta para la asistencia del diagnóstico histopatológico de la sobreexpresión de la proteína HER2.

La solución propuesta, junto a la clasificación de las imágenes de tejido, utiliza la incertidumbre de la clasificación como un elemento relevante para el diagnóstico. Al determinar el grado de confianza de la respuesta, el patólogo tendrá un factor a evaluar para el diagnóstico del tejido.

A continuación, se presentan el objetivo general y objetivos específicos de esta memoria.

### 1.3.1. Objetivo General

Construir un algoritmo de clasificación de la sobreexpresión de la proteína HER2 en imágenes de tejidos, basado en el enfoque *bayesian deep learning*, para la asistencia en el diagnóstico histopatológico.

### 1.3.2. Objetivos Específicos

- Diseñar un *framework* para clasificar tejidos en imágenes de alta resolución (WSI).
- Implementar un modelo de clasificación de imágenes, basado en *bayesian deep learning*.
- Medir la incertidumbre asociada a los datos y al modelo de clasificación.

## 1.4. Contribuciones

Este trabajo explora esta brecha que existe respecto a la incertidumbre de los modelos de clasificación, en el campo del diagnóstico histopatológico. Aportando el *framework* `her2bdl` [2] para investigadoras e investigadores, con el objetivo de facilitar el desarrollo, entrenamiento y evaluación de nuevos modelos de clasificación, con el enfoque en la estimación de la incertidumbre. Además, `her2bdl` integra herramientas para facilitar la colaboración, replicación los experimentos y divulgación de resultados.

[2] her2bdl: Bayesian Deep Learning for HER2 tissue image classification: https://github.com/sborquez/her2bdl

El *framework* es utilizado en el problema del diagnóstico de las sobreexpresión de la proteína HER2. Implementando un modelos especializados en la clasificación de DS de muestras tratadas con IHC: *stain separation MC-dropout model* y *stain separation aleatoric model*. Entregando una herramienta para de asistencia para el diagnóstico, que permita a patólogas y patólogos obtener resultados cuantitativos replicables.

Los resultados de esta memoria pueden ser aplicados a diferentes campos y escenarios dentro del ámbito médico, como por ejemplo, la identificación de signos de retinopatía diabética en imágenes oculares. Sin embargo, su aplicabilidad no se limita a la medicina, los resultados pueden extenderse y aplicarse a una variedad de problemas de clasificación, donde exista riesgo asociado a la clasificación tomada por el algoritmo.

# CAPÍTULO 2
# MARCO CONCEPTUAL

En este capítulo se introduce al lector los conceptos claves para digitalización de tejido y la solución propuesta. Se definen términos como *whole slide imaging*, la incertidumbre, redes neuronales artificiales, *deep learning* y el *bayesian deep learning*. Finalmente, este capítulo cierra con una discusión bibliográfica, una análisis de las soluciones propuestas por estudios previos.

## 2.1. Whole Slide Imaging

El término *whole slide imaging* (WSI) [Farahani *et al.*, 2015, DICOM, 2020], también definido como "microscopio virtual", corresponde al procedimiento de escanear una muestra contenida en el portaobjetos de vidrio de un microscopio, como por ejemplo, tejido mamario tratado con IHC, obteniendo como resultado una imagen digital o *digital slide*.

### 2.1.1. Digital Slides

La *digital slide* (DS) obtenida es una imagen de alta resolución de la muestra. Esta tecnología permite a los patólogos poder almacenar y compartir imágenes de tejidos, así como abrir la posibilidad de desarrollar herramientas computacionales para el análisis de estas imágenes digitales.

Las imágenes tienen un gran tamaño y poseen alto nivel de detalle. Por lo general, una muestra de 20mm largo y 12 mm de ancho, digitalizada con un aumento de 40X, resulta en una imagen con una resolución de .25$\mu$m/píxel (micrómetros por píxel, o mpp) y un tamaño de alrededor de 80.000 $\times$ 60.000 píxeles. Por convención [DICOM, 2020], una resolución de .25mpp es denominada como 40X, imágenes con una resolución de .5mpp, son denominadas como 20X, 2.5mpp como 4X y 4mpp como 2X.

El formato de una DS permite almacenar múltiples resoluciones en un documento. Estas imágenes, con diferentes niveles de aumento, son organizadas para el patólogo en una estructura piramidal multi-capa (ver Figura 2). Esta característica resulta conveniente para los patólogos, permitiéndoles utilizar distintos resoluciones durante el análisis.

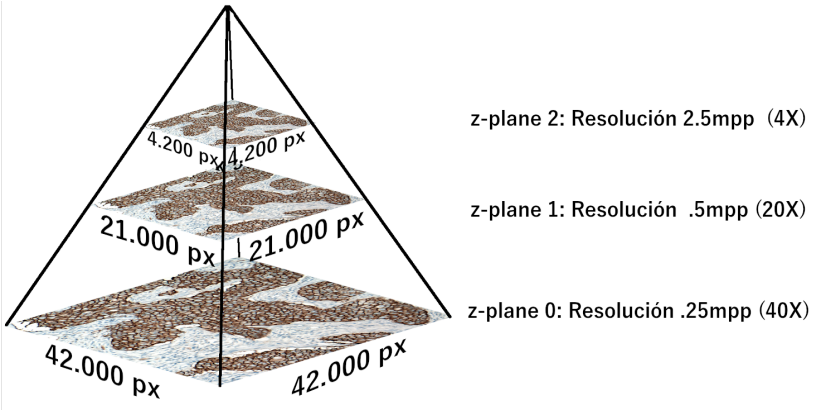



Figura 2: Estructura piramidal de una *digital slide* - Esta consiste de múltiples imágenes con diferentes resoluciones (la altura de la pirámide corresponde al nivel de aumento). En la base de la pirámide se encuentra la imagen con mayor resolución. Una o más niveles intermedios pueden crearse, con resoluciones intermedias, para facilitar la recuperación de la imagen a resoluciones arbitrarias.

Fuente: Elaboración propia.

## 2.2. Procesamiento de Imágenes Digitales

El procesamiento de imágenes digitales [Gonzalez y Woods, 2002] radica en el procesamiento de imágenes digitales por medio de una computadora digital. Esta disciplina proporciona las diversas herramientas y algoritmos utilizados en el análisis automático de las *digital slides*.

Se define a una imagen digital $I_{(w,h)}(i,j)$, como una función de dos dimensiones $i \times j$, definida para un dominio discreto $[0,w) \times [0,h)$. Una imagen digital se compone de un número finito de elementos, cada uno de los cuales tiene una ubicación $(i,j)$ y uno o más valores de intensidad descrito por $I$. Estos elementos se denominan *picture element*, comúnmente llamado *píxel*.

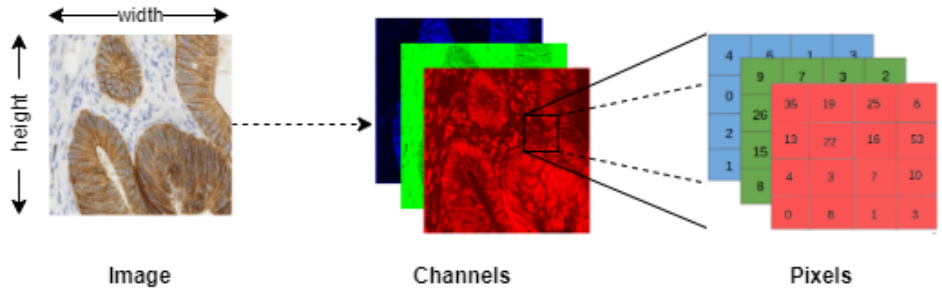



Figura 3: Imagen digital.
Fuente: Elaboración propia.

La cantidad de valores de intensidad de cada píxel, denominados canales, está dado por el espacio de color de la imagen digital. Los valores del píxel, junto al espacio de colores

determinan el color representado en el píxel. El espacio de colores comúnmente usado por las imágenes digitales es el RGB (*red*, *green*, *blue)*, donde cada canal representa la intensidad de rojo, verde y azul de un píxel.

La mayor parte de los algoritmos de procesamiento de imágenes, se construyen a partir de dos operaciones fundamentales:

- **Convolución**: Esta operación realiza una suma de productos entre una imagen $I$ y un filtro o *kernel*. El *kernel* es una matriz cuyo tamaño define la vecindad de operación y cuyos coeficientes determinan la naturaleza del filtro. La aplicación se realiza con una ventana deslizante, aplicando un producto entre el filtro y el vecindario de la imagen. La Figura 5 muestra un ejemplo de esta operación.
- **Operaciones morfológicas**: conjunto de operaciones que procesan imágenes binarias, por ejemplo máscaras de segmentación, basadas en un elemento estructurante. Las operaciones morfológicas básicas son la **dilatación**, la cual agrega píxeles a los límites de los objetos de una imagen, mientras que la **erosión**, se encarga de eliminar los píxeles en los límites de los objetos.

  La combinación de estas dos operaciones resulta en nuevas operaciones. La **apertura** u *opening* (erosión+dilatación) para eliminar objetos pequeños de una imagen conservando la forma y el tamaño de los objetos más grande. La operación de **cierre** o *closing* (dilatación+erosión), se utiliza para rellenar pequeños agujeros de una imagen conservando la forma y el tamaño de los objetos de la imagen.

### 2.2.1. OpenSlide

Al leer una imagen digital desde el disco, las herramientas para el procesamiento digital de imágenes suelen cargar completamente la imagen en la RAM. Al no estar diseñadas para la imágenes de alta resolución, como lo es una DS, el gran tamaño de estas puede causar problemas durante la lectura.

Las *digital slides* requieren de un espacio en memoria mucho mayor a la RAM disponible en los sistemas, requiriendo varios gigabytes luego de su descompresión. En consecuencia, es necesario de herramientas especializadas para el procesamiento de *digital slides*.

*OpenSlide* [Goode *et al.*, 2013] es una librería en *C* que provee una interfaz simple para leer de *digital slides*. *OpenSlide* permite el desarrollo de herramientas para el análisis y visualización de las *digital slides* en varios lenguajes. La librería puede ser utilizada en aplicaciones escritas en *C*, *C++*, *Java* o *Python*.

La librería soporta la estructura piramidal de una DS, proporcionando una interfaz para el análisis de estas imágenes y desarrollo de algoritmos a multinivel.

## 2.3. Redes Neuronales Artificiales

Las redes neuronal artificiales, o en inglés *artificial neural netoworks* (ANN), son una familia de modelos computacionales inspirado en el funcionamiento de cerebro animal. El objetivo de estos algoritmos es el de modelar comportamientos inteligentes como la reconocimiento de imágenes o el procesamiento del lenguaje natural.

Las ANN son un subconjuntos de algoritmos pertenecientes al campo del aprendizaje automático o *machine learning* (ML). Esto implica que, el desempeño de estos algoritmos en completar una tarea asignada depende de un conjunto de datos, es decir, utilizan una serie de ejemplos, y de esta forma, aprender como realizar la tarea asignada.

### 2.3.1. Representación

La ANN [Goodfellow *et al.*, 2016, Bishop, 2006] es un modelo conformado por un conjunto de neuronas interconectadas, agrupadas en capas. La complejidad del modelo, o bien, el espectro de tareas que puede realizar una ANN viene dado, en gran parte, por su arquitectura. Diseñar una nueva arquitectura de una ANN puede ser reducirse a la definición de los siguientes tres componentes:

- Número de capas de la red.
- Tipo de conexiones entre capas.
- Tipo de neuronas o funciones de activación.

El primer componente influye directamente en el tamaño de la red y la cantidad de parámetros de esta. El tipo de conexiones puede influenciar también en la cantidad de parámetros de la red. Sin embargo, el interés de modificar este componente radica en modificar la capacidad de representar información internamente, como se verá en las secciones 2.4.1 y 2.5. Finalmente, modificar el tipo de neurona no altera la cantidad de parámetros, pero permite abordar problemas no-lineales, o alterar al procedimiento de optimización.

El diagrama de la Figura 4 representa la arquitectura de una ANN simple, está conformada por tres capas: a la izquierda se encuentra la capa de entrada; a la derecha, se encuentra la capa de salida; en el medio se encuentran las capas ocultas, en este caso, solo posee una sola capa oculta. El diagrama presenta las dependencia (tipo de conexiones) entre neuronas y cuál es el flujo de la información, desde la entrada a la salida.

El espacio de hipótesis de una ANN, es decir, los diferentes modelos posibles funciones que puede aprender la ANN a partir del conjunto de datos, se encuentra definido por las distintas conexiones posibles entre neuronas de la red. Cuando la ANN está aprendiendo de los datos,

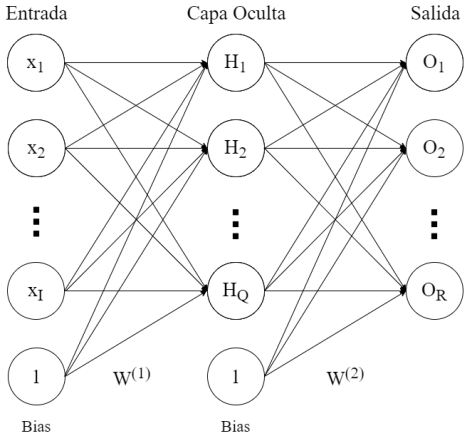



Figura 4: Red Neuronal Artificial con una capa oculta.
Fuente: Elaboración propia.

esta ajusta los denominados pesos (*weights*) de las conexiones. Estos pesos indican cual es el aporte, representado como un valor real, de las neuronas de la capa anterior a la entrada de una neurona en particular.

Una ANN puede representarse, como se describe en [Higham y Higham, 2019], como una función $F$, parametrizada por los pesos $W$. Esta transforma un conjunto de entrada, representada por un vector de características $\vec{x}$, a un conjunto de salida, representados por el vector objetivo $\vec{y}$. Esta transformación es solo una aproximación al valor real, por lo que se defina a la salida como el vector $\hat{\vec{y}}$. Con el fin de simplificar la notación, se omite el $\vec{\cdot}$, en los vectores.

$$F : \mathbb{R}^I \longrightarrow \mathbb{R}^R$$
$$F(x, W) \mapsto \hat{y}$$

Utilizando la Figura 4 como una instancia simple de una ANN, es posible construir una notación base para la definición de $F$. Esta notación puede ser extendida para arquitecturas más complejas discutidas en esta memoria.

Con el objetivo de describir a la ANN utilizando operaciones matriciales, se incluye el *bias* $x_0$, como un elemento más de $x$, tal que $x \in \mathbb{R}^{(I+1)} : x = (x_0, x_1, x_2, ..., x_I)^\intercal$. De la misma forma, se define el vector $H \in \mathbb{R}^{(Q+1)}$, el cual representa a las neuronas contenidas en la capa oculta más el *bias*, como $H = (H_0, H_1, H_2, ..., H_Q)^\intercal$. Se define también el vector $O \in \mathbb{R}^R$, el cual contiene los valores correspondiente a la capa de salida, $O = (O_1, O_2, ..., O_R)^\intercal$.

Además, se definen las matrices de pesos: $W^{(1)} \in \mathbb{R}^{Q\times(I+1)}$, compuesta por $W^{(1)} = \{w^{(1)}_{qi}\}$, con $i \in [0, ..., I]$ y $q \in [1, ..., Q]$; y la matriz $W^{(2)} \in \mathbb{R}^{R\times(Q+1)}$, compuesta por $W^{(2)} = \{w^{(2)}_{qi}\}$, con $q \in [0, ..., Q]$ y $r \in [1, ..., R]$. Estas son los pesos de la primera y segunda capa,

respectivamente.

Cada valor $H_q$ se obtiene de la combinación lineal de los valores de la capa anterior, en este caso corresponden al vector de entrada:

$$H_q = \sigma^{(1)}(\sum_{i=0}^{I} w_{qi}^{(1)} x_i), \quad q = 1, ..., Q$$
$$H = \sigma^{(1)}(W^{(1)} x)$$

Donde $\sigma^{(1)}(\phi)$ es una función escalar no lineal, correspondiente a la función de activación de la capa oculta 1.

De igual forma, se puede obtener los valores de la capa de salida $O$.

$$O_r = \sigma^{(2)}(\sum_{q=0}^{I} w_{rq}^{(2)} h_q), \quad r = 1, ..., R$$
$$O = \sigma^{(2)}(W^{(2)} H)$$
$$\hat{y} = O$$

Donde $\sigma^{(2)}(\phi)$ es una función escalar no lineal, correspondiente la función de activación de la capa de salida.

El procedimiento descrito para obtener el vector de aproximación $\hat{y}$ se denomina como *forward-propagation*.

### 2.3.2. Evaluación

Determinar el desempeño de la ANN permite distinguir cuales parámetros realizan una mejor aproximación.

La función de pérdida $\mathcal{L}$ permite evaluar la aproximación de la ANN para los pares $(\vec{x}, \vec{y}) \in \mathcal{D}$. Un ejemplo de función de pérdida es el error cuadrado:

$$\mathcal{L}(F(\vec{x}, W), \vec{y}) = ||F(\vec{x}, W) - \vec{y})||_2^2$$

La definición de $\mathcal{L}$ depende del problema, así como de la representación utilizada para los datos.

La función de costo $J$, es una función de $W$ parametrizada en un conjunto de datos de entrenamiento $\mathcal{D}$, compuesto por pares de valores $(\vec{x}_i, \vec{y}_i)$. Esta permite evaluar el desempeño de la ANN para un conjunto de datos utilizando una función de pérdida $\mathcal{L}$.

$$J(W) = \frac{1}{|\mathcal{D}|} \sum_{(\vec{x}_i, \vec{y}_i) \in \mathcal{D}} \mathcal{L}(F(\vec{x}_i, W), \vec{y}_i)$$

### 2.3.3. Optimizacón

Encontrar el mejor conjunto de pesos $W$ se convierte en un problema de optimización, donde el objetivo es encontrar el $W$ que minimice el riesgo empírico $R_{\text{emp}}$ para un conjunto de datos de entrenamiento $\mathcal{D}_{\text{train}}$:

$$\begin{aligned} \min_W R_{\text{emp}}(W) &= \min_W J(W) \\ &= \min_W \frac{1}{|\mathcal{D}_{\text{train}}|} \sum_{(\vec{x}_i, \vec{y}_i) \in \mathcal{D}_{\text{train}}} \mathcal{L}(F(\vec{x}_i, W), \vec{y}_i) \end{aligned}$$

El algoritmo de *back-propagation* [Goodfellow *et al.*, 2016] permite propagar la información obtenida de la función de costo en la capa de salida, devolviéndose hasta la capa de entrada. El método de *back-propagation* permite obtener, para cada capa, el gradiente en función de los pesos, de esta forma, poder ajustar los pesos de cada capa métodos de optimización basados en gradientes [Sun *et al.*, 2019] como *stochastic gradient descent*, *RMSProp*, *ADAM*, entre otros.

## 2.4. Deep Learning

Los diversos estudios y aplicaciones del *deep learning* (DL) ha demostrado que generar arquitecturas de ANN con cada vez más capas y más neuronas en cada capa, ha permitido aproximar funciones cada vez más complejas. Siempre que se dispongan de la cantidad de datos suficientes, generar modelos de mayor tamaño puede mejorar el desempeño de modelo, en un gran espectro de tarea.

### 2.4.1. Redes Neuronales Convolucionales

Las redes neuronales convolucionales [LeCun *et al.*, 2010], o *convolutional neural networks* (CNN), son un tipo de ANN que se especializan en el procesamiento de datos presentados en

estructura de grilla. Ejemplos de estos tipos de datos son las imágenes o series de tiempo, datos que presentan una correlación espacial o temporal, respectivamente.

Estos tipos de redes utilizan dos tipos de capas, las cuales le otorgan la característica de ser invariante al desplazamiento, estas son:

- **Capas convolucionales (Convolutional layers)**: Son el bloque de construcción fundamental de las CNN. Consiste en un *kernel* o filtro con parámetros ajustables. Durante el *forward-propagation*, se obtiene un mapa de características como el resultado de realizar una convolución discreta entre el filtro y el tensor de entrada, como ilustra la Figura 5. La propiedad de invariabilidad al desplazamiento, se debe a que la capa comparte los mismos pesos, los parámetros del filtro, en diferentes conexiones entre neuronas. Las capas convolucionales se definen por: el tamaño del filtro (*kernel size*), o espacio de percepción; la cantidad de canales de salida; el paso (*stride*) para aplicar la ventana deslizante; y el como se tratan los casos de borde (*padding*).

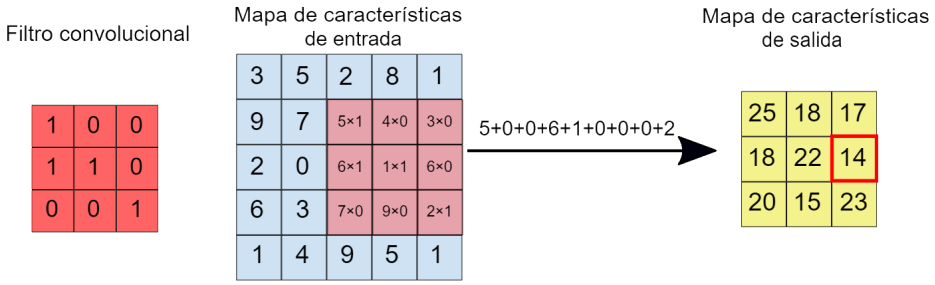



Figura 5: Operación de una capa de convolución - A la izquierda, una convolución de 3×3. En el centro, se realiza la convolución 3×3 en el mapa de atributos de entrada de 5×5. A la derecha, la imagen resultante.
Fuente: developers.google.com

- **Capas de agrupamiento (Pooling layers)**: Utilizadas para reducir el tamaño de entrada al extraer las características relevantes. Esto le otorga la capacidad de invariante al desplazamiento y a la rotación. Existen dos tipos de capas de agrupamientos: *average pooling*, el cual extraer el promedio de los valores de un vecindario; *max pooling*, el cual extrae el máximo valor. Al igual que las capas convolucionales, se encuentran parametrizadas por el tamaño del filtro (*pool size*), el paso (*stride*), y como se se tratan los casos de borde (*padding*). La Figura 6, ilustra el uso de una capa *max pooling*.

Las arquitecturas CNN suelen presentar una estructura similar a la que se ve en la Figura 7, utilizando una serie de capas convolucionales, seguidas de capas de *pooling*. Esto permite extraer características espaciales de la entrada.

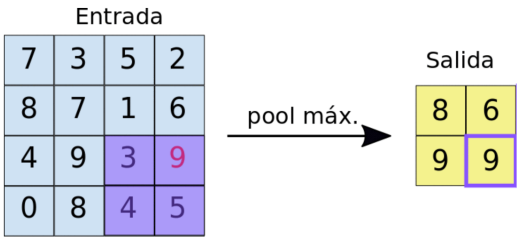



Figura 6: Operación de una capa de *pooling* - A la izquierda: La reducción máxima realizada en un mapa de atributos de 4×4 con un filtro de 2×2 y un segmento de 2. A la derecha: El resultado de la operación de reducción máxima. Debes tener en cuenta que el mapa de atributos resultante ahora es de 2×2, lo que preserva solo los valores máximos de cada mosaico.
Fuente: developers.google.com

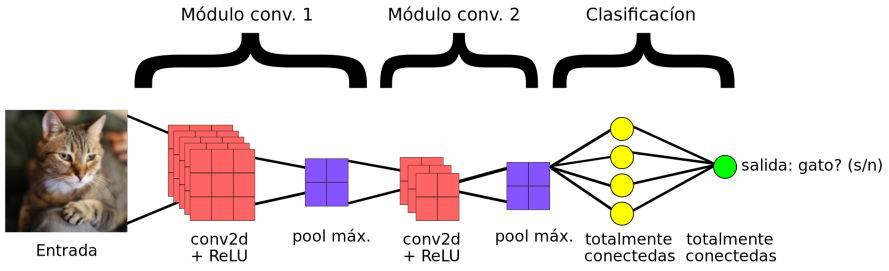



Figura 7: Arquitectura de una red neuronal convolucional.
Fuente: developers.google.com

### 2.4.2. Clasificación de Imágenes

La clasificación de imágenes es una tarea fundamental del campo de la visión por computadora (CV). Se define como la tarea de asignar a una imagen $x_i$ una, y solo una, etiqueta o clase $c_k$ de un conjunto definido $C$, con $K$ clases. El caso particular, cuando la cantidad de clases es dos ($K = 2$), se denominada **clasificación binaria**. En otro caso ($K > 2$), se define como **clasificación multi-clase**.

Lograr esta tarea requiere de la extracción de información, o características relevante de la imagen, que permitan determinar la clase a la que pertenece. La clasificación de imágenes puede abordarse, desde el campo del ML, como una tarea de aprendizaje supervisado, donde el clasificador aprende a determinar la clase de una imagen a partir del una serie de ejemplos, conjunto de imágenes previamente clasificadas por un ser humano.

Es posible definir una ANN $F$ que cumpla con la tarea de clasificación de imágenes:

$$F : I_{(w \times h)} \longrightarrow C$$
$$F(x) \mapsto \hat{c}_k$$

Donde $x$ es una imagen digital $I$, con un ancho $w$ y alto $h$, representada como un tensor de 3 dimensiones. Y $C$ es el conjunto discreto de posibles $K$ clases.

A continuación, se definen los componentes básicos de las métricas que se utilizan para evaluar el desempeño de los modelos de clasificación [Sokolova y Lapalme, 2009]. Para cada clase $c_k$, utilizando los ejemplos clasificados se calculan las siguientes cantidades:

- **Verdadero Positivo (*True Positive* (TP))**: Cantidad de ejemplos correctamente clasificados en la clase $c_k$.
- **Verdadero Negativo (*True Negative* (TN))**: Cantidad de ejemplos correctamente clasificados en una clase distinta a $c_k$.
- **Falso Positivo (*False Positive* (FP))**: Error de tipo 1. Cantidad de ejemplos incorrectamente clasificados en la clase $c_k$.
- **Falso Negativo (*False Negative* (FN))**: Error de tipo 2. Cantidad de ejemplos incorrectamente clasificados en una clase distinta a $c_k$.

A partir de estas cantidades se construye la matriz de confusión, de donde se puede obtener las métricas de clasificación. La Tabla 2 muestra las métricas por cada una de las clases $c_k$. La Tabla 3 muestra las métricas promediadas para las $K$ clases en $C$.

| Métrica | Fórmula | Descripción |
|---|---|---|
| Precisión (*precision*) | $\frac{TP_k}{TP_k+FP_k}$ | PPV - Concordancia de la clase verdadera con la clase asignada por el clasificador. |
| Sensibilidad (*recall*) | $\frac{TP_k}{TP_k+FN_k}$ | TPR - Efectividad del clasificador en identificar la clase $c_k$. |
| Exactitud (*accuracy*) | $\frac{TP_k+TN_k}{TP_k+FN_k+FP_k+TN_k}$ | ACC - Efectividad general de un clasificador para la clase $c_k$. |

Tabla 2: Métricas de desempeño para la clasificación de la clase $c_k$ - $TP_k$ son los TP de la clase $c_k$, $TN_k$ son los TN de la clase $c_k$, $FP_k$ son los FP de la clase $c_k$ y $FN_k$ son los FN de la clase $c_k$.

Fuente: [Sokolova y Lapalme, 2009]

Junto a estas métricas, está también la **curva ROC** [Fawcett, 2006] (ver Figura 8). Una curva *receiver operating characteristic*, o curva ROC, es un diagrama gráfico que ilustra la capacidad de diagnóstico de un clasificador binario a medida que varía su umbral de discriminación. La curva ROC se crea trazando el *recall* o la tasa de verdaderos positivos (TPR) frente a la tasa de falsos positivos (FPR) en varios valores de umbral.

| Métrica | Fórmula | Descripción |
|---|---|---|
| Precisión$_M$ (*precision*$_M$) | $\frac{\sum_{k=1}^{K} \frac{TP_k}{TP_k+FP_k}}{K}$ | PPV$_M$ - Promedio por clase de la concordancia de la clase verdadera con la clase asignada por el clasificador. |
| Sensibilidad$_M$ (*recall*$_M$) | $\frac{\sum_{k=1}^{K} \frac{TP_k}{TP_k+FN_k}}{K}$ | TPR$_M$ - Efectividad por clase promedio en identificar cada clase. |
| Exactitud Promedio (*Average Accuracy*) | $\frac{\sum_{k=1}^{K} \frac{TP_k+TN_k}{TP_k+FN_k+FP_k+TN_k)}}{K}$ | ACC$_M$ - Efectividad por clases promedio de un clasificador. |
| Tasa de Error (*Error Rate*) | $\frac{\sum_{k=1}^{K} \frac{FP_k+FN_k}{TP_k+FN_k+FP_k+TN_k)}}{K}$ | ERR$_M$ - Error por clase promedio del clasificador. |

Tabla 3: Medidas para la clasificación multi-clase de las $K$ clases en $C$. $TP_k$ son los TP de la clase $c_k$, $TN_k$ son los TN de la clase $c_k$, $FP_k$ son los FP de la clase $c_k$ y $FN_k$ son los FN de la clase $c_k$. $M$ indican que se utiliza un macro-promedio respectivamente. Una métrica calculada por un macro-promedio, calculará independientemente la métrica por cada clase y promediará los resultados, cada clase es tratada con la misma relevancia. En cambio, un micro-promedio, el cual calcula cada métrica pesando la contribución de cada clase al promedio.

Fuente: [Sokolova y Lapalme, 2009].

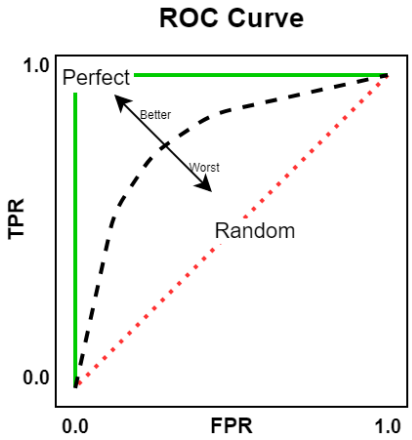



Figura 8: Curva ROC - La evaluación de un modelo se representa por una curva en el gráfico. Un modelo es mejor frente a otro si su curva se acerca más a la curva perfecta (verde).
Fuente: Elaboración propia.

## 2.5. Bayesian Deep Learning

El enfoque *bayesian deep learning* (BDL) [Gal, 2016] es una interpretación probabilista de los modelos de DL. Permite extender las ideas y herramientas desarrolladas en el campo del DL para la estimación de la incertidumbre. Esto se logra representando la salida de la ANN como una distribución de probabilidad, o como se describirá en esta sección, con una nueva formulación de las ANN, donde los pesos $W$ de la red dejan de ser descritos como escalares, en cambio, los pesos del modelo bayesiano $w$, pasan a ser descritos por distribuciones de probabilidad $w \sim \Omega$.

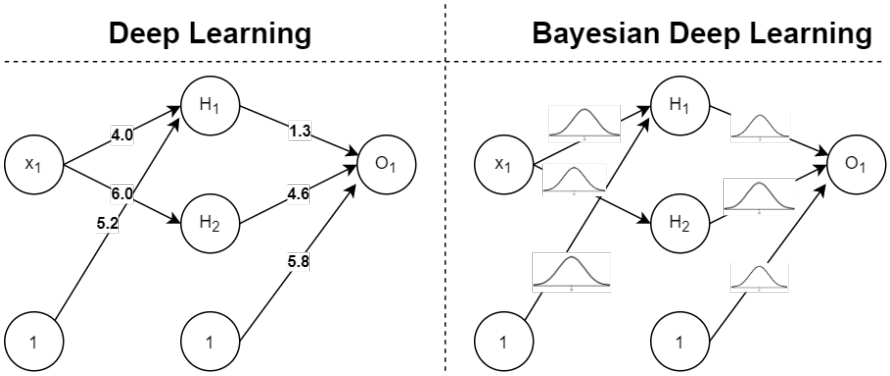



Figura 9: Deep Learning vs. Bayesian Deep Learning.
Fuente: Elaboración propia.

### 2.5.1. Cuantificar la Incertidumbre

El término incertidumbre [Kendall y Gal, 2017], en el contexto de modelos predictivos, se define como la falta de certeza asociada la respuesta del modelo. La incertidumbre es inherente de los modelos predictivos, suele estar asociados a conceptos como: el grado de confianza del modelo y la dispersión de las predicciones. Cuantificar la incertidumbre consiste en asociar una medida a la incertidumbre.

La incertidumbre se puede categorizar en dos tipos [Der Kiureghian y Ditlevsen, 2009, Gal, 2016].

- **Incertidumbre epistémica**: Asociada al conocimiento (o parámetros) del modelo. Es el resultado de la falta de conocimiento, reflejado en la práctica como la falta de datos u observaciones utilizadas para generar el modelo. La incertidumbre epistémica también es conocida como la incertidumbre del modelo. Esta tiene la característica de poder reducirse al incrementar la cantidad de datos utilizados al ajustar los parámetros de los modelos.
- **Incertidumbre aleatoria**: Asociada al ruido en las observaciones. Esta incertidumbre está ligada con la aleatoriedad intrínseca del fenómeno estudiado, tanto así como de

los instrumentos de medición utilizadas durante las observaciones. La incertidumbre aleatoria es una propiedad de los datos y no del modelo. Esta puede ser resultado, por ejemplo, del ruido en el sensor. Se encuentra fuera de alcance del modelo predictivo, por lo que a diferencia de la incertidumbre epistémica, no es posible su reducción al incrementar el volumen de datos.

### 2.5.2. Modelo Probabilístico

Para formalizar el modelamiento de la incertidumbre en DL [Gal, 2016], se utilizan las herramientas provistas por la interpretación bayesiana de la teoría de probabilidades.

Considerando el conjunto de datos $\mathcal{D} := \{X, Y\}$, donde $X := \{x_1, \ldots, x_N\}$ e $Y := \{y_1, \ldots, y_N\}$. Se desea determinar el parámetro $w$ de la función (o modelo) $y = F(x, w)$ tal que, $w$ sea el valor que mayor probabilidad de haber generador $\mathcal{D}$.

La distribución $\Omega$ captura los parámetros $w$ más probables dados nuestros datos observados, esta puede ser descrita con la distribución a *posteriori* utilizando el teorema de Bayes:

$$p(w|X, Y) = \frac{p(Y|X, w)p(w)}{p(Y|X)}$$

En esta expresión, la distribución a *priori* $p(w)$ representa la creencia previa a observar los datos $\mathcal{D}$. La distribución *likelihood* $p(Y|X, w)$, es computada por el modelo probabilístico, en este caso, una ANN. Y $p(Y|X)$ es el factor de normalización o *model evidence*.

Para determinar la *predictive distribution*, la probabilidad para un nuevo valor $x^*$, se realiza la denominada *inferencia bayesiana*, computando la siguiente integral:

$$p(y^*|x^*, X, Y) = \int p(y^*|x^*, w)p(w|X, Y)dw$$

Como señala el autor “La teoría de la probabilidad bayesiana nos ofrece herramientas con base matemática para razonar sobre la incertidumbre del modelo, pero generalmente tienen un costo computacional prohibitivo”[Gal y Ghahramani, 2016]. Esto debido a que el cálculo de $p(Y|X)$, denominado *likelihood marginalization*, requiere de una integración sobre el espacio de $w$:

$$p(Y|X) = \int p(Y|X, w)p(w)dw$$

El cual no puede hacerse de manera analítica, en consecuencia, la distribución a *posteriori* debe ser aproximada por métodos de optimización como la inferencia variacional

[Jordan *et al.*, 1999], determinando la distribución varicional parametrizada por $\theta$ óptima $q_\theta^*(w)$, la cual minimiza la divergencia Kullback-Leibler con $p(w|X,Y)$.

Permitiendo aproximar la *predictive distribution*:

$$p(y^*|x^*,X,Y) \approx \int p(y^*|x^*,w)q_\theta^*(w)dw$$

En esta nueva formulación probabilista de las ANN, los parámetros aprendidos por la red, es decir, sus pesos $W$, ya no son valores específicos, en cambio, se tiene que los parámetros de la ANN vienen dado por $w \sim \Omega$. La tarea de la ANN es aprender esta distribución a través de la aproximación $\hat{w} \sim q_\theta^*(w)$.

### 2.5.3. Estimación de la Incertidumbre Epistémica

En el trabajo [Gal y Ghahramani, 2016], se propone una implementación de BDL llamado *Monte-Carlo dropout*. Esta propone el uso de la integración de Monte-Carlo (MC) como una solución alterna a la inferencia varicional.

La implementación de MC-dropout consiste en la inclusión de capas de regularización *dropout* [Srivastava *et al.*, 2014], antes de cada capa con parámetros entrenables de la arquitectura. Esta capa de *dropout* funciona desactivando aleatoriamente, con probabilidad $D_{\text{rate}}$, las conexiones en una red neuronal, con el objetivo de regularizar la red.

Los autores de la técnica de *dropout* señalan que el uso de este se realiza solo durante la fase de entrenamiento. Sin embargo, MC-dropout propone utilizar las desactivación de conexiones durante la fase de inferencia. De esta forma, puede ser utilizado como un método de aproximación al BDL, asociando a cada peso una distribución *Bernoulli*$(1 - D_{\text{rate}})$. Este método permite ajustar la distribución $q_\theta^*(w)$ durante la fase de entrenamiento de la red, y generar muestras de los pesos de la red $\hat{w}_t$, con cada nueva instanciación del *dropout*.

Considerando a la red neuronal $F$ con parámetros $\hat{w}$, un clasificador multi-clase. La función ***softmax*** genera un vector de probabilidades (normalizadas) con un valor para cada clase, utilizando la siguiente normalización:

$$\hat{p}(y = c_k|x,\hat{w}) := \frac{\exp \hat{f}_k(x)}{\sum_j \exp \hat{f}_j(x)}$$

Donde $\hat{f}$ es el vector de predicciones no normalizados, generado por el modelo de clasificación para la entrada $x$, conocido también como ***logits***, en la capa previa a una función de normalización *softmax*.

$$\hat{p}(y|x,\hat{\omega}) := [\hat{p}(y=c_1|x,\hat{\omega}),\ldots,\hat{p}(y=c_k|x,\hat{\omega})] = softmax(\hat{f}(x)) = F(x,\hat{\omega})$$

La *predictive distribution* se puede aproximar con la repetición de $T$ *stochastics forward passes*, donde cada repetición evalúa $F(x,\hat{\omega}_t)$ utilizando un muestra $\hat{\omega}_t$ de la distribución $q^*_\theta(\omega)$.

$$\begin{aligned} p(y=c_k|x,\mathcal{D}_{\text{train}}) &= \int p(y=c_k|x,w)p(w|\mathcal{D}_{\text{train}})dw \\ &\approx \frac{1}{T}\sum_t \hat{p}(y=c_k|x,\hat{w}_t) \qquad \hat{\omega}_t \sim q^*_\theta(\omega) \\ &= \hat{p}(y=c_k|x,\mathcal{D}_{\text{train}}) \end{aligned}$$

En cuanto a la definición multi-clase, se considera el vector de tamaño $K$:

$$\hat{p}(y|x,\mathcal{D}_{\text{train}}) := [\hat{p}(y=c_1|x,\mathcal{D}_{\text{train}}),\ldots,\hat{p}(y=c_k|x,\mathcal{D}_{\text{train}})]$$

La *predictive distribution* es un estimador sesgado. El sesgo disminuye a medida que se aumenta la cantidad de muestras $T$ para la integración de MC.

La simplicidad de esta implementación de modelos bayesianos no impone alteración alguna a las herramientas y métodos ya establecidos en el campo del DL. Ajustar la distribución de los pesos a los datos de entrenamiento, se logra reutilizando los métodos de optimización basados en gradientes, sin la necesidad definir nuevas funciones de pérdida. De igual forma, con la excepción de añadir capas de *dropout*, no se imponen restricciones a la arquitectura del modelo.

Sin embargo, existe un *tradeoff* por la simplicidad. Al utilizar esta implementación, el dominio de distribuciones de los pesos $\Omega$ se acota a una familia de distribuciones de Bernoulli, donde el parámetro de probabilidad $p$ de las distribuciones debe ser ajustado manualmente, previo al entrenamiento, con el parámetro $D_{\text{rate}}$ de las capas de *dropout*.

### 2.5.4. Medidas de Incertidumbre Epistémica

Utilizando la implementación de MC-dropout es posible estimar la incertidumbre epistémica de una predicción $x$ utilizando medidas de incertidumbre para modelos de clasificación [Gal, 2016]:

- ***Predictive Entropy*** $\mathbb{H}$: Definida como la cantidad de información contenida por la *predictive distribution*.

$$\begin{aligned}\mathbb{H}[y|x, D_{\text{train}}] &:= -\sum_k p(y = c_k|x, D_{\text{train}}) \log(p(y = c_k|x, D_{\text{train}})) \\ &\approx -\sum_k (\frac{1}{T}\sum_t \hat{p}(y = c_k|x, \hat{\omega}_t)) \log(\frac{1}{T}\sum_t \hat{p}(y = c_k|x, \hat{\omega}_t)) \\ &= -\sum_k \hat{p}(y = c_k|x, \mathcal{D}_{\text{train}}) \log(\hat{p}(y = c_k|x, \mathcal{D}_{\text{train}}))\end{aligned}$$

- **Mutual Information** $\mathbb{I}$: Una medida de la dependencia mutua entre dos variables. Cuantifica la cantidad de información (en unidades como Shannon, comúnmente llamadas bits) obtenida sobre una variable aleatoria al observar la otra variable aleatoria.

$$\begin{aligned}\mathbb{I}[y, \omega|x, D_{\text{train}}] &:= \mathbb{H}[y|x, D_{\text{train}}] - \mathbb{E}_{p(\omega|D_{\text{train}})}[\mathbb{H}[y|x, \omega]] \\ &= -\sum_k p(y = c_k|x, D_{\text{train}}) \log(p(y = c_k|x, D_{\text{train}})) \\ &\quad + \mathbb{E}_{p(\omega|D_{\text{train}})}[\sum_k p(y = c_k|x, \omega) \log(p(y = c_k|x, \omega))] \\ &\approx -\sum_k (\frac{1}{T}\sum_t \hat{p}(y = c_k|x, \hat{\omega}_t)) \log(\frac{1}{T}\sum_t \hat{p}(y = c_k|x, \hat{\omega}_t)) \\ &\quad + \frac{1}{T}\sum_k \sum_t \hat{p}(y = c_k|x, \hat{\omega}_t) \log(p(y = c_k|x, \hat{\omega}_t)) \\ &= -\sum_k \hat{p}(y = c_k|x, \mathcal{D}_{\text{train}}) \log(\hat{p}(y = c_k|x, \mathcal{D}_{\text{train}}) \\ &\quad + \frac{1}{T}\sum_k \sum_t \hat{p}(y = c_k|x, \hat{\omega}_t) \log(p(y = c_k|x, \hat{\omega}_t))\end{aligned}$$

Ambas medidas de capturan distintos aspectos de la incertidumbre epistémica. En primer lugar, la *predictive entropy* mide la incertidumbre en la predicción. Mientras que la *mutual information* captura la confianza del modelo en su predicción. Este comportamiento puede observarse en la Tabla 4:

| Case | Stochastic Forward-Passes | Predictive Distribution | $\mathbb{H}$ | $\mathbb{I}$ |
|---|---|---|---|---|
| 1. | $\{(0.0, 1.0), \ldots, (0.0, 1.0)\}$ | $(0.0, 1.0)$ | $0.0$ | $0.0$ |
| 2. | $\{(0.5, 0.5), \ldots, (0.5, 0.5)\}$ | $(0.5, 0.5)$ | $\ln(2)$ | $0.0$ |
| 3. | $\{(1.0, 0.0), (0.0, 1.0), \ldots, (1.0, 0.0)\}$ | $(0.5, 0.5)$ | $\ln(2)$ | $\ln(2)$ |

Tabla 4: Ejemplo de medidas de incertidumbre epistémica - Muestra la estimación de incertidumbre en tres casos de predicciones para clasificación binaria.

El ejemplo tomado de [Gal, 2016], permite distinguir como la *predictive entropy* captura la dispersión de la predicción en los casos 2 y 3; y como la *mutual information* captura la incongruencia entre las predicciones individuales del caso 3.

### 2.5.5. Estimación de la Incertidumbre Aleatoria

El otro tipo de incertidumbre es la incertidumbre aleatoria, también conocida como incertidumbre de los datos. Este es un tipo de incertidumbre irreductible y es el ruido inherente a las observaciones. La implementación de MC-dropout no captura este tipo de incertidumbre, en cambio, la estimación de la incertidumbre aleatoria se logra implementando el método discutido en [Kendall y Gal, 2017].

El modelo aleatorio o ***aletoric model*** es un método que, al igual que el modelo epistémico, es relativamente simple de implementar sobre arquitecturas y los métodos de DL existentes.

En el caso de la tarea de clasificación, el modelo aleatorio requiere una modificación en la salida del modelo de clasificación $F$. Esta salida es remplazada por $\hat{f}$, el vector de valores *logits*; y por la medida de incertidumbre aleatoria *predictive variance* $\sigma^2$:

$$\hat{f}(x) := F^{\text{logits}}_{\text{aleatoric}}(x, \hat{w})$$

$$\hat{\sigma}^2(x) := F^{\sigma}_{\text{aleatoric}}(x, \hat{w})$$

Para generar la predicción de normalizada:

$$\hat{p}(y|x, \hat{w}) = F^{p}_{\text{aleatoric}}(x, \hat{w}) := softmax(F^{\text{logits}}_{\text{aleatoric}}(x, \hat{w}))$$

Para el entrenamiento, la función de pérdida es una versión modificada de una *negative log-likelihood*, la cual requiere de $T$ muestras de MC, utilizando la varianza estimada por el modelo:

$$\tilde{f}_t(x) := \hat{f}(x) + \hat{\sigma}^2(x) \cdot \epsilon_t, \quad \epsilon_t \sim \mathcal{N}(\vec{0}, I_K)$$

$\tilde{f}_t(x)$ es un vector *logit* corrompido con un vector de ruido, el cual es tomado de una distribución normal con media $\vec{0}$ y matriz de covarianza $\hat{\sigma}^2(x) \cdot I_K$. La función de pérdida *negative log-likelihood* se define como:

$$\mathcal{L}_{\text{aleatoric}}(\{\hat{f}(x), \hat{\sigma}^2(x)\}, y) = \text{log}(\frac{1}{T}\sum_t \text{exp}(\tilde{f}_{t,c^*}(x) - \text{log}(\sum_j \text{exp}(\tilde{f}_{t,j}(x)))) \quad (1)$$

Donde $c^*$ corresponde al índice de la clase verdadera descrito por $y$.

Esta función permite ajustar los parámetros del modelo a la tarea de clasificación, al mismo tiempo de ajustar la *predictive variance* de los datos.

## 2.6. Estado del Arte

En la siguiente sección se realizará un análisis de las técnicas y soluciones propuestas en el campo del análisis computacional de imágenes de histopatología.

El problema de HER2 *scoring*, abordado desde el desarrollo de algoritmos automáticos, ha sido ampliamente tratado en la literatura. La mayor parte de estos trabajos pueden ser categorizados en dos enfoques de investigación.

En el primero enfoque se aborda la tarea de segmentación, donde su objetivo es identificar estructuras en la imagen, por ejemplo, la membrana celular o núcleos. Este método permite el análisis morfológico por célula y la cuantificación de las proteínas de la membrana celular. Entre estos podemos encontrar trabajos como [Ficarra *et al.*, 2011], donde el problema es tratado con algoritmos clásicos de procesamiento de imágenes, utilizando filtros de deconvolución de color [Ruifrok y Johnston, 2001], para detección de la tinción DAB. Por otro lado, se han desarrollado soluciones utilizando métodos de ML, como en [Pezoa, 2017], donde el problema es abordado con *Support Vector Machines*, donde cada píxel es clasificado en "membrana sobreexpresada"(clase positiva) o "membrana no sobreexpresada"(clase negativa), describiendo cada píxel por un conjunto de características de color y textura, construidos a partir de la imagen. El problema de segmentación también a sido abordado con técnicas de DL, en [Lagree *et al.*, 2021], los autores realizan una comparación de arquitecturas CNN para segmentación de imágenes de tejidos, evaluado en el conjunto de datos MoNuSeg [Kumar *et al.*, 2017].

En el segundo enfoque, el problema es abordado directamente como un problema de clasificación binaria (positivo o negativo) o de multi-clase (0, 1+, 2+, 3+). En este apartado, se destaca la elaboración de *HER2 Scoring Constest* [Qaiser *et al.*, 2018] del Departamento de Informática de la Universidad de Warwick, Reino Unido. El evento celebrado el año 2016, constó con la participación de 105 equipos. La mayor parte de las soluciones propuestas compartían un método de tres etapas: pre-procesamiento de la imagen WSI, esto es, dividir la imagen en pequeños *patches*; clasificación de *patches*, utilizando arquitecturas de ANN del estado del arte para la clasificación de imágenes; y pos-procesamiento de los resultados, el cual a partir de las predicciones de los *patches*, generan un *scoring* del tejido. Actualmente el *dataset* está disponible para fines académicos, utilizado por diversos trabajos para el entrenamiento y evaluación de nuevos modelos.

Sin embargo, también hay trabajos que abordan ambos enfoques simultáneamente. Entre estos se destacan *Her2Net* [Saha y Chakraborty, 2018], una arquitectura de DL que combina capas convolucionales y recurrentes, para segmentar las membranas celulares y núcleos, como cuantificación de la sobreexpresión, utilizada para el HER2 *scoring*.

En cuanto al enfoque bayesiano, ARA [Rączkowski *et al.*, 2019] es un *framework* de *active learning*, para el entrenamiento de un modelo de clasificación. Aborda un problema similar a HER2 *scoring*, la discriminación de tejido en imágenes WSI de pacientes con cáncer colo-

rrectal. Utiliza un enfoque BDL en conjunto con la estrategia de aprendizaje supervisado *active learning*. Esta estrategia consiste en un entrenamiento iterativo, donde las imágenes con una alta incertidumbre estimada, son revisadas por un patólogo para luego ser re-entrenar el modelo con estas. Logrando minimizar la cantidad de imágenes etiquetadas necesarias para entrenar al modelo. Este trabajo demuestra la factibilidad de implementar modelos basados en BDL para la clasificación de imágenes, en especial, en el contexto histopatológico y las imágenes de alta resolución WSI.

De los trabajos anteriores, se destaca la importancia de la deconvolución de color[Ruifrok y Johnston, 2001], siendo una característica relevante tanto para la segmentación como la clasificación. Un tema recurrente en los trabajos visto es la La selección de la región de interés (ROI). Sin embargo, este tema se encuentra fuera del alcance de esta memoria. Finalmente, el problema de clasificación de una DS es abordado con la misma estrategia, dividir de la imagen de alta resolución en pequeños *patches*, posibilitando el uso de modelos de DL para la clasificación.

# CAPÍTULO 3
# PROPUESTA DE SOLUCIÓN

Esta memoria se desenvuelve con el desarrollo del *framework* para el análisis de WSI utilizando algoritmos de BDL. En este capítulo, se describirá el *framework* `her2bdl`, destacando las características principales de la herramienta desarrollada y su funcionamiento para el análisis de las *digital slides*.

Junto a esto, se estudia como la estimación de la incertidumbre, a través de la implementación de modelos de BDL, que pueden ser aplicados a escenarios médicos como herramienta para la interpretabilidad de los modelos de DL. En este contexto, esta memoria propone las meta-arquitectura de submodelos `MC-dropout model` y `aleatoric model`, el algoritmo *2-step predictive distribution*, para el cálculo eficiente de la *predictive distribution*.

El capítulo termina con la definición de las tareas de clasificación de imágenes, las cuales son abordadas en el capítulo siguiente. Se entrega una descripción de los conjuntos de datos y sus objetivos.

La siguiente lista agrupa los recursos mencionados en este capítulo:

- **Repositorio her2bdl**: https://github.com/sborquez/her2bdl/tree/dev
- **Docker Image**: https://hub.docker.com/repository/docker/sborquez/her2bdl
- **Reportes en Weights & Biases**: https://wandb.ai/sborquez/her2bdl/reportlist

## 3.1. her2bdl: Framework para Clasificar Tejidos en Imágenes de Alta Resolución

`her2bdl: Bayesian Deep Learning for HER2 tissue image classification` [3] es un proyecto de código abierto desarrollado en el lenguaje de programación `Python`. Su objetivo principal es el análisis para WSI y evaluación de la sobreexpresión de la proteína HER2, con el uso de algoritmos de BDL.

`her2bdl` permite agilizar el proceso de la investigación y el desarrollo de nuevas soluciones para problemas de clasificación de imágenes, especializándose en el problema de *HER2 scoring* en imágenes de WSI.

[3] her2bdl: Bayesian Deep Learning for HER2 tissue image classification: https://github.com/sborquez/her2bdl

Dispone de una variedad de herramientas de uso general, útiles para las distintas etapas del ciclo de vida del ML. Permite el diseño, entrenamiento y evaluación de modelos de DL, con soporte para modelos bayesianos y el estimación de la incertidumbre (BDL). Además, `her2bdl` integra herramientas para facilitar la colaboración, replicación los experimentos y divulgación de resultados.

A continuación se describirán las principales características de `her2bdl`.

- **Estructura modular**: `her2bdl` define una estructura de desarrollo que separa los tres módulos de un proyecto de DL: Modelos (`model`) para la implementación de arquitecturas adaptadas al problema abordado; datos (`data`) para la extracción, transformación y carga de datos, utilizados para alimentar a los modelos; y visualización (`visualization`) con herramientas de exploración de los datos, medidas de desempeño y visualización de resultados. El objetivo de este diseño es facilitar la extensión *framework*, así como la adaptación a nuevos contextos distintos al análisis de DS.

- **Soporte para software de gestión de carga de trabajo**: Dispone de *scripts* para la ejecución de tareas en ambientes de high-performance computing (HPC), utilizando el gestor de carga de trabajo `Slurm` [4]. Así como soporte para software de gestión de colas `Task-Spooler` [5] para ejecución en cola en una máquina local.

- **Definición de modelos de DL y BDL extendiendo la API de `Keras`**: Esta API proporciona una abstracción de alto nivel a través del uso de los modelos y capas (*Models & layers*) esenciales para el diseño de las redes neuronales. `Keras` [Chollet, 2015] a sido adoptado por la comunidad científica gracias a su simple interfaz que permite una rápida experimentación. Actualmente, `Tensorflow 2` [Abadi *et al.*, 2015] a integrado internamente el la API de `Keras`, permitiendo aprovechas las capacidades de multiplataforma y entrenamiento eficiente en GPU.

- **Desacoplar experimentos del código**: Con el uso de archivos de configuración, los experimentos son definidos en un formato estandarizados utilizando el lenguaje *YAML*. Separar los experimentos del código permite simplifica el proceso de definir, ejecutar y compartir experimentos.

- **Dockers containers**: Con el objetivo de facilitar la portabilidad del ambiente de desarrollo, entrenamiento y evaluación, `her2bdl` utiliza *Dockers containers* [6] para encapsular el código y las dependencias en una unidad auto-contenida. El *framework* puede ser compartido y ejecutado de forma rápida y fiable en distintos ambientes, independientemente de si el hardware corresponde a una máquina local, HPC o servicios *cloud*. La imagen de `her2bdl` se encuentra disponible en https://hub.docker.com/repository/docker/sborquez/her2bdl.

[4] Slurm - Workload Manager. https://slurm.schedmd.com/overview.html
[5] Task-Spooler - A simple unix batch system. https://vicerveza.homeunix.net/~viric/soft/ts/
[6] What is a Container? - https://www.docker.com/resources/what-container

- **Integración con *Weights & Biases***: La plataforma de *Weights & Biases* (W&B) [Biewald, 2020] ofrece servicios de monitoreo y reportes a través de su aplicación web [7] para facilitar la colaboración y compartir resultados. `her2bdl` integra las herramientas de W&B a través se su API en `Python`. Esto permite registrar en tiempo real el estado actual de los experimentos, visualizar las diferentes métricas evaluadas, y la elaboración de reportes. W&B es un servicio independiente del equipo utilizado para entrenar modelos, lo que permite monitorear tareas ejecutándose en distintos nodos de computación: local, HPC, o *cloud*. Los experimentos realizados para esta memoria se encuentran disponibles en https://wandb.ai/sborquez/her2bdl/reportlist.

`her2bdl` es un *framework* enfocado en el desarrollo, entrenamiento y evaluación de modelos de BDL en el ámbito de WSI. Sin embargo, la mayor parte de sus características son transversales e independientes del contexto utilizado. `her2bdl` fue diseñado con el objetivo de ser reutilizado, extendido y adaptarlo para distintos casos de usos de BDL.

## 3.2. Análisis de Imágenes de Alta Resolución

Una DS posee una resolución que resulta prohibitiva, en cuanto al uso de memoria y tiempo de cómputo requerido, para la aplicación de modelos de DL. Es por esto que, al trabajar con estas imágenes con de alta resolución, resulta necesario la implementación de etapas extras, previas al entrenamiento y evaluación del modelo.

Con el fin de utilizar los modelos clásicos de DL, se opta por una estrategia de dividir las imágenes de alta resolución en pequeñas secciones o *patches*. A partir de una de las muestras contenidas en la DS, se generan un conjunto de imágenes de menor tamaño, las cuales son utilizadas para entrenar al modelo. Realizar esta división permite aumentar el volumen de datos, impactando positivamente el desempeño de los modelos entrenados.

Sin embargo, la tarea de clasificación de la sobreexpresión debe ser a nivel de la muestra y no de los *patches*. Con el objetivo de generar una predicción de la muestra, se debe considerar una etapa de post-procesamiento que permita obtener una predicción a nivel de muestra, a partir de las predicciones individuales obtenidas del conjunto de *patches*.

De esta forma, el flujo de trabajo para el procesamiento de una DS está compuesto por cuatro etapas:

1. Preprocesamiento asistido de *digital slide*.
2. Generación de conjuntos de datos.

[7] Weights & Biases - Developer tools for machine learning. https://wandb.ai/site

3. Definición de modelo de clasificación.
4. Evaluación y estimación de incertidumbre.

### 3.2.1. Preprocesamiento Asistido de *Digital Slide*

La primera etapa, ilustrada en la Figura 10, se basa en la artículo de *IBM Developers* [Eriksson y Hu, 2018]. Consiste en localizar las regiones de interés o *region of interest* (ROI) presentes en una DS. A partir de estas regiones, se generarán los conjuntos de datos necesarios para el entrenamiento y evaluación del modelo.

Se considera una ROI como una región de la imagen la cual contiene características relevantes para la discriminación de la sobreexpresión. En particular, se considera una ROI en WSI a las regiones de las imágenes con muestras del tejido histopatológico.

Con el objetivo de determinar las ROIs, se implementa un método semi-automático que permita identificar objetos relevantes contenidos en una DS. El método descrito a continuación, requiere de la asistencia humana para discriminar en última instancia cuáles de los objetos identificados son relevantes para el modelo.

La etapa de *preprocesamiento asistido de digital slide* se divide en cuatro pasos:

(a) **Generación de imagen**

Consiste en la generación de una imagen de baja resolución de la imagen original. Esto se logra gracias a la estructura piramidal multi-nivel de una DS.

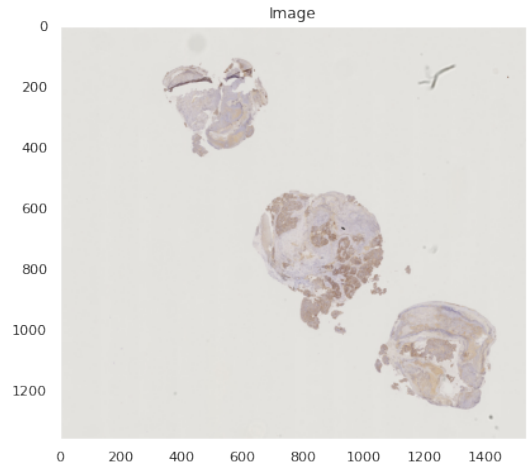
La imagen generada, al tener una escala mucho menor que la imagen original, permite realizar operaciones comunes de procesamiento de imágenes digitales: aplicación de filtros, operaciones morfológicas, y *thresholding*; Sin aumentar en gran medida por el tiempo de cómputo requerido.

(b) **Separación de primer plano de fondo.**

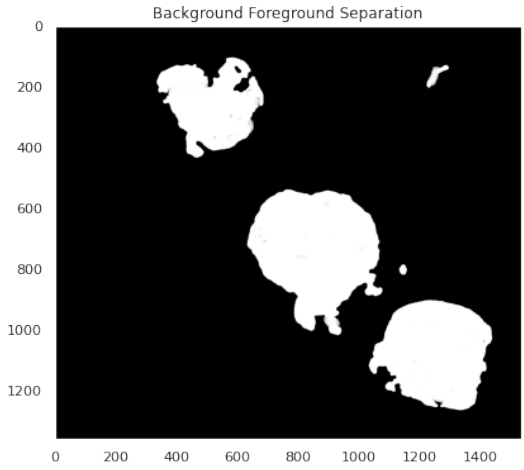
Determinar las regiones que contienen información se logra al segmentar la imagen en dos clases: fondo, caracterizado por ser una región uniforme de color gris, y el primer plano, compuesto por estructuras y tejidos de la muestra histopatológica.

Al considerar que las muestras presentan una variedad de estructuras, colores y características que derivan de las estructuras celulares, el procedimiento de separación de primer plano de fondo está compuesto por el consenso de la aplicación de *threshold de Otsu* [8] a cinco características derivadas de la imagen: intensidad del complemento de canal verde, para resaltar las estructuras intracelulares de tonos azules y rojos; intensidad del canal Hue en el espacio HSV, relacionado con la saturación de la imagen;

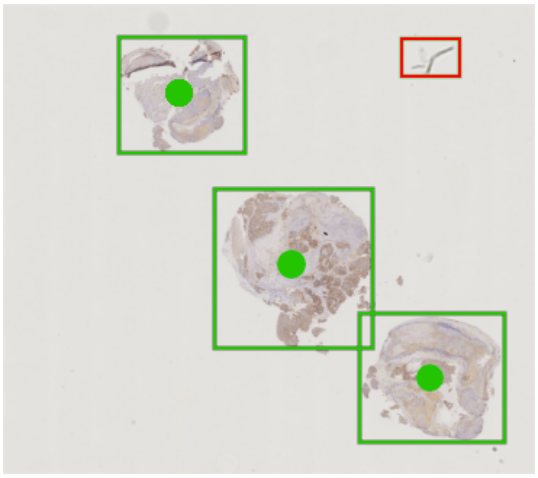
[8] Otsu Thresholding with scikit-image in Python: https://scikit-image.org/docs/stable/api/skimage.filters.html#threshold-otsu



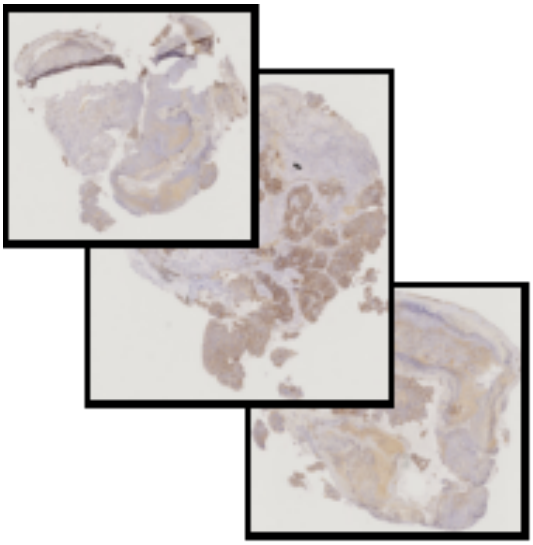

(a) Generación de imagen.




(b) Separación de primer plano de fondo.



(c) Localización y discriminación de objetos.



(d) Identificación y registro de objetos.

Figura 10: Preprocesamiento asistido de *digital slide* - Consiste en cuatro pasos para extraer los objetos relevantes a partir de una DS. En este ejemplo, la imagen es cargada a una resolución tratable (a) y se le aplica la separación de fondo (b). Como resultado, cuatro objetos son detectados automáticamente, pero solo tres elementos (en color verde) fueron seleccionados por el experto (c) y pasan al siguiente paso (d). El objeto desconocido en rojo ignorado. Fuente: Elaboración propia.

entropía, relacionado con la complejidad o uniformidad contenida en un vecindario de píxeles; y las intensidades de tinción hematoxilina y DAB. Cada una de estas características, al aplicar el *threshold* de Otsu, resaltará distintas estructuras de los tejidos.

Luego, se aplican a la segmentación resultante una operación morfológica de apertura (*opening*) seguida de una operación de cierre (*closing*) con el objetivo de remover ruido. La operación de apertura remueve pequeñas estructuras, mientras que la ope-

ración de cierre remueve agujeros en la máscara de segmentación.

(c) **Localización y discriminación de objetos.**

La localización de objetos de interés se realiza al determinar las regiones segmentadas que cumplen con un mínimo de área predefinida.

Como resultado del paso (b), se obtiene una segmentación robusta a las variaciones de color y estructura presente en los tejidos. Sin embargo, la presencia de anomalías en la imagen, como pueden ser marcas de tinta, desenfoque, u elementos ajenos a la muestra, pueden resultar la preselección de falsos positivos. De igual forma, algunas imágenes pueden contener una muestra de contraste, es decir, una muestra con un *score* de sobreexpresión conocido, utilizada por el patólogo para comparar con la muestra del paciente. Dado lo anterior, es necesaria la intervención humana en el procedimiento para discriminar las regiones propuestas por el algoritmo no son de interés, o también dar la posibilidad de proponer nuevas regiones.

(d) **Identificación y registro de objetos.**

Finalmente, las regiones seleccionadas son identificadas y almacenadas en un archivo CSV, en el cual se identifican tanto la imagen de origen, coordenadas en la imagen, dimensiones de la región y la clase de sobreexpresión a la que pertenece si se encuentra disponible.

La lista resultante es dividida para generar los conjuntos de entrenamiento (*train*) y evaluación (*test*), teniendo en consideración el balance de clases de sobreexpresión HER2.

Es preciso señalar que la predicción se realiza a nivel de objeto, por lo que la división de los conjuntos de datos se realiza en base a la lista de objetos identificados. Se debe considerar también que una DS puede tener varias muestras (objetos identificados) de un mismo paciente, en consecuencia, objetos pertenecientes a un mismo DS pueden terminar en conjuntos de datos distintos. Sin embargo, en la práctica esta división no afecta a la evaluación de modelo.

### 3.2.2. Generación de Conjunto de Datos

La segunda etapa utiliza la lista de objetos identificados, a partir de esta genera pequeños recortes (*patches*) de cada uno de los objetos en la lista. Los *patches* no son generados a partir de la imagen original de alta resolución, el alto nivel de detalle que se obtiene a partir de una imagen obtenida por el microscopio digital, con magnificación x40 (nivel 0), no son necesarios para la tarea de clasificación.

Se debe considerar el equilibrio entre un campo de percepción amplio que permita la mayor cantidad de información del espécimen, cantidad de *patches* generados para el entrenamiento del modelo, y el nivel de detalle de las estructuras celulares relevantes para determinar

el *score*. Teniendo esto en consideración, se determinó utilizar *patches* con una resolución de $240 \times 240$ píxeles, obtenidos a partir de un *downsampling* de $8.0$ (nivel 3) de la imagen original.

Para la generación de *patches* se implementan dos estrategias de generación (ver Figura 11), destinadas al evaluación y entrenamiento respectivamente:

- **Grid Patch Generator**: Es una estrategia determinista destinada a la evaluación de muestras. La muestra es dividida en una grilla equiespaciada sin superposición, abarcando cada región de la muestra.
- **Monte-Carlo Patch Generator**: Es una estrategia estocástica destinada al entrenamiento del modelo. Genera para cada iteración del entrenamiento modelo un nuevo conjunto de *patches*, tomados aleatoriamente de la muestra considerando una distribución uniforme en la región de interés. Es una técnica de *data augmentation* para aumentar tanto el volumen y como la variabilidad del conjunto de entrenamiento, el cual, al utilizar junto a rotaciones aleatorias respecto a sus ejes horizontal y vertical, permite reducir el error de validación durante el entrenamiento [Combalia y Vilaplana Besler, 2018].

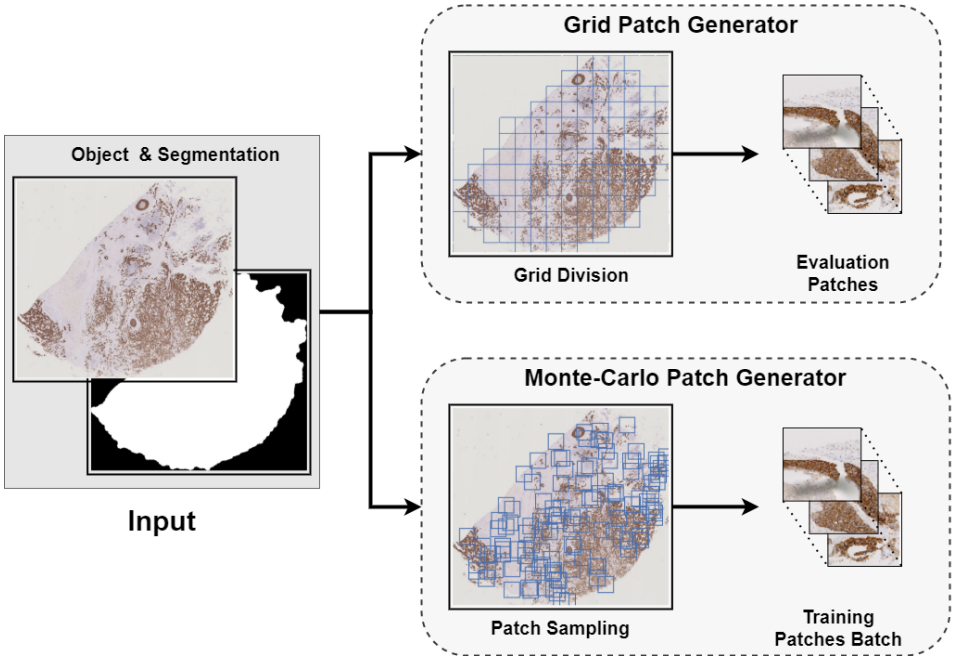



Figura 11: Patch Generators - Un objeto identificado puede ser convertido en un conjunto de pequeñas imágenes para ser utilizadas por un modelo de DL. Se disponen de dos estrategias de división: *grid*, para una división determinista sin superposición; o *sampling* para generar muestras aleatorias de toda la imagen.

Fuente: Elaboración propia.

Es posible que durante la generación de *patches*, independiente de la estrategia utilizada, se generen imágenes conformada en mayor parte por fondo. Utilizando la máscara de segmentación provista de la etapa anterior, los generadores pueden determinar el porcentaje del *patch* correspondiente al fondo, de esta forma filtrar aquellas imágenes sin contenido relevante.

### 3.2.3. Definición de Modelo de Clasificación

Durante la tercera etapa, se determinan los hiper-parámetros de la arquitectura del modelo. A través de un proceso iterativo de entrenamiento, validación y ajuste de hiper-parámetros, es posible definir un modelo óptimo para el problema de clasificación de *patches*.

En esta etapa, se utilizan las metodologías ya establecidas de DL para el ajuste de los parámetros de los modelos.

### 3.2.4. Evaluación y Estimación de Incertidumbre

Finalmente, en la cuarta etapa (ver Figura 12) utiliza el modelo definido en la etapa anterior para predecir la clase correspondiente a cada una de las muestras pertenecientes al conjunto de evaluación. Para lograr esto, el modelo realiza una predicción y estimación de incertidumbre epistémica para cada *patch* de la muestra. Estas predicciones individuales luego son agregadas para obtener un consenso respecto a la clase de la muestra. La discusión de los métodos de agregación será abordada en la Sección 3.5.

## 3.3. Modelos de Incertidumbre

Como se describe en la Sección 2.5, el BDL es un enfoque que extiende la familia de algoritmos de DL. Otorga nuevas herramientas a los modelos de ANN, dando la capacidad de estimar la incertidumbre.

Se diferencian dos tipos o fuentes de incertidumbre en los modelos de BDL: **incertidumbre epistémica**, referente a la desconocimiento propio del modelo; y la **incertidumbre aleatoria**, la cual es intrínseca de los datos.

Para ambos casos es posible modificar las arquitecturas y algoritmos de DL para modelar la incertidumbre. En esta sección se describe la construcción de dos tipos de modelos, diseñados para capturar la incertidumbre epistémica y aleatoria respectivamente.

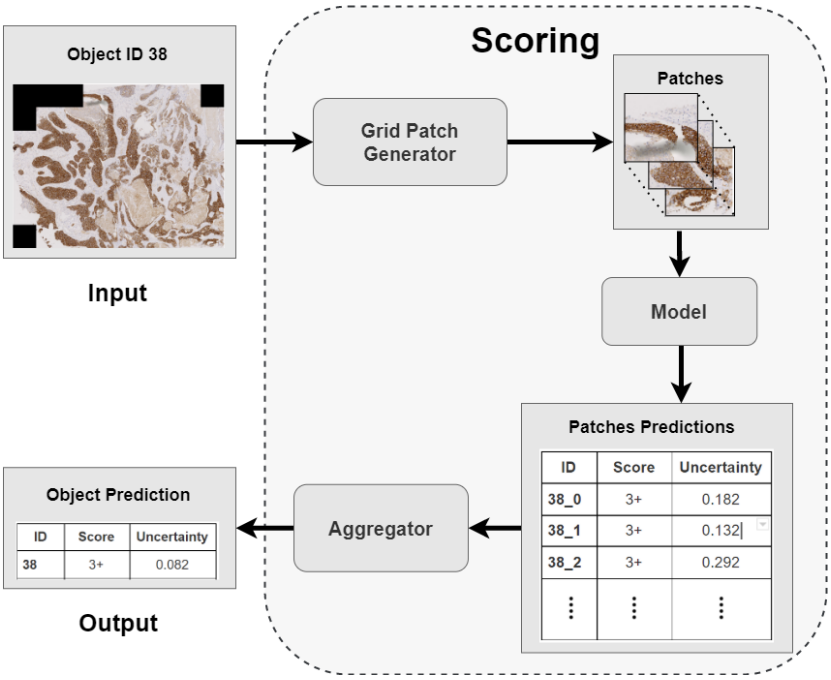



Figura 12: Evaluación y estimación de incertidumbre (*scoring*) - A partir de la predicción individual de *patches* que componen a la imagen de alta resolución, se construye una respuesta de consenso para la muestra.

Fuente: Elaboración propia.

### 3.3.1. Incertidumbre Epistémica: Monte-Carlo Classification Model

El modelo para capturar la incertidumbre epistémica está basado en el algoritmo de *Monte-Carlo dropout* (*MC-dropout*) propuesto en [Gal y Ghahramani, 2016], discutido en la Sección 2.5. En el artículo, el autor propone una aproximación de los modelos bayesianos con el uso de las capas de *dropout*. Esta aproximación consiste en agregar una capa de *dropout* entre cada capa de una arquitectura clásica DL, manteniendo los *dropout* activos durante la fase de entrenamiento y la fase inferencia.

El uso de *MC-dropout* presenta una serie de ventajas frente a otras implementaciones de BDL. Una integración perfecta con el *framework* de DL ya establecido. Al solo requerir añadir capas de *dropout*, la propagación del error y el procedimiento de entrenamiento no se ven afectados, esto permite utilizar los algoritmos basados en *backpropagation* para el ajuste de los pesos del modelo.

Sin embargo, el uso de *MC-dropout* también impone algunas desventajas, en particular en la representatividad de distribuciones de los pesos del modelo.

En esta memoria se propone la meta-arquitectura **Monte-Carlo dropout classification model**, también referida como `MC-dropout model`. Esta meta-arquitectura es una abstracción sobre la definición de los modelos de clasificación de imágenes, utilizando la implementación

de *Monte-Carlo dropout* para representar la incertidumbre.

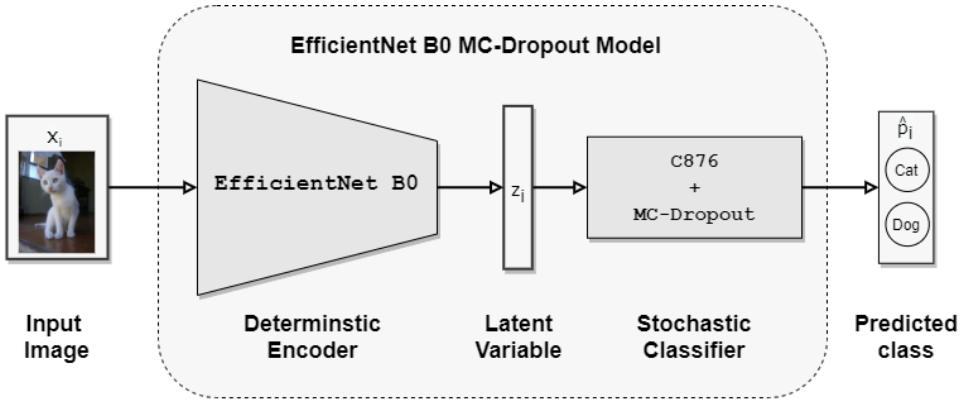



Figura 13: Arquitectura de EfficientNet B0 MC-Dropout - Instancia de la meta-arquitectura MC-dropout model para el problema de clasificación binaria *cats vs. dogs*.
Fuente: Elaboración propia.

`MC-dropout model` establece la definición de un modelo bayesiano $F$ a partir de la composición de dos submodelos $E$ y $C$:

- `Encoder` $E$: El primer modelo se especializa en la extracción de características de la imagen de entrada, codificándola en un vector de variables latentes $z$. $E$ es un **modelo determinista**, es decir, para una imagen dada $x_i$, el modelo siempre computará el mismo resultado $z_i$.

  $$z = E(x, \hat{\omega}^{(E)})$$

  Donde $\hat{\omega}^{(E)}$ denota a los parámetros del modelo $E$. El modelo `Encoder` puede ser definido y entrenado desde cero, no obstante, el enfoque presentado permite además reutilizar arquitecturas ya establecidas para la clasificación de imágenes. Para lograr esto, se utiliza la estrategia de *transfer learning* [Weiss *et al.*, 2016]. Esta estrategia consiste en reutilizar el conocimiento adquirido por una ANN al resolver una tarea, adaptar la ANN a una nueva tarea similar, finalmente reajustar el modelo a los nuevos datos. Esto permite reducir la cantidad de datos necesarios para obtener resultados competentes.

  En la práctica, esto se logra reemplazando las últimas capas de una ANN con nuevas capas, adaptando el modelo a las nuevas clases definidas por el nuevo problema, reutilizando las capas previas y sus pesos ajustados al problema original. Sin embargo, dado que la salida `Encoder` consiste en un vector de variables latentes, solo requiere remover las últimas capas del modelo original, sin añadir nuevas capas adaptadas a las nuevas clases.

Esta estrategia permite que el `Encoder` pueda ser reemplazado por cualquiera de los modelos pre-entrenados[9] disponibles en el paquete de `Keras`. Por ejemplo, en la arquitectura mostrada en la Figura 13, el `Encoder` corresponde a una instancia de la arquitectura `EfficientNet` [Tan y Le, 2019].

- `Classifier` $C$: El segundo modelo utiliza como entrada el vector latente $z$ computado por $E$, y predice para la imagen $x$ su grado pertenencia a las $k$ clases. $C$ es un **modelo estocástico** (*stochastic model*) cuya salida varia dado una misma entrada $z$, implementa el método de *MC-dropout*, y en su última capa realiza una operación *softmax*.

$$\hat{p}(y|z, \hat{\omega}_t^{(C)}) = C(z, \hat{\omega}_t^{(C)})$$

$$p(y = c_k|z, \mathcal{D}_{\text{train}}^{(z)}) \approx \frac{1}{T} \sum_t \hat{p}(y = c_k|z, \hat{\omega}_t^{(C)})$$

$$\hat{y} = \text{argmax}_k p(y = c_k|z, \mathcal{D}_{\text{train}}^{(z)})$$

Donde $\mathcal{D}_{\text{train}}^{(z)}$ corresponde al conjunto de datos en el espacio de variables latentes, $\hat{\omega}_t^{(C)}\}$ denota a los parámetros del modelo $C$, y el subíndice $t$ denota a una muestra de las $T$ *forward passes*.

La propuesta de *Monte-Carlo classification model* solo define los requisitos de entrada y salida del modelo $C$ y la implementación de *MC-dropout*. Satisfaciendo estas restricciones, el resto del diseño de la arquitectura es libre, puede definirse y adaptarse al problema que se desea resolver.

En este trabajo, se utiliza un arquitectura para el modelo `Classifier` definida por una secuencia de capas densas con *MC-dropout*. Por simplicidad, se utiliza la notación **CXX** para representar la configuración de la arquitectura del modelo `Classifier`. Al usar esta notación, cada X representa a una capa densa con $2^X$ unidades, seguida de una capa *dropout*. De esta forma, la configuración **C876** utilizada en la Figura 13, representa a una ANN de tres capas densas con 256, 128, y 64 unidades respectivamente.

La composición de estos dos modelos da forma al *Monte-Carlo classification model* $F$:

$$\hat{p}(y|x, \hat{\omega}_t) = F(x, \hat{\omega}_t) = C(E(x, \hat{\omega}^{(E)}), \hat{\omega}_t^{(C)})$$

$$p(y = c_k|x, \mathcal{D}_{\text{train}}) \approx \frac{1}{T} \sum_t p(y = c_k|x, \hat{\omega}_t)$$

[9] Keras Applications - https://keras.io/api/applications/

$$\hat{y} = \text{argmax}_k p(y = c_k | x, \mathcal{D}_{\text{train}})$$

Donde $\hat{\omega}_t := \{\hat{\omega}^{(E)}, \hat{\omega}_t^{(C)}\}$ denota a los parámetros del modelo $F$.

La Figura 13 es una instancia la meta-arquitectura propuesta, adaptada a la clasificación binaria de perros y gatos. En la Sección 4.1 se describirá con más detalle esta arquitectura.

### 3.3.2. *Predictive Distribution*: Estimación Eficiente

El Algoritmo 1 es la implementación utilizada para estimar de la *predictive distribution* $p(y|x, \mathcal{D}_{\text{train}})$. Esta requiere múltiples *forward passes* con la misma entrada $x_i$. El cálculo de la *predictive distribution* puede resultar redundante dado que el resultado del modelo determinista $E$, la variable latente $z_i$, será la misma para cada *forward pass*. Junto a esto, se debe considerar que el número de operaciones realizadas por el modelo determinista $E$ suele ser mayor que el del modelo de estocástico $C$, por lo que la operación utiliza más recursos de los necesarios.

**Algorithm 1** Estimación de *predictive distribution*.

**procedure** PredictiveDistribution($F, x, T$)
    $Y \leftarrow \{\}$
    **for** $t$ **in** [1, ..., $T$] **do**
        $y_t \leftarrow$ F(x) ▷ *Forward pass* estocástico.
        $Y \leftarrow Y \cup y_t$
    $p \leftarrow$ `batch_mean`($Y$) ▷ Predictive distribution.
    **return** $p$

La estructura de modelos compuestos permite la formulación del algoritmo ***2-steps predictive distribution*** (ver Algoritmo 2) la cual propone un cálculo eficiente de la *predictive distribution* reduciendo el tiempo de ejecución.

**Algorithm 2** Estimación eficiente de *predictive distribution*.

**procedure** 2-Step Predictive Distribution($F, x, T$)
    ($E, C$) $\leftarrow F$ ▷ División en submodelos.
    $z \leftarrow E(x)$ ▷ Paso 1: *Forward pass* determinista.
    $Z \leftarrow$ `repeat`($z, T$) ▷ Paso 2: *Forward pass* estocástico.
    $Y \leftarrow C(Z_i)$
    $p \leftarrow$ `batch_mean`($Y$) ▷ Predictive distribution.
    **return** $p$

Esta mejora se fundamenta en la división de la operación de *forward pass* en dos pasos, correspondientes a los submodelos deterministas y estocásticos.

En un primer paso, se computa el *forward pass* con el modelo de determinista $E$ para una entrada $x$, obteniendo la variable latente $z$, evitando las operaciones redundantes.

En el segundo paso, la variable latente $z$ es copiada a un *batch* de tamaño $T$, el modelo de estocástico $C$ calcula las $T$ muestras del *stochastic forward pass* realizando solo un *forward pass*, de esta forma beneficiándose de la paralelización de la GPU.

Esta descripción puede extenderse fácilmente para *batch* de entradas $X$, como muestra el Algoritmo 3.

**Algorithm 3** Estimación eficiente de *predictive distribution* para un batch $X$.

**procedure** 2-Step Predictive Distribution - Batch($F, X, T$)
  ($E$, $C$) $\leftarrow F$ ▷ División en submodelos.
  $Z \leftarrow E(X)$ ▷ Paso 1: *Forward pass* determinista.
  $P \leftarrow \{\}$ ▷ Paso 2: *Forward pass* estocástico.
  **for** $z_i$ **in** $Z$ **do**
    $Z_i \leftarrow \texttt{repeat}(z_i, T)$
    $Y_i \leftarrow C(Z_i)$
    $p_i \leftarrow \texttt{batch_mean}(Y_i)$ ▷ Predictive distribution.
    $P \leftarrow P \cup p_i$
  **return** $P$ ▷ Predictive distribution batch.

Al ajustar los tamaños de los *batches* para cada modelo según las capacidades computacionales del *hardware* utilizado, es posible reducir el tiempo cómputo y los recursos necesarios al calcular la *predictive distribution*. La Tabla 5 muestra la mejora en el tiempo de ejecución al utilizar el método propuesto, estos tiempos fueron obtenidos repitiendo 10 veces cada operación, utilizando el modelo de clasificación binaria mostrado en la Figura 13 y una entrada $X$ de 32 imágenes.

| **EfficientNet MC-Dropout** | | **Forward Pass** $T = 1$ | **Predictive Distribution** $T = 200$ | **2-Steps Predictive Distribution** $T = 200$ |
|---|---|---|---|---|
| **Inter i7-7500U** | $F(X)$ | 5.449 ± 1.745 [s] | 111.770 ± 8.774 [s] | 12.039 ± 1.373 [s] |
| | $E(X)$ | 6.269 ± 5.100 [s] | 114.534 ± 34.545 [s] | 3.673 ± 0.492 [s] |
| | $C(Z)$ | 0.347 ± 0.104 [s] | 5.456 ± 0.657 [s] | 10.142 ± 1.892 [s] |
| **GeForce RTX 3060** | $F(X)$ | 4.086 ± 10.775 [s] | 38.424 ± 0.825 [s] | 3.721 ± 0.005 [s] |
| | $E(X)$ | 0.262 ± 0.195 [s] | 39.343 ± 0.050 [s] | 0.195 ± 0.001 [s] |
| | $C(Z)$ | 0.067 ± 0.015 [s] | 12.173 ± 0.015 [s] | 3.951 ± 0.003 [s] |

Tabla 5: Tiempo de ejecución para la estimación de *Predictive Distribution*

### 3.3.3. Incertidumbre Aleatoria: Aletoric Classification Model

Como se discute en la sección 2.5, para modelar la incertidumbre aleatoria se utiliza la implementación planteada en [Kendall y Gal, 2017]. Aplicando este método, es posible estimar la incertidumbre aleatoria, la **varianza predictiva**, para cada entrada $x_i$.

Con el fin reutilizar una arquitectura definida para el modelo epistémico, se propone la meta-arquitectura **aleatoric classification model** (`aleatoric model`), cuya construcción se basa en la modificación de una instancia de `MC-dropout model` existente, convirtiendo el modelo a una versión determinista y añadiendo la varianza como salida del modelo:

- `Encoder` $E$: El primer modelo no sufre modificaciones. En cambio, el modelo y sus pesos entrenados son copiados al nuevo `aleatoric model`.

$$z = E(x, \hat{w}^{(E)})$$

- `Aleatoric Classifier` $C_{\text{aleatoric}}$: Para satisfacer estos requisitos, el modelo estocástico $C$ es modificado. La primera modificación consiste en la eliminación de las capas *dropout*, esto permite obtener una versión determinista de `Classifier`.

  En segundo lugar, se debe añadir la varianza predictiva $\hat{\sigma}^2$ como una la salida del modelo.

  Finalmente, dado que el método funciona añadiendo un ruido de una distribución normal $\epsilon \sim \mathcal{N}$, a los valores previos a la operación *softmax*. Esta operación de normalización es separada del modelo $C_{\text{aleatoric}}$, como resultado el modelo se puede reescribir como:

$$\hat{c}(z) := C^{\text{logits}}_{\text{aleatoric}}(z, \hat{w}^{(C)})$$

$$\hat{p}(y|z, \hat{w}^{(C)}) = C^{p}_{\text{aleatoric}} := softmax(\hat{c}(z))$$

$$\hat{\sigma}^2(z) = C^{\sigma}_{\text{aleatoric}}(z, \hat{w}^{(C)})$$

  Donde $\hat{c}$ corresponde a los valores *logits*, y $\hat{\sigma}^2$ es la estimación de la varianza. Notar que en esta versión los pesos $\hat{w}^{(C)}$ ya no dependen del subíndice $t$.

De esta forma, la composición del modelo es la siguiente:

$$\hat{f}(x) := F^{\text{logit}}_{\text{aleatoric}}(x, \hat{\omega}) = C^{\text{logits}}_{\text{aleatoric}}(E(x, \hat{\omega}^{(E)}), \hat{\omega}^{(C)})$$

$$\hat{p}(y|x, \hat{\omega}) = F^{p}_{\text{aleatoric}}(x, \hat{\omega}) := softmax(C^{\text{logits}}_{\text{aleatoric}}(E(x, \hat{\omega}^{(E)}), \hat{\omega}^{(C)}))$$

$$\hat{\sigma}^2(x) = F^{\sigma}_{\text{aleatoric}}(x, \hat{\omega}) = C^{\sigma}_{\text{aleatoric}}(E(x, \hat{\omega}^{(E)}), \hat{\omega}^{(C)}))$$

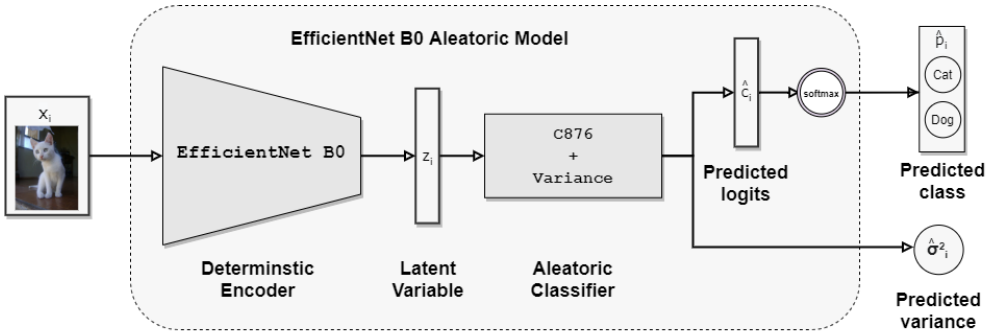



Figura 14: Arquitectura de EfficientNet B0 Aleatoric - Instancia del modelo aleatorio, construido a partir de la instancia definida para el problema de clasificación binaria *cats vs. dogs*. Fuente: Elaboración propia.

La Figura 14 ejemplifica el modelo aleatorio, este fue obtenido a partir de la instancia utilizada en la Figura 13.

Para el entrenamiento, el modelo utiliza la función de pérdida *negative log likelihood* (ver Ecuación 1). Esta puede ser reescrita utilizando la definición de submodelos:

$$\tilde{p}_t(y|x,\hat{w}) := softmax(\hat{f}(x) + \hat{\sigma}_i^2(x) \cdot \epsilon_t)$$

$$\mathcal{L}_{\text{aleatoric}}(\{\hat{f}(x), \hat{\sigma}^2(x)\}, y) = \text{log}(\frac{1}{T}\sum_t \tilde{p}_t(y = C^*|x,\hat{w})) \quad (2)$$

Donde $\tilde{p}_t(y|x,\hat{w})$ es una predicción corrompida por $\epsilon_t$, para la entrada $x$. $C^*$ es la clase correspondiente de la entrada $x$. Notar que los pesos en esta versión están fijos, debido a que el modelo es determinista. En cambio, el comportamiento estocástico se encuentra ahora en la evaluación de la función de pérdida con $T$ muestras de $\epsilon$.

## 3.4. Stain Separation Layer: Deconvolución de Color

El procedimiento de IHC, en el contexto de la clasificación de la sobreexpresión de la proteína HER2, permite resaltar estructuras celulares relevantes para la asignación del *score*. En presencia de sobreexpresión, el biomarcador DAB produce una tinción café oscuro de gran intensidad en la membrana celular. En contraste, la Hematoxilina produce una tinción azul de núcleo celular, y la eosina una tinción magenta. Esto se puede apreciar en la Figura 15. La cantidad de tinción depositada provee un indicador de la distribución de la proteína HER2, o de las estructuras a la cual se fija la tinción. Es evidente que cuantificar esta intensidad de tinción proporciona información de gran valor para métodos automáticos de clasificación.

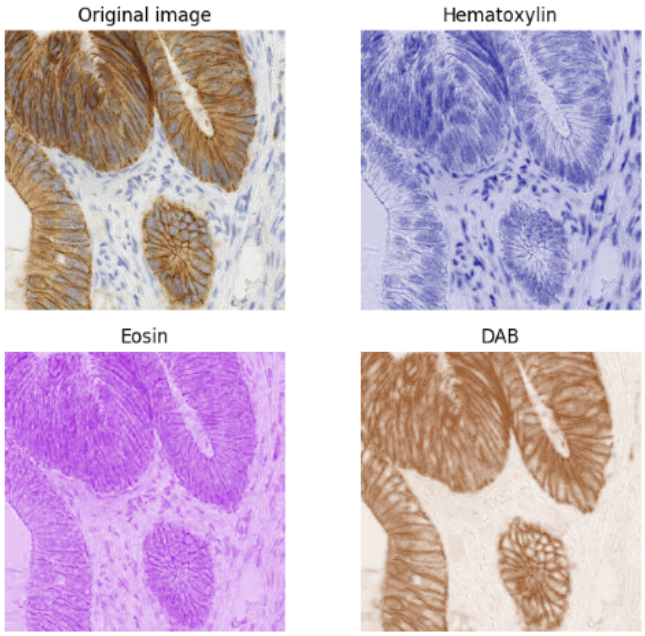



Figura 15: Descomposición de colores HED - la imagen original se puede descomponer en un nuevo espacio de colores HED (*Hematoxilina-Eosina-DAB*). Cada componente resalta estructuras celulares de interés para el su análisis.
Fuente: scikit-image.org

La idea de extraer características basándose en el análisis de las tinciones resultantes del procedimiento IHC ya ha sido estudiado en trabajos previos. El uso de un kernel de deconvolución de color [Ruifrok y Johnston, 2001] es utilizado para descomponer la imagen IHC en los tres componentes *Hematoxilina*, *Eosina* y *DAB*, o también llamado **espacio de colores HED**. Luego, estos componentes son utilizados para la generación de características secundarias para realizar la identificación de las estructuras celulares. En particular, el componente *DAB* es utilizado para la tarea de segmentación de la membrana celular [Ficarra *et al.*, 2011, Pezoa, 2017].

De lo anterior, se puede inferir que utilizar una imagen en el espacio de colores HED, puede resultar beneficioso para el desempeño de modelo en la clasificación de la sobreexpresión.

En este trabajo, se propone la implementación de la `stain separation layer`, una capa que incorpora la descomposición de los colores como parte de la arquitectura del modelo. `stain separation layer` es una capa convolucional con parámetros no entrenables, predefinidos por el kernel descrito en [Ruifrok y Johnston, 2001], la cual permite transformar una imagen desde el espacio de colores RGB al espacio de colores HED o al **espacio de colores HDX** (sin Eosina).

La Figura 16 ilustra el funcionamiento de esta nueva capa. La transformación se logra con una simple convolución con tres filtros. Esta operación consiste en el producto del filtro con los

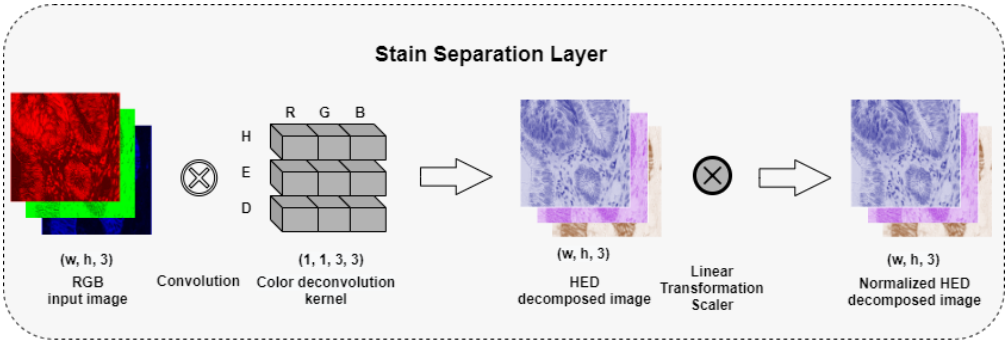



Figura 16: Stain separation layer - La descomposición de los colores se logra con la aplicación de un kernel de deconvolución de color. Luego, el resultado es normalizado.
Fuente: Elaboración propia.

tres componentes RGB de cada píxel de la imagen de entrada. Cada filtro corresponde a uno de los tres componentes de salida (Hematoxilina, Eosina y DAB, por ejemplo). Finalmente, la imagen resultante es normalizada.

## 3.5. Métodos de Agregación

En el flujo de trabajo presentado en la Sección 3.2, en el último paso de *evaluación y estimación de incertidumbre* (ver Figura 12) requiere de un componente que convierta las predicción y estimaciones de incertidumbre individuales de los *patches*, a una sola respuesta de consenso asignada a la muestra del DS. El componente `Aggregator`, es un operador $G$ que cumple con lo siguiente:

$$p_i = p(y|x_i, \mathcal{D}_{\text{train}}) \approx \frac{1}{T}\sum_t p(y = c_k|x_i, \hat{\omega}_t)$$

$$u_i \approx \text{EpistemicUncertainty}(F, x_i)$$

$$\{\bar{p}_u(y|x_1, \ldots, x_m, \mathcal{D}_{\text{train}}), \bar{u}\} = G(p_1, \ldots, p_m; u_1, \ldots, u_m) \tag{3}$$

Donde $x_i$ corresponde a uno de los $m$ *patches* generados para la evaluación (*grid patch generator*), $u_i$ es una medida de incertidumbre epistémica asociada al modelo $F$ y al *patch* $x_i$. $\bar{p}_u$ y $\bar{u}$ son el resultado de aplicar el operador $G$. El primero corresponde a la respuesta consenso de las predicciones individuales, bajo la medida de incertidumbre epistémica $u$. El segundo corresponde a la estimación de la incertidumbre epistémica de la predicción de consenso.

Es preciso señalar la selección de la incertidumbre epistémica sobre la aleatoria. Si bien ambos tipos de incertidumbre pueden llegar a utilizarse como argumento para la agregación, se decide utilizar la incertidumbre epistémica. La decisión se basa en la interpretación de la agregación como una respuesta pesada en la confianza de la predicción individual, la cual es estimada por la incertidumbre epistémica.

El operador `Aggregator` puede ser implementado de diversas maneras que cumpla con estas condiciones. En esta memoria se implementan tres métodos, los cuales son descritos a continuación.

### 3.5.1. Average Aggregator

El método **average aggregator**, también es llamado **base aggregator**, consiste en un promedio de las predicciones individuales.

La estimación de la incertidumbre epistémica $\bar{u}$ se obtiene a partir de la predicción $\bar{p}_u$ en el caso de $\mathbb{H}$. La estimación $\mathbb{I}$ se realiza a partir de las predicciones individuales $\hat{p}_i$.

**Algorithm 4** Agregación de predicciones utilizando el promedio.

**procedure** Average Aggregator($p_1, \ldots, p_m;\ u_1, \ldots, u_m$)
    $\bar{p} \leftarrow$average($\{p_1, \ldots, p_m\}$)
    $\bar{\mathbb{H}} \leftarrow \mathbb{H}(\bar{p})$
    $\bar{\mathbb{I}} \leftarrow \mathbb{I}(\{p_1, \ldots, p_m\})$
    $\bar{u} \leftarrow \{\bar{\mathbb{H}}, \bar{\mathbb{I}}\}$
    **return** $\bar{p}, \bar{u}$

### 3.5.2. Threshold Aggregator

Como se ha discutido anteriormente, el valor de incertidumbre epistémica estimada se encuentra relacionada con la confianza del modelo. De esto se sigue que para el par $\{\hat{p}_i, \hat{u}_i\}$, cuando el valor de $\hat{u}_i$ es alto, se puede interpretar que el modelo no tiene confianza en la predicción $\hat{p}_i$.

El método de **threshold aggregator** se construye a partir de esta idea. Utiliza solo aquellas predicciones con una baja incertidumbre, es decir, un alto grado de confianza.

Requiere de un parámetro $t$, un valor correspondiente al umbral (*threshold*) máximo de incertidumbre. Luego, se obtiene el promedio a partir de las predicciones individuales $\hat{p}_i$ que tengan una incertidumbre $\hat{u}_i$ menor a $t$.

**Algorithm 5** Agregación de predicciones utilizando el promedio con umbral de incertidumbre.

**procedure** Threshold Aggregator($t;\ p_1, \ldots, p_m;\ u_1, \ldots, u_m$)
    $P \leftarrow \{\}$
    **for** $i$ **in** $1, \ldots, m$ **do** ▷ Filtrar predicciones por umbral de incertidumbre.
        **if** $u_i \leq t$ **then**
            $P \leftarrow P \cup p_i$
    $\bar{p} \leftarrow$ average($P$)
    $\bar{\mathbb{H}} \leftarrow \mathbb{H}(\bar{p})$
    $\bar{\mathbb{I}} \leftarrow \mathbb{I}(P)$
    $\bar{u} \leftarrow \{\bar{\mathbb{H}}, \bar{\mathbb{I}}\}$
    **return** $\bar{p}, \bar{u}$

### 3.5.3. Mixture Aggregator

Utilizando la misma interpretación de la incertidumbre que el método anterior, **mixture aggregator** utiliza la incertidumbre $u_i$ para ponderar la influencia de cada predicción $p_i$ en la respuesta de consenso. En este sentido, las predicciones con menor incertidumbre deben tener una mayor influencia en la computación de $\bar{p}$, así, el peso $w_i$ debe ser inversamente proporcional a la incertidumbre.

**Algorithm 6** Agregación de predicciones utilizando el promedio ponderado por incertidumbre.

**procedure** Mixture Aggregator($p_1, \ldots, p_m;\ u_1, \ldots, u_m$)
    $w \leftarrow \{\}$
    **for** $i$ **in** $1, \ldots, m$ **do** ▷ Calcular los pesos de cada predicción.
        $w_i \leftarrow$ exp$(-u_i)$
        $w \leftarrow w \cup w_i$
    $w \leftarrow w/$sum$(w)$
    $\bar{p} \leftarrow$ average$_w(\{p_1, \ldots, p_m\}, w)$
    $\bar{\mathbb{H}} \leftarrow \mathbb{H}(\bar{p})$
    $\bar{\mathbb{I}} \leftarrow \mathbb{I}(\{p_1, \ldots, p_m\})$
    $\bar{u} \leftarrow \{\bar{\mathbb{H}}, \bar{\mathbb{I}}\}$
    **return** $\bar{p}, \bar{u}$

## 3.6. Conjunto de datos

A continuación, se describirán los conjuntos de datos utilizados para validar la solución.

### 3.6.1. Cats vs. Dogs

El conjunto de datos *cats vs. dogs* [Elson *et al.*, 2007] es una colección de imágenes etiquetadas de perros y gatos. Desarrollado por *Petfinder.com* y *Microsoft*, fue usado originalmente para diferenciar usuarios humanos de robot como una prueba CAPTCHA (*Completely Automated Public Turing test to tell Computers and Humans Apart*).

Figura 17: Ejemplos de *Cats vs. Dogs*.
Fuente: tensorflow.org

*cats vs. dogs* se ha convertido en un conjunto de datos estándar de CV, utilizado principalmente para la clasificación de imágenes en una de las dos clases: gato (*class 0*) o perro (*class 1*). El problema de clasificación binaria ya ha sido resuelto por diversas arquitecturas de CNN, sin embargo, el conjunto de datos es utilizado como una prueba estándar para nuevas arquitecturas o métodos de clasificación de imágenes.

El conjunto de datos está formado por $23262$ imágenes. Las clases se encuentran balanceadas (ver Figura 18) con $11658$ imágenes de la clase *Cat* y $11604$ de la clase *Dog*. Con el fin de utilizar estas imágenes como entrada de una CNN, se les implica *padding* y/o reducción de tamaño para que todas las imágenes que componen al conjunto de datos tengan la misma resolución. En consecuencia, las imágenes cuya resolución es inferior a la objetivo, pasan a estar compuesta en mayor parte por píxeles sin información.

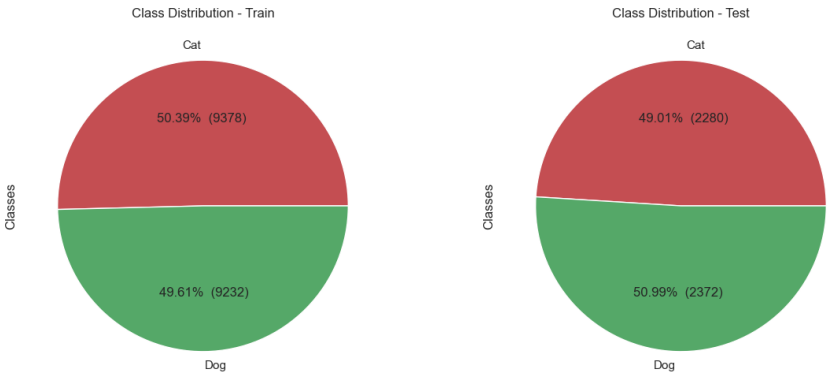



Figura 18: Distribución de clases de *Cats vs. Dogs* - El conjunto de datos fue dividido en 80 % Train y 20 % Test.

Fuente: Elaboración Propia

### 3.6.2. HER2 Scoring in Breast Cancer Histology Images

El concurso *HER2 Challenge Contest* [Qaiser *et al.*, 2018] celebrado el año 2016 por el *Tissue Image Analytics (TIA) Centre*, *Dept of Computer Science*, *University of Warwick*, en colaboración con la *Pathological Society of Great Britain & Ireland*. Su objetivo es acelerar el desarrollo de algoritmos para la automatización de la clasificación de la sobreexpresión HER2, también llamado *HER2 Scoring*, en imágenes WSI con IHC para el diagnóstico del cáncer de mama.

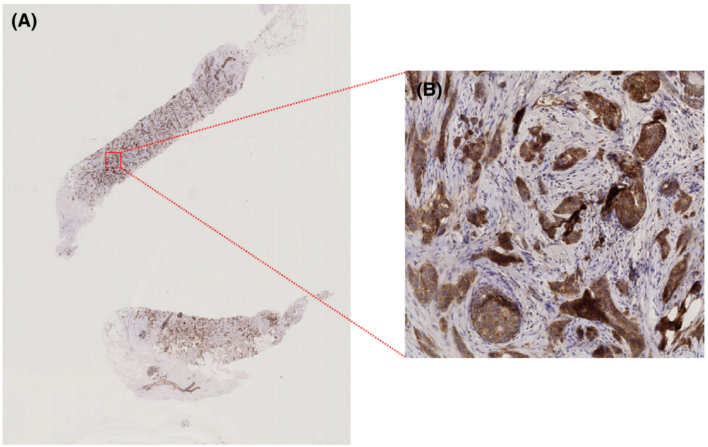



Figura 19: Comparación de tamaño de *digital slide* y *patch* - Las *digital slides* (DS) son imágenes de gigapíxeles de *slides* teñidas con IHC (A). Al extraer pequeños *patches* (B), la resolución más pequeña permite la aplicación de métodos de *deep learning* sin uso de memoria y tiempo de cálculo excesivos.

Fuente: HER2 Challenge Contest [Qaiser *et al.*, 2018]

El conjunto de datos está conformado por una colección de DSs de histología escaneadas en un *Hamamatsu NanoZoomer C9600*. La digitalización de la muestra de histología, permite obtener una imagen con un aumento de x4 a x40 de aumento, proceso comparable a la observación a través microscopio estándar realizado por el patólogo. La Figura 19 es un ejemplo del conjunto de datos. Las DSs son imágenes de gigapíxeles almacenadas en una estructura piramidal de múltiples resoluciones, organizado en niveles (*levels*). La Figura 20 muestra la distribución de resolución de estas imágenes a su máximo aumento *(level 0)*.

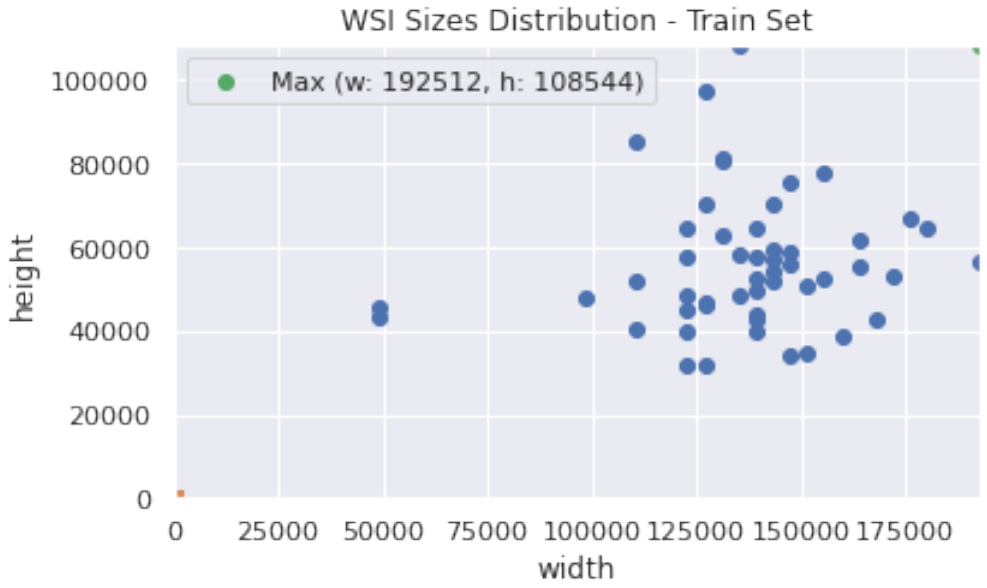



Figura 20: Distribución de tamaños de *digital slides* en el conjunto de entrenamiento.
Fuente: Elaboración propia

El conjunto de datos del concurso incluye 172 imágenes de WSI extraídas de 86 casos de carcinomas de mama invasivos e junto a los portaobjetos teñidos con H&E y HER2. Este conjunto de datos se encuentra dividido en dos subconjuntos: *training* y *test*. Solo el conjunto de datos *training* dispone de los *ground truth* (GT), dado que el acceso a las etiquetas del conjunto de *test* se restringe en pos de una competencia justa. Los valores de GT son tomados de los informes clínicos emitidos sobre los casos de carcinomas de mama invasivos, realizados en un centro de referencia para patología mamaria (Nottingham University Hospitals, NHS Trust).

En esta memoria, sólo se considerará el conjunto *training*, ya que es necesario la disponibilidad del GT para realizar evaluaciones de los algoritmos propuestos, independientemente de terceros agentes.

El conjunto de datos utilizado consta de 52 DS con casos igualmente distribuidos para las cuatro posibles puntuaciones (*score*) de HER2 (0, 1+, 2+ y 3 +). De este conjunto se generan los conjuntos de datos utilizados en esta memoria. Luego de la etapa de *preprocesamiento asistido de digital slide*, expuesto en la Sección 3.2, el conjunto de objetos identificados se di-

vide en *training/test* con una razón de $80/20$ (ver Figura 21), correspondiente a los conjuntos de datos utilizados para el entrenamiento y evaluación de los modelos, respectivamente.

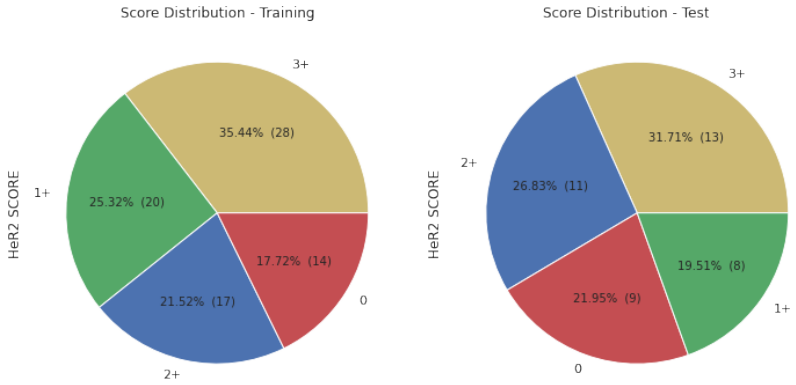



Figura 21: Distribución de clases en el conjunto de entrenamiento y evaluación.
Fuente: Elaboración propia

La evaluación utilizada por *HER2 Challenge Contest* difiere de la evaluación tradicional de los modelos de clasificación. Los tres criterios o métricas definidas se construyen en base a la importancia clínica y las implicaciones del *score* HER2 predicho, en la vida del paciente. En particular, la evaluación propuesta en [Qaiser *et al.*, 2018] considera el impacto de una clasificación errónea, asignando una puntuación según las clases confundidas, como se muestra en la Tabla 6. Dado esto, los tres criterios utilizados son los siguientes:

1. **Puntos de acuerdo (*Agreement Points*)**: Se considera la suma de los puntos obtenidos por cada predicción. Estos valores son definidos en la Tabla 6.

| **Predicted True** | **0** | **1+** | **2+** | **3+** |
|---|---|---|---|---|
| **0** | 15 | 15 | 10 | 0 |
| **1+** | 15 | 15 | 10 | 0 |
| **2+** | 2.5 | 2.5 | 15 | 5 |
| **3+** | 0 | 0 | 10 | 15 |

Tabla 6: Puntos de acuerdo para *score* predichos - Una puntuación de 0/1+ si se interpreta como 3+ o viceversa es un error grave, mientras que una puntuación de 2+ como 0/1+ niega a algunos pacientes un tratamiento válido; una puntuación de 3+ para un caso de 2+ pasa por alto la prueba FISH y puede tratar algunos casos erróneamente (que habrían sido FISH negativos) con medicamentos tóxicos, mientras que una puntuación de 3+ real degradada a 2+ requiere un gasto adicional de la prueba FISH, pero el resultado final probablemente será el mismo y, por lo tanto, esto no debe considerarse un error grave.
Fuente: HER2 Challenge Contest [Qaiser *et al.*, 2018]

2. **Confianza ponderada (*Weighted Confidence*)** $W_c$: Se obtiene como la suma del valor de confianza $c$ obtenido en cada predicción. Se considera de si la predicción es correcta o incorrecta:

$$W_c = \begin{cases} \frac{2c-c^2}{2} & \text{Clasificado correctamente} \\ \frac{1-c^2}{2} & \text{Clasificado incorrectamente} \end{cases} \tag{4}$$

3. **Combinado (*Combined*):** Considera la suma del producto de las dos medidas anteriores para cada predicción.

# CAPÍTULO 4
# VALIDACIÓN DE LA SOLUCIÓN

En este capítulo, se realiza la validación de `her2bdl` es utilizado en dos escenarios distintos.

En el primer escenario, correspondiente al problema de clasificación binaria *cats vs. Dogs*, se demuestra que el *framework* cumple con funcionales de uso general, descritas en la Sección 3.1, permitiendo abordar un problema de clasificación de imágenes, utilizando modelos de *bayesian deep learning*. Junto a esto, el experimento permite realizar un análisis e interpretación de la medición de la incertidumbre, utilizando la implementaciones de modelos propuestos: `MC-dropout model` y *aleatoric model*.

El resto de este capítulo se enfoca en el uso de la herramienta desarrollada en el problema de *HER2 Challenge Contest* de la Universidad de Warwick, demostrando su aplicabilidad para la clasificación de la sobreexpresión de la proteína HER2 en imágenes de alta resolución (WSI).

Junto a los resultados descritos en este capítulo, se encuentran disponibles a través de la plataforma web de *Weight & Biases*, tres reportes interactivos de los experimentos realizados. Se invita al lector a explorar los resultados en estos reportes interactivos utilizando los siguientes enlaces:

- *Binary Classification with MC-dropout Models*

  wandb.ai/sborquez/her2bdl/reports/Binary-Classification-with-MC-Dropout-Models–VmlldzozMTg0OTE

- *Evaluation of the Stain Separation Layer for HER2 Scoring*

  wandb.ai/sborquez/her2bdl/reports/HER2-Scoring-with-Uncertainty-Models–Vmlldzo2ODg2MDM

- *HER2 Scoring with Uncertainty Models*

  wandb.ai/sborquez/her2bdl/reports/HER2-Scoring-with-Uncertainty-Models–Vmlldzo2ODg2MDM

## 4.1. Clasificación Binaria

La medición de la incertidumbre para los modelos de DL puede desempeñar un papel fundamental en el para la gestión de riesgos. Para comprender el mecanismo y la interpretación de la incertidumbre aleatoria y epistémica, en este experimento se analiza los modelo de BDL en un contexto más simple y sin riesgos, la clasificación de imágenes con el conjunto de datos *cats vs. dogs*.

En este escenario de clasificación binaria, cada imagen $x_i$ se debe determinar si pertenece a la clase 0 *Cat* o clase 1 *Dog*. Se utiliza la meta-arquitectura *MC-Dropout model* para medir

la incertidumbre epistémica, definiendo la arquitectura **EfficientNet B0 MC-dropout model**, presentada anteriormente con la Figura 13.

Para este conjunto de datos, es conveniente usar un modelo de clasificación de imágenes entrenado en un conjunto de datos similar. El modelo `Encoder` $E$ es reemplazado por una instancia de *EfficientNet*. Esta arquitectura fue propuesta en [Tan y Le, 2019], conformado por una familia de CNN que alcanza un desempeño del estado del arte en tareas de clasificación de imágenes, con la particularidad de utilizar una arquitectura eficiente con una cantidad reducida de parámetros. Formulado con el objetivo de escalar de manera metódica, ofrece un conjunto de modelos predefinidos, de B0 a B7, ajustable a los recursos computacionales disponibles.

En particular, $E$ es definido como una instancia del modelo **EfficientNet B0**, la arquitectura de menor tamaño. El modelo utiliza una imagen de entrada $x_i$ de tamaño $224 \times 244$, y computa un vector de variables latente $z_i$ de tamaño $62720$. Considerando el dominio de *cats vs. dogs*, imágenes de dos animales, es conveniente reutilizar los pesos entrenados para el conjunto de datos ImageNet [Russakovsky *et al.*, 2015]. En cuanto al modelo de clasificador estocástico $C$, este es definido con una configuración **C876** con un *dropout rate* $D_{\text{rate}} = 0.5$, con dos neuronas de salidas, correspondiente a las clases *Cat* y *Dog*.

Por otro lado, *aleatoric model* se construye a partir de *EfficientNet B0 MC-dropout model*. El modelo resultante,**EfficientNet B0 aleatoric model**, se encuentra en la Figura 14.

Para ambas arquitecturas, en los anexos se encuentran la Tabla 13 con detalles las capas y cantidad de parámetros de cada modelo.

Se entrenan múltiples instancias de ambos modelos, utilizando el algoritmo de optimización RMSprop y un *learning rate* de $0.0001$. Experimentos preliminares mostraron bajo rendimientos en con los algoritmos SDG o basados en momento como ADAM. Reutilizar pesos pre-entrenados permite reducir la cantidad de *epochs* del entrenamiento a un máximo de 10 *epochs*, sin afectar negativamente el desempeño de los modelos.

### 4.1.1. Métricas de Clasificación

La Figura 22 muestra los valores de función de pérdida para los conjuntos de entrenamiento y validación, obtenidos de los modelos epistémicos. En general, las métricas de rendimiento, tanto la matriz de confusión como la curva ROC (ver Figura 23) muestran que este modelo puede clasificar casi perfectamente imágenes de perros y gatos.

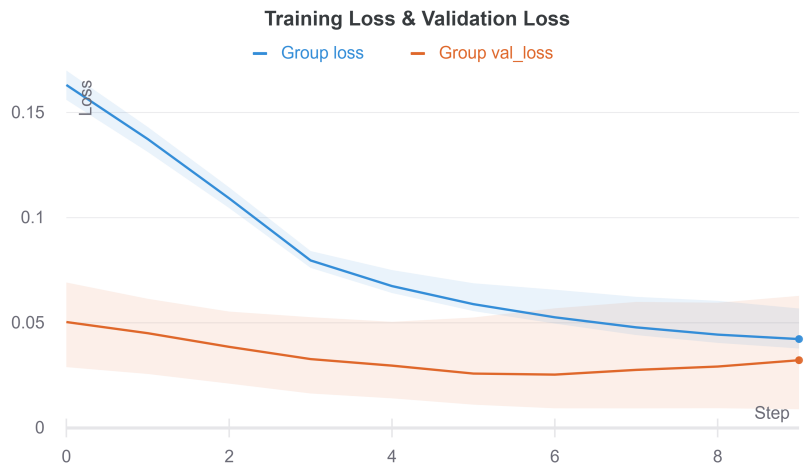



Figura 22: Entrenamiento de modelo epistémico para *cats vs. dogs*
Fuente: Elaboración propia.

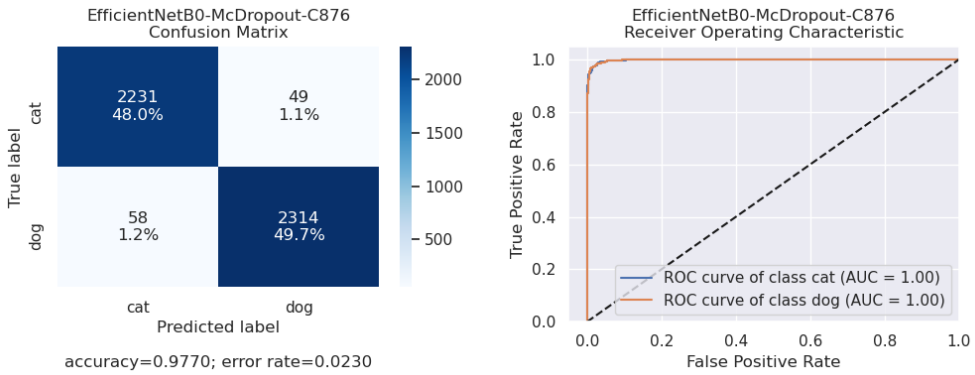



Figura 23: Matriz de Confusión y curva ROC para *cats vs. dogs* - Muestra los resultados del modelo de mejor desempeño, evaluado el conjunto de datos de *test*.
Fuente: Elaboración propia.

### 4.1.2. Incertidumbre Epistémica

Recordando la discusión de la incertidumbre epistémica de la Sección 2.5.1, corresponde a la incertidumbre que representa el desconocimiento del modelo; también se conoce como la incertidumbre del modelo porque se puede interpretar como la confianza en la predicción.

Las dos medidas de incertidumbre epistémica evaluadas son la *predictive entropy* $\mathbb{H}$, la cual captura la incertidumbre en la predicción. La *mutual information* $\mathbb{I}$ captura la confianza del modelo en su predicción.

Se evalúa el mejor modelo obtenido en la fase de entrenamiento en el conjunto de *test*. La Figura 24 muestra la distribución de los valores de incertidumbre estimadas, junto a los ejemplos del conjunto *test* con mayor valor de incertidumbre.

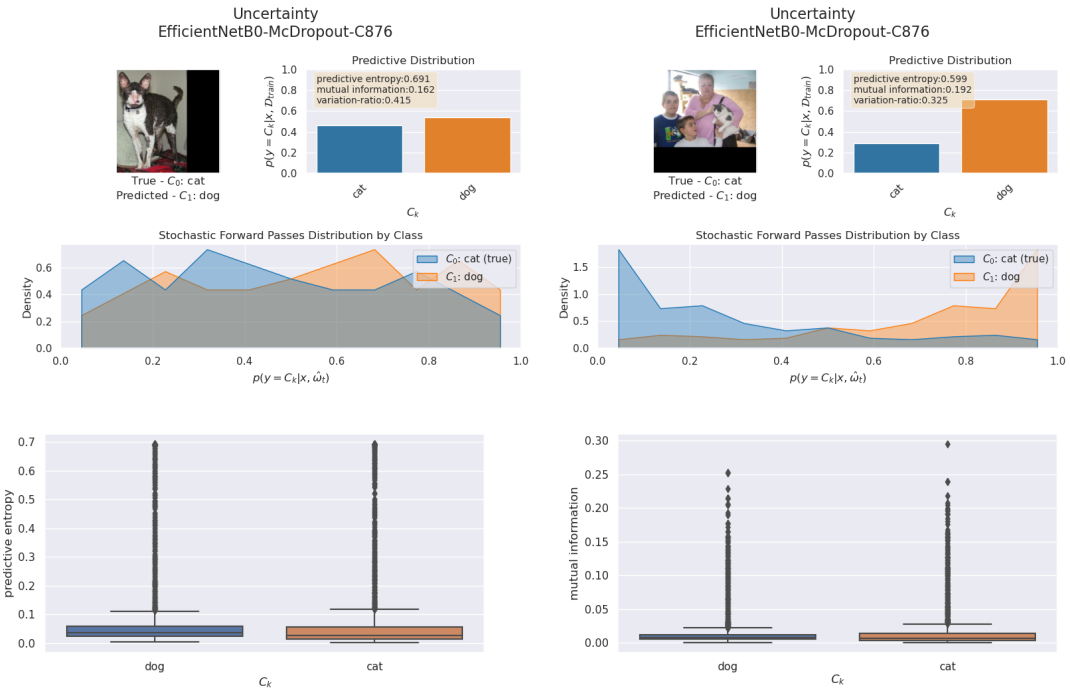



Figura 24: Incertidumbre epistémica en *cats vs. dogs* - Para ambas medidas de incertidumbre, *predictive entropy* izquierda y *mutual information* derecha, la fila superior muestra un ejemplo del conjunto de datos de *test* con una medida de incertidumbre alta. Mientras que la fila inferior, muestra la distribución de la incertidumbre para ambas clases.
Fuente: Elaboración propia.

Los resultados de la incertidumbre epistémica muestran que la media de la *predictive entropy* y la *mutual information*, para el conjunto de evaluación, están cercanas a 0, por lo que, en términos generales, el modelo tiene un alto nivel de confianza y sin sesgo de clases. No obstante, algunos valores atípicos tienen una alto valor incertidumbre epistémica.

La Figura 25 contiene una colección de estas muestras con valores de incertidumbre alto. El modelo tiene una baja confianza con estos ejemplos, la causa de estos valores pueden agruparse en tres categorías. La primera fila muestra que una imagen con una pequeña región de píxeles que contiene información es un rasgo de entradas con incertidumbre.

La segunda fila contiene imágenes fuera de la distribución del conjunto de entrenamiento. El modelo fue entrenado con imágenes de perros y gatos, al predecir imágenes con otro tipo de contenido, el modelo no computará las características que permiten discernir efectivamente entre perros y gatos. Las imágenes de personas o el texto son un problema grave, principalmente cuando éstos errores se presentan en los conjuntos de datos oficiales utilizados por la comunidad.

Finalmente, la tercera fila contiene imágenes de perros y gatos con rasgos que pueden resultar confusos incluso para las personas. Por ejemplo, los animales pueden estar en una pose

Figura 25: Ejemplos con alta incertidumbre epistémica - Se pueden categorizar en tres tipos de ejemplos de alta incertidumbre: En la categoría (primera fila), se encuentran imágenes con poca información. La segunda categoría (segunda fila), corresponde a imágenes que no pertenecen a la distribución del conjunto de datos de entrenamiento. En la tercera categoría (tercera fila), se encuentran aquellas imágenes que presentan características confusas.
Fuente: Elaboración propia.

extraña. También se dan casos como la primera imagen, donde la perspectiva puede resultar engañosa, haciendo ver a un perro mirando a la cámara, cuando en realidad es la parte superior de la cabeza de un gato. O incluso pueden darse ejemplos como un gato y un perro con características de la otra clase, como un perro de orejas puntiagudas y un gato de pelaje similar a la de un pitbull.

Se realizó una encuesta a un grupo de 43 personas con un subconjunto de imágenes evaluadas por el modelo, por cada imagen se les consultaba un valor entre $0$ y $1$, el cual representaba grado de pertenencia a las clases $C_0$: *Cat* y $C_1$: *Dog*, donde un valor cercano a $0$ significa que pertenece a la clase $C_0$, mientras que un cercano valor $1$ corresponde a $C_1$. Como se muestra en la Tabla 7, las personas tienen las mismas dificultades al determinar la clase de las imágenes con poca información o que no pertenecen a conjunto de datos. Sin embargo, son más flexibles para identificar el concepto de gato o perro en una imagen.

| x | EfficientNet | Humanos |
|---|---|---|
| | $p(y = C_1\|x, \mathcal{D}_{\text{train}}) = 0.997$<br>$\mathbb{H} = 0.020$<br>$\mathbb{I} = 0.004$ | $p(y = C_1\|x) = 0.993$<br>$\mathbb{H} = 0.038$<br>$\mathbb{I} = 0.016$ |
| | $p(y = C_1\|x, \mathcal{D}_{\text{train}}) = 0.642$<br>$\mathbb{H} = 0.652$<br>$\mathbb{I} = 0.110$ | $p(y = C_1\|x) = 0.524$<br>$\mathbb{H} = 0.691$<br>$\mathbb{I} = 0.141$ |
| | $p(y = C_1\|x, \mathcal{D}_{\text{train}}) = 0.443$<br>$\mathbb{H} = 0.686$<br>$\mathbb{I} = 0.120$ | $p(y = C_1\|x) = 0.494$<br>$\mathbb{H} = 0.692$<br>$\mathbb{I} = 0.285$ |
| | $p(y = C_1\|x, \mathcal{D}_{\text{train}}) = 0.532$<br>$\mathbb{H} = 0.691$<br>$\mathbb{I} = 0.162$ | $p(y = C_1\|x) = 0.130$<br>$\mathbb{H} = 0.387$<br>$\mathbb{I} = 0.204$ |

Tabla 7: EfficientNet vs. Humanos - La tabla compara las predicciones realizadas por el modelo epistémico EfficientNet frente a las respuestas dadas por un grupo de alrededor de 40 personas. Es preciso señalar que de estos últimos, las respuestas individuales son interpretadas como las $T$ *forward passes* del modelo epistémico.
Fuente: Elaboración propia.

### 4.1.3. Incertidumbre Aleatoria

El otro tipo de incertidumbre es la incertidumbre aleatoria, también conocida como incertidumbre de los datos. Este es un tipo de incertidumbre irreductible y es el ruido inherente a las observaciones. El modelo aleatorio entrenado, es utilizado para computar la **predictive variance** para las imágenes pertenecientes al conjunto *test*.

La Figura 26 muestra un ejemplo del conjunto de evaluación un valor alto de incertidumbre aleatoria. Los resultados muestran además que existe una diferencia en la distribución de la incertidumbre para cada clase. Se observa que la clase *Dog* tiende a tener una mayor incertidumbre que la clase *Cat*.

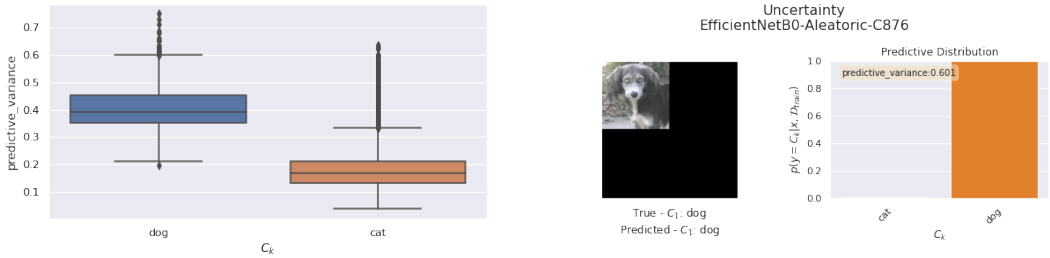



Figura 26: Incertidumbre aleatoria en *cats vs. dogs* - A la izquierda se encuentra la distribución por clases. A la derecha, un ejemplo del conjunto de datos *test* con alta *predictive variance*.
Fuente: Elaboración propia.

Considerando que un valor más alto de varianza significa un “ruido”más alto en una imagen de entrada, las muestras con una gran región negra también pueden ser fuente de incertidumbre. Por otro lado, la diferencia entre clases puede explicarse por la variabilidad de las características de los perros en contraste con la baja variabilidad de las características de los gatos. Las diferencias entre las razas, formas, tamaños y pieles de los perros son visualmente diferentes a las de los gatos. Se podría inferir que la varianza puede verse afectada por características propias de la clase.

Figura 27: Ejemplos con alta incertidumbre aleatoria - Se puede atribuir el origen de la alta *predictive variance* de la clase *Dog* a la diversidad de atributos que caracterizan a la clase, dado por la variedad de razas de perros existentes. En el caso contrario, la uniformidad de los atributos los gatos pueden dar paso a una menor *predictive variance* de la clase *Cat*.
Fuente: Elaboración propia.

## 4.2. Evaluación de *Stain Separation Layer*

En esta sección se evalúa el impacto de la capa `stain separation layer` (descomposición de color) en la tarea de clasificación de *patches*, comparando la capacidad de generalización de los modelos de clasificación de *patches*, utilizando tres espacios de color HED, HDX y RGB. Esta comparación se realiza con una estrategia de *crossvalidation* con *5-folds*.

La arquitectura definida para la clasificación de *patches* utilizada en este experimento, consiste en una instancia de la meta-arquitectura `MC-dropout model` ajustada para la tarea de clasificación de *patches* (ver Figura 28). Para el modelo $E$, se utiliza el modelo `stain separation encoder`, una CNN que utiliza como primera capa una `stain separation layer`, los detalles de este modelo se encuentran en el anexo, en la Tabla 14. El modelo utiliza una imagen de entrada $x_i$ , y computa un vector de variables latente $z_i$ de tamaño $10816$.

Por otro lado, el clasificador $C$ se define a partir de la notación **CXX** con un *dropout rate* $D_{\text{rate}} = 0.2$.

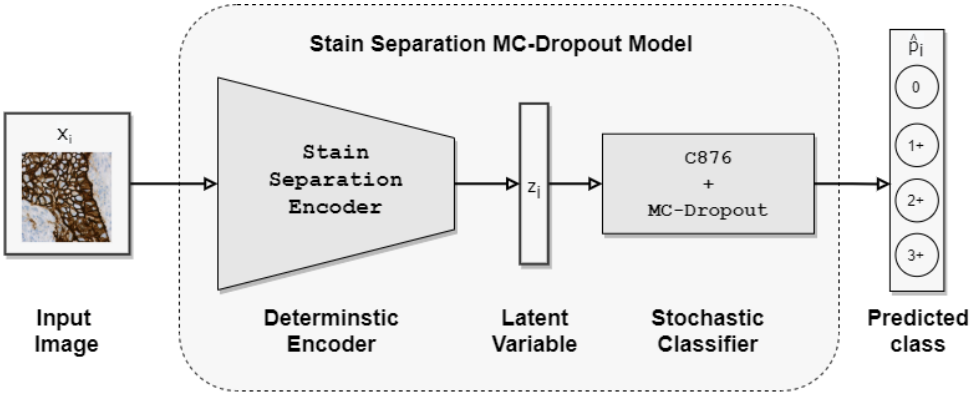



Figura 28: Instancia de modelo stain separation MC-dropout
Fuente: Elaboración propia.

La evaluación de los hiper-parámetros consiste en la búsqueda de dos variables:

- ***stain separation layer***: El espacio de color de la imagen de entrada. Se evalúa el impacto de aplicar la con tres espacios de colores: separación de color con hematoxilina, eosina y DAB (HED); Separación de colores con hematoxilina y DAB (HDX); y sin separación de color (RGB).
- **La configuración del clasificador**: El número de configuraciones posibles es enorme. Con el fin de reducir el espacio de exploración, solo se comparan dos configuraciones: C777 y C876. Esta decisión afecta el rendimiento de los modelos ya que estas configuraciones no se han optimizado para la tarea de clasificación de la sobreexpresión. Sin embargo, el objetivo es realizar la comparación para la descomposición de los colores, por lo que estas variantes son utilizada para contrastar el desempeño, independiente del clasificador utilizado.

Cada modelo se evalúa con una estrategia de *crossvalidation* de 5-*folds*, con divisiones del conjunto de datos que conserven el balance de las clases. Se utiliza el algoritmo de optimización RMSprop y un *learning rate* de $0.0001$, iterando un máximo de $100$ *epochs* pero considerando un *early stop* si el *validation loss* no mejora en 15 *epochs*.

La Figura 29 muestra las métricas de clasificación promedio para cada configuración de modelo en los 5 *folds*. Como era de esperar, los modelos con capas de separación de manchas (HED y HDX) obtienen un mejor desempeño frente a la no aplicación de deconvolución de color (RGB). También se observa que los modelos HDX logran el mayor rendimiento de clasificación.

En cuanto al desempeño de cada modelo, la Figura 30 muestra los resultados desagregados

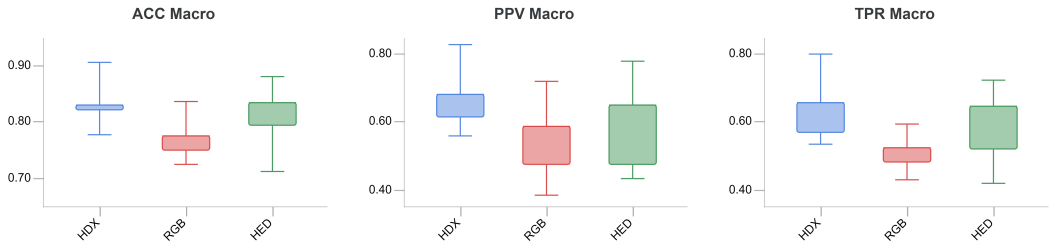



Figura 29: Comparación de desempaño de *stain separation layer* - Los gráficos muestran la distribución del desempaño de aplicando distintas capas de descomposición de color.
Fuente: Elaboración propia.

para cada configuración evaluada. Se puede observar de estos resultados que el modelo **HDX-C876** supera a los otros modelos en el rendimiento general, pero no por un gran margen.

Por el contrario, los resultados de la configuración del clasificador no difieren mucho entre los dos (C77 y C876). Sin embargo, se utilizará la configuración del modelo HDX-C876 para los siguientes experimentos, dado que esta configuración obtiene los valores más altos de desempeño.

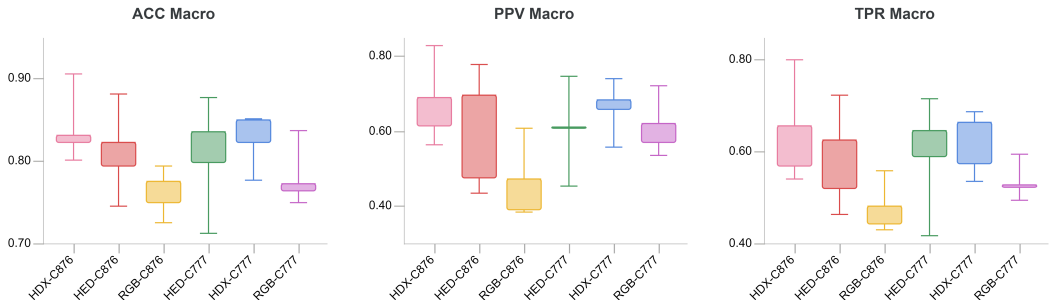



Figura 30: Medidas de desempeño para *crossvalidation* - Los gráficos muestran la distribución del desempaño de cada configuración. Los modelos son evaluados en 5 *folds*.
Fuente: Elaboración propia.

## 4.3. Evaluación de Clasificación de *Patches*

En este experimento, se realiza una evaluación del desempeño de los modelos para la clasificación de la sobreexpresión de HER2 a nivel de *patches*. Además, se presenta un análisis de la estimación de la incertidumbre epistémica y aleatoria para el contexto del HER2 *Scoring*.

Con el objetivo de analizar el impacto de hiper-parámetro *dropout rate* $D_{\text{rate}}$, el cual condiciona la distribución de los pesos aprendidos por el modelo, se definen tres nuevas variantes a partir de la mejor configuración obtenida en el experimento anterior, correspondiente a tres valores distintos para $D_{\text{rate}}$ : $0.2$, $0.5$ y $0.8$.

Múltiples instancias de cada modelo se entrenan utilizando la división *train/validation* del

conjunto de entrenamiento. Se utiliza el algoritmo de optimización RMSprop y un *learning rate* de $0.0001$, iterando un máximo de $200$ *epochs* pero considerando un *early stop* si el *validation loss* no mejora en 15 *epochs*.

### 4.3.1. Métricas de Clasificación

La evolución de los modelos entrenados se resume en la Figura 31. Se puede observar como la variación de $D_{\text{rate}}$ afecta al proceso de entrenamiento. Valores altos de $D_{\text{rate}}$ requerirán más *epochs* para poder converger. Esta diferencia también se manifiesta en la evolución de la incertidumbre epistémica evaluada en el conjunto de entrenamiento, donde se destaca la misma tendencia para los distintos valores de $D_{\text{rate}}$. En lo que respecta a las métricas de clasificación, no se observa alguna diferencia que se pueda correlacionar con los valores de $D_{\text{rate}}$ de cada configuración.

Para cada valor de $D_{\text{rate}}$, el modelo epistémico con menor *validation loss* es evaluado en el conjunto *test*. De esta se obtiene el desempeño de clasificación y las medidas de incertidumbre epistémica analizadas a continuación.

| **Model** | $D_{\text{rate}}$ | **ACC Macro** | **PPV Macro** | **TPR Macro** |
|---|---|---|---|---|
| HDX-C876-0.2 | 0.2 | **0.803** | **0.641** | **0.553** |
| HDX-C876-0.5 | 0.5 | 0.789 | 0.534 | 0.522 |
| HDX-C876-0.8 | 0.8 | 0.788 | 0.533 | 0.518 |

Tabla 8: Evaluación de modelos clasificadores de *patches*.
Fuente: Elaboración propia.

La Tabla 8 resume las métricas de clasificación para cada modelo, agregadas respecto a las 4 clases. Se observa que las métricas no son del todo satisfactorias, en especial la *precision* y *recall* presentan valores cercanos al $0.5$. A pesar de ello, el modelo con $D_{\text{rate}} = 0.2$ obtiene los mejores valores, pero no por un gran margen.

Por otro lado, la Figura 32 y Tabla 9 muestran el comportamiento por clase. Se puede observar un comportamiento similar para cada uno de los tres modelos. El rendimiento por de la clase 3+ presenta un valor de *recall* (TPR) más alto frente a las demás clases, lo que significa que es más fácil identificar cuando está presente la sobreexpresión de la proteína HER2.

Por un lado, también se observa que la clase 0 tiene el peor desempeño en cuanto a *recall*. Esto es un comportamiento esperado, los *patches* que pertenecen a la clase 0 no suelen tener características relevantes para determinar su clasificación, es más, la falta de estos determina su clase. La clase 0 se caracteriza por no presentar tinción de la membrana celular. Esto es un problema al clasificar a nivel de *patch*, debido a que la regiones con falta de tinción puede encontrarse en los *patches* pertenecientes a las otras tres clases.

Este problema también afecta de manera inversa a las demás clases, la falta de característi-

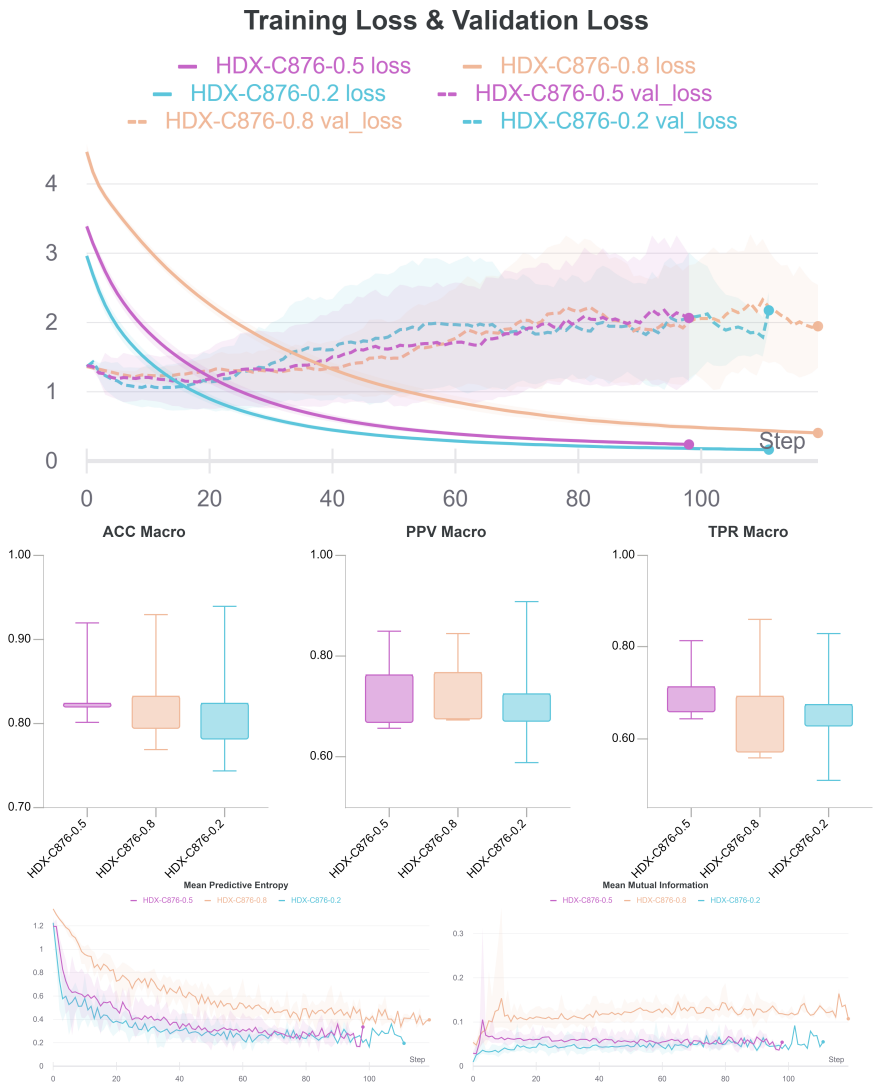



Figura 31: Entrenamiento de modelos epistémicos para clasificación de *patches*.
Fuente: Elaboración propia.

| **Model** | $D_{\text{rate}}$ | **ACC** (0/1+/2+/3+) | | | | **PPV** (0/1+/2+/3+) | | | | **TPR** (0/1+2+/3+) | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| HDX-C876-0.2 | 0.2 | 0.832 | 0.760 | 0.761 | 0.859 | 0.889 | 0.465 | 0.449 | 0.760 | 0.205 | 0.494 | 0.616 | 0.896 |
| HDX-C876-0.5 | 0.5 | 0.804 | 0.756 | 0.742 | 0.853 | 0.536 | 0.433 | 0.419 | 0.749 | 0.300 | 0.298 | 0.596 | 0.896 |
| HDX-C876-0.8 | 0.8 | 0.806 | 0.758 | 0.740 | 0.846 | 0.546 | 0.437 | 0.416 | 0.735 | 0.288 | 0.292 | 0.589 | 0.903 |

Tabla 9: Evaluación de modelos clasificadores de *patches* por clase.
Fuente: Elaboración propia.

cas relevantes en un *patch* afecta directamente a la *precision* (PPV) de estas tres clases, por ejemplo, al clasificar erróneamente a la clase 0 como 3+.

Se puede observar además que en la matriz de confusión, los modelos entrenados tienden a clasificar erróneamente las clases 2+, sobreestiman a la clases 1+ como clase 2+, o clasifican erróneamente a la clase 2+ como 3+. Esto es una consecuencia directa de la definición de

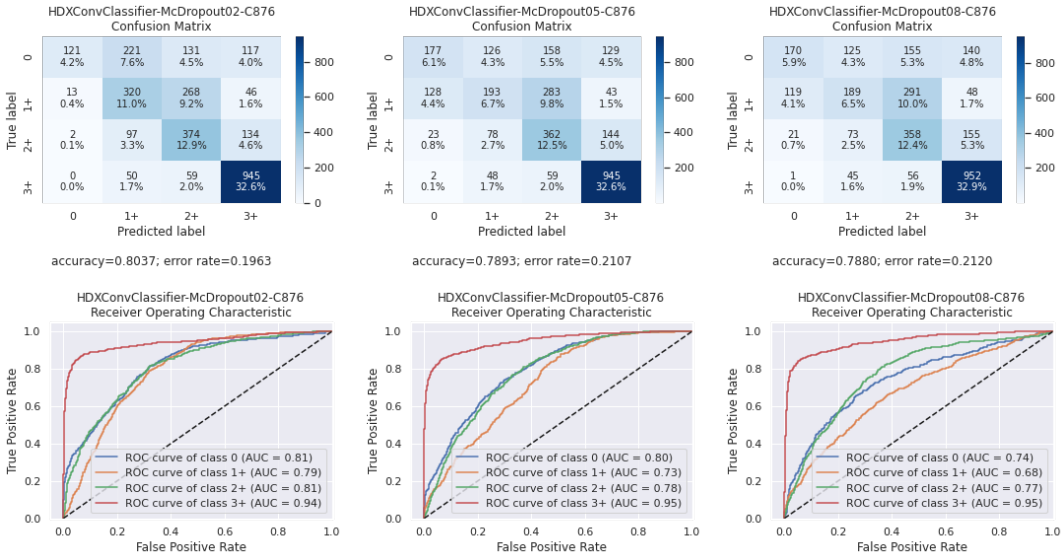



Figura 32: Evaluación de modelos clasificadores de *patches* por clase.
Fuente: Elaboración propia.

la clase 2+. La clase 2+ representa una clasificación equívoca: las muestras que caen en la clasificación clase 2+ puede contener, por ejemplo, membranas intensamente teñidas como la clase 3+, pero con menos del 10 % de todo el tejido. Como el modelo no puede acceder a la imagen completa, este no tiene suficiente información para realizar una correcta predicción del *patch*.

A pesar de estos resultados, se espera que al evaluar la muestra de tejido completa, las predicciones individuales erróneas sean irrelevantes al agregar la predicción final.

### 4.3.2. Incertidumbre Epistémica

Los modelos epistémicos son evaluados con el conjunto de datos *test*, la Figura 33 muestra la distribución de la incertidumbre epistémica, *predictive entropy* y *mutual information*, para todos los *patches* generados para este conjunto de datos. Existe una clara diferencia en cuanto a la incertidumbre por clase, este comportamiento se repite tanto para los distintos modelos, como para las dos medidas de incertidumbre. En estos gráficos se observa como la clase con una menor incertidumbre corresponde a la clase 3+. Cabe mencionar que existen una gran cantidad de *outliers* correspondientes a esta clase.

También se puede observar una tendencia de la media de las medidas de incertidumbre. A medida que el *score* baja de categoría, el valor de la medio de la incertidumbre aumenta. Este comportamiento se sobrepone a lo discutido en la sección anterior, donde la clase 0 es la categoría más confundida por los modelos.

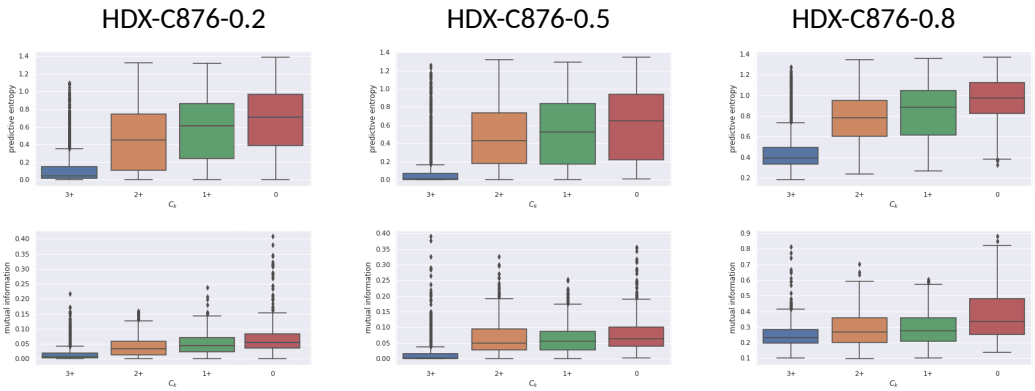



Figura 33: Incertidumbre epistémica en clasificación de *patches* - Distribución de incertidumbre, *predictive entropy* (arriba) y *mutual information* (abajo).
Fuente: Elaboración propia.

| **Model** | $D_{\text{rate}}$ | **Mean** $\mathbb{H}$ | **Mean** $\mathbb{I}$ |
|---|---|---|---|
| HDX-C876-0.2 | 0.2 | 0.418 | 0.039 |
| HDX-C876-0.5 | 0.5 | 0.391 | 0.052 |
| HDX-C876-0.8 | 0.8 | 0.712 | 0.292 |

Tabla 10: Medias de incertidumbre epistémica para modelos de clasificación de *patches*.
Fuente: Elaboración propia.

Una segunda observación surge de comparar entre modelos. Como se detalla la Tabla 10, a mayor valor de $D_{\text{rate}}$, mayor es la media de los valores de incertidumbre epistémica.

La Figura 34 muestra algunos ejemplos de predicciones. En la primera columna se encuentran ejemplos con una alta incertidumbre. Estos valores se pueden atribuir a la falta de información relevante en el *patch*. En particular, la primera imagen (A) pertenece a la clase 3+, pero el *patch* no posee atributos que puedan correlacionarse con esta clase. Por otro lado, tanto las imágenes (C) y (E), ejemplifican la respuesta del modelo ante entradas con poca información. En estos casos, el modelo tiende a distribuir las probabilidad de las cuatro clases de manera uniforme.

La segunda columna de la Figura 34 destaca algunos casos particulares, en la imagen (B) se encuentra un *patch* clasificado correctamente como clase 0, pero el resultado que se ve opacado por un valor alto de incertidumbre. En cuanto a las imágenes (D) y (F), ambas presentan una baja estimación de incertidumbre, no obstante, esto no es indicador de una buena clasificación, ya que esta última imagen (F) a pesar de la alta confianza del modelo, se encuentra erróneamente clasificada.

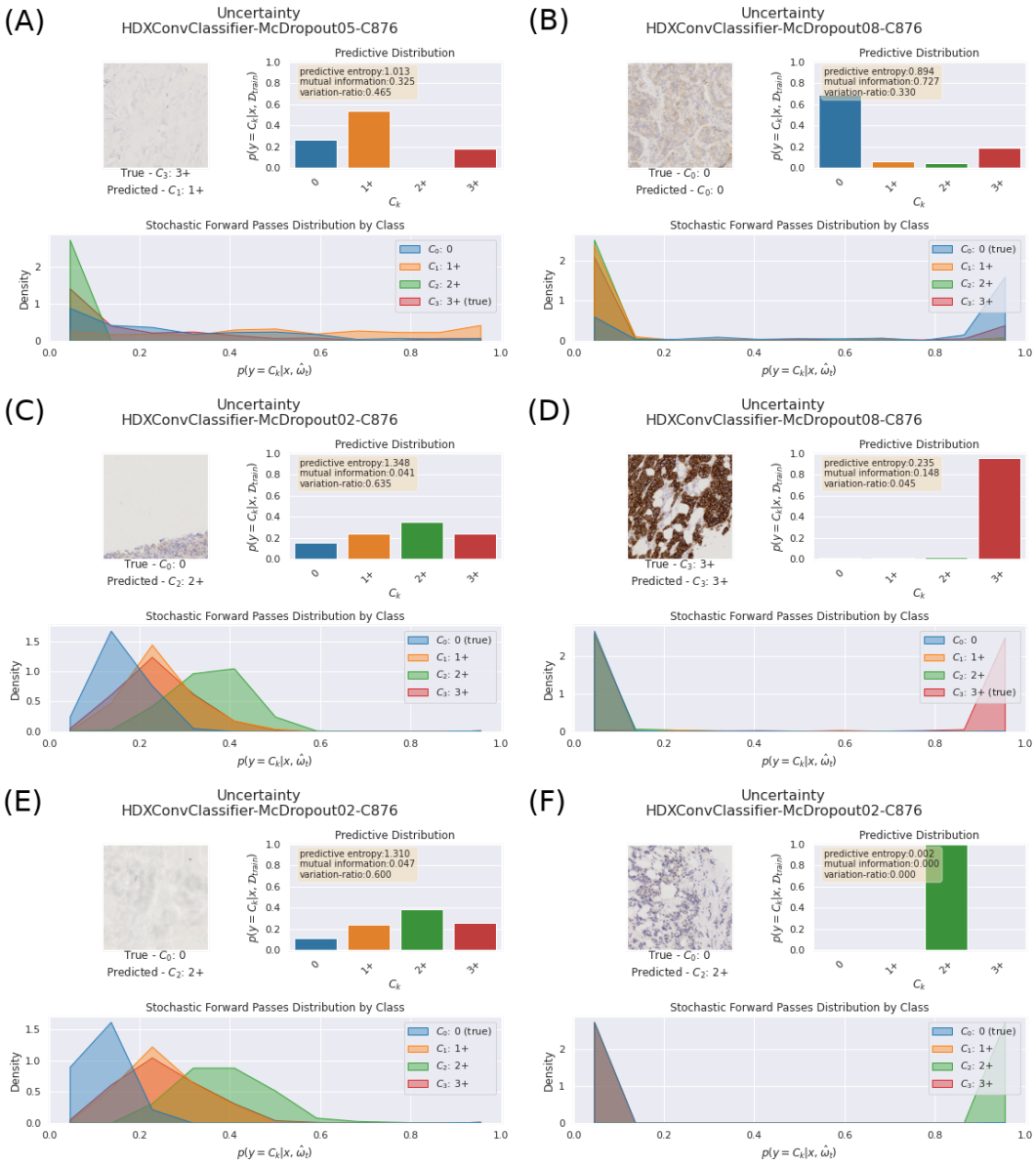



Figura 34: Predicciones destacadas de incertidumbre epistémica de *patches*.
Fuente: Elaboración propia.

### 4.3.3. Incertidumbre Aleatoria

Para la estimación de la incertidumbre aleatoria en el contexto de la clasificación de *patches*, se construye el modelo `stain separation aleatoric` (ver Figura 35), definido a partir del modelo `stain separation MC-dropout`.

El modelo es entrenado con los mismos parámetros que la versión epistémica, y se evalúa en el conjunto de datos *test*.

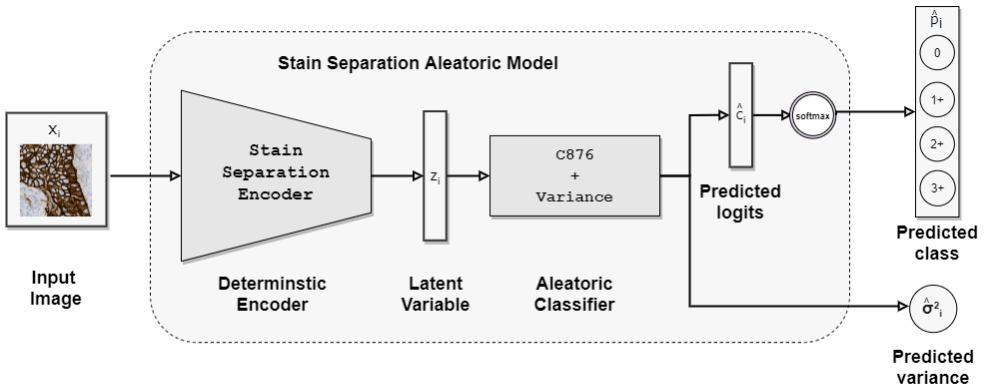



Figura 35: Instancia de modelo stain separation aleatoric
Fuente: Elaboración propia.

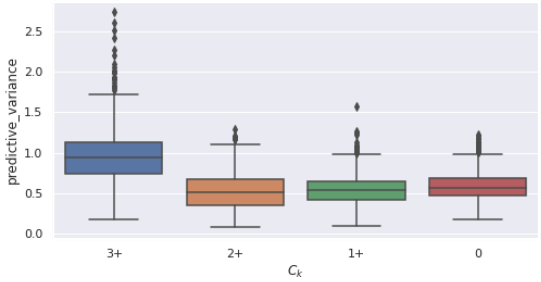



Figura 36: Distribución de incertidumbre aleatoria en clasificación de *patch* por clase.
Fuente: Elaboración propia.

Se observa en la Figura 36 que, a diferencia de la incertidumbre epistémica, los resultados muestran un valor de la media de incertidumbre aleatoria mayor para la clase 3+. Como se ejemplifica en la Figura 37 (fila superior), las imágenes perteneciente a la clase 3+ se caracterizan especialmente por la intensidad de la tinción café DAB. Sin embargo, dada la complejidad de las estructuras celulares de los tejidos, la tinción de las membranas celulares pueden generar una gran diversas de formas, texturas e intensidades de tinción.

Por otro lado, se observa en la Figura 37 imágenes de la baja varianza de la clase 2+. Los *patches* que pertenecen a esta clase suelen ser presentar características similares entre ellas, debido a la uniformidad de los atributos que caracterizan a la clase 2+

Los resultados obtenidos permiten reafirmar la interpretación del experimento con el problema *cats vs. dogs*. Se puede establecer una relación entre la alta varianza de una predicción con la diversidad de atributos que caracterizan a la clase. Lo que se refleja en la diferencia de las varianzas entre las clases.

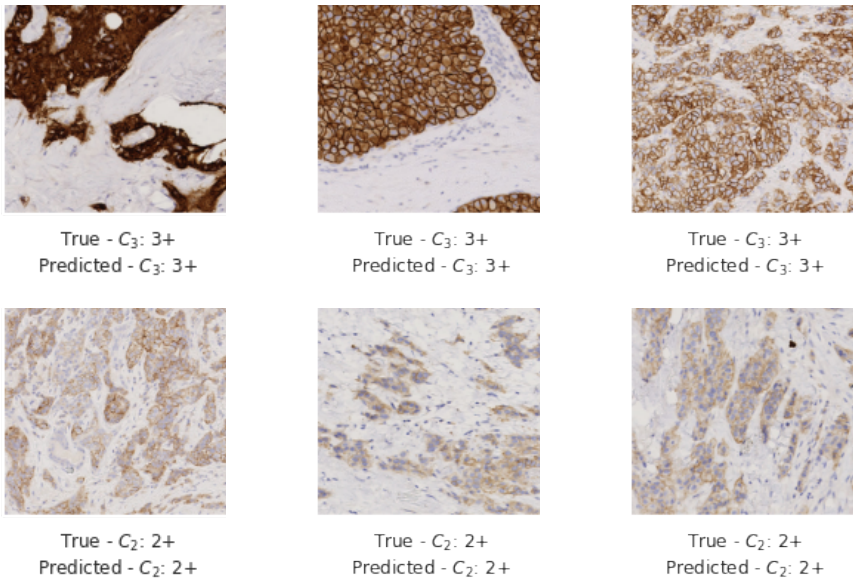



Figura 37: Ejemplos destacados de aleatoria en clasificación de *patch* - La primera fila muestra ejemplos de la clase 3+ con alta varianza, mientras que la segunda fila muestra ejemplos de la clase 2+ con una baja varianza.
Fuente: Elaboración propia.

## 4.4. Evaluación de la Sobreexpresión

Finalmente, en esta sección se analizan la predicción de la clasificación de la sobreexpresión a nivel de muestra.

Utilizando los tres modelos de mejor desempeño obtenidos en la Sección 4.3, se obtiene la predicción de clase y estimación de incertidumbre a nivel de *patch*, de cada uno de los objetos que conforman el conjunto de datos *test*. Estas predicciones son utilizadas como entradas para los distintos métodos agregación, definidos en la Sección 3.5.

Se determinan distintas configuraciones para los métodos de agregación, variando el criterio de medida de incertidumbre. Estas configuraciones se resumen en la Tabla 11.

| **Method** | $t$ | $u$ | **Legend** |
|---|---|---|---|
| Base | | | B |
| Threshold | 0.25 | $\mathbb{I}$ | Th_I |
| Threshold | 0.1 | $\mathbb{I}$ | Tl_I |
| Mixture | | $\mathbb{I}$ | M_I |
| Threshold | 0.5 | $\mathbb{H}$ | Tl_H |
| Threshold | 0.69 | $\mathbb{H}$ | Th_H |
| Mixture | | $\mathbb{H}$ | M_H |

Tabla 11: Configuraciones de métodos de agregación.
Fuente: Elaboración propia.

### 4.4.1. Criterios de Evaluación

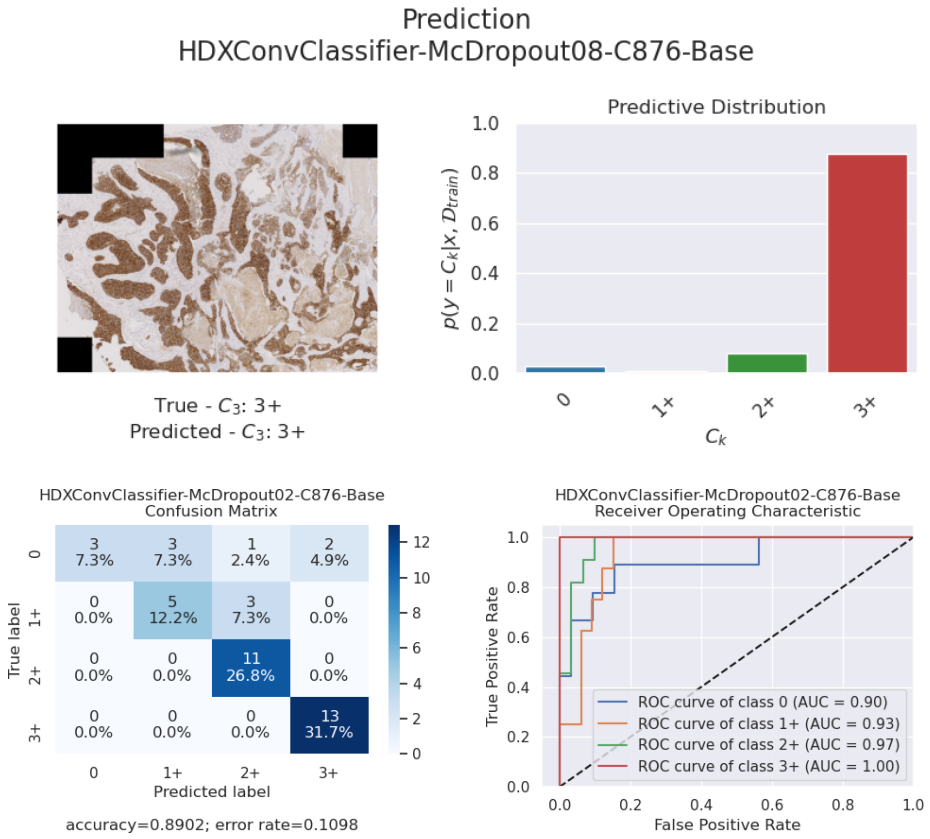



Figura 38: Evaluación de método de agregación - La fila superior muestra un ejemplo de predicción de una muestra. En la fila inferior, se muestra la matriz de confusión y curva ROC del método de agregación de mejor desempeño.
Fuente: Elaboración propia.

Las predicciones resultantes de cada método son evaluados utilizando las métricas de clasificación y los criterios de evaluación propuestos en *HER2 Challenge Contest*, presentados en la Sección 3.6.2. Los resultados de esta evaluación se encuentran en la Tabla 12, estos se encuentran ordenados por el criterio *Combined*. Como se puede observar, no existe una mayor diferencia entre los distintos métodos de agregación utilizados.

Al comparar los métodos *base*, se observa que el modelo $D_{\text{rate}} = 0.2$ tienen un mejor desempeño en los seis criterios, frente a los modelos *Base* $D_{\text{rate}} = 0.5$ y $D_{\text{rate}} = 0.8$. La tendencia se repite de manera generalizada para el resto de los métodos de agregación.

En cuanto al desempeño de los métodos de agregación, se pueden llegar a dos conclusiones: A medida que aumenta el valor de $D_{\text{rate}}$, los métodos que consideran la incertidumbre comienzan a dominar sobre el método *Base*, debido a que este modelo obtuvo un mejor desempeño a nivel de *patch*, y que el parámetro se encuentra se encuentra positivamente correlacionado con la incertidumbre de los modelos, al aumentar $D_{\text{rate}}$, las predicciones

individuales pueden alcanzar mayor valor de incertidumbre, lo que en consecuencia le dan una mayor influencia durante la agregación.

En segundo lugar, los métodos que consideran a la *mutual information* $\mathbb{I}$, obtienen mejores resultados frente a los métodos que utilizan la *predictive entropy* $\mathbb{H}$.

| Legend | $D_{\text{rate}}$ | Method | $u$ | $t$ | Mean $\mathbb{H}$ | Mean $\mathbb{I}$ | Combined | $W_c$ | Points | ACC Macro | PPV Macro | TPR Macro |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| B-0.2 | 0.2 | Base | | | 0.775 | 0.356 | **246.36** | 17.438 | **565.0** | **0.89** | **0.806** | **0.74** |
| Th_I-0.2 | 0.2 | Threshold | $\mathbb{I}$ | 0.25 | 0.777 | 0.359 | 245.818 | 17.401 | 565.0 | 0.89 | 0.806 | 0.74 |
| M_I-0.2 | 0.2 | Mixture | $\mathbb{I}$ | | 0.77 | 0.358 | 244.947 | 17.349 | 565.0 | 0.878 | 0.789 | 0.712 |
| Tl_I-0.2 | 0.2 | Threshold | $\mathbb{I}$ | 0.1 | 0.749 | 0.363 | 244.757 | 17.364 | 565.0 | 0.866 | 0.775 | 0.684 |
| Th_H-0.2 | 0.2 | Threshold | $\mathbb{H}$ | 0.69 | 0.649 | 0.394 | 243.107 | **17.628** | 560.0 | 0.89 | 0.805 | 0.736 |
| M_H-0.2 | 0.2 | Mixture | $\mathbb{H}$ | | 0.711 | 0.374 | 242.646 | 17.406 | 560.0 | 0.878 | 0.781 | 0.708 |
| Tl_H-0.2 | 0.2 | Threshold | $\mathbb{H}$ | 0.5 | 0.58 | 0.4 | 241.311 | 17.471 | 560.0 | 0.878 | 0.781 | 0.708 |
| | | | | | | | | | | | | |
| Tl_I-0.5 | 0.5 | Threshold | $\mathbb{I}$ | 0.1 | 0.708 | 0.369 | 232.215 | **17.217** | **535.0** | 0.841 | 0.733 | 0.649 |
| Th_I-0.5 | 0.5 | Threshold | $\mathbb{I}$ | 0.25 | 0.746 | 0.355 | **231.378** | 16.985 | 535.0 | **0.854** | 0.755 | **0.676** |
| M_I-0.5 | 0.5 | Mixture | $\mathbb{I}$ | | 0.742 | 0.356 | 231.374 | 16.993 | 535.0 | 0.854 | 0.755 | 0.676 |
| B-0.5 | 0.5 | Base | | | 0.749 | 0.354 | 231.302 | 16.971 | 535.0 | 0.854 | 0.755 | 0.676 |
| Tl_H-0.5 | 0.5 | Threshold | $\mathbb{H}$ | 0.5 | 0.493 | 0.335 | 229.285 | 17.201 | 525.0 | 0.854 | **0.759** | 0.676 |
| M_H-0.5 | 0.5 | Mixture | $\mathbb{H}$ | | 0.67 | 0.362 | 228.53 | 17.112 | 525.0 | 0.854 | 0.759 | 0.676 |
| Th_H-0.5 | 0.5 | Threshold | $\mathbb{H}$ | 0.69 | 0.583 | 0.36 | 226.518 | 17.01 | 525.0 | 0.854 | 0.759 | 0.676 |
| | | | | | | | | | | | | |
| M_I-0.8 | 0.8 | Mixture | $\mathbb{I}$ | | 0.903 | 0.2 | **219.347** | **15.725** | 525.0 | **0.829** | **0.714** | **0.626** |
| Th_I-0.8 | 0.8 | Threshold | $\mathbb{I}$ | 0.25 | 0.887 | 0.194 | 214.036 | 15.169 | **540.0** | 0.805 | 0.681 | 0.57 |
| B-0.8 | 0.8 | Base | | | 0.91 | 0.201 | 212.705 | 15.371 | 515.0 | 0.817 | 0.694 | 0.603 |
| M_H-0.8 | 0.8 | Mixture | $\mathbb{H}$ | | 0.838 | 0.202 | 209.741 | 15.349 | 510.0 | 0.805 | 0.682 | 0.575 |
| Tl_I-0.8 | 0.8 | Threshold | $\mathbb{I}$ | 0.1 | 0.777 | 0.163 | 209.17 | 14.623 | 537.5 | 0.817 | 0.489 | 0.588 |
| Tl_H-0.8 | 0.8 | Threshold | $\mathbb{H}$ | 0.5 | 0.554 | 0.145 | 204.921 | 14.582 | 522.5 | 0.793 | 0.544 | 0.548 |
| Th_H-0.8 | 0.8 | Threshold | $\mathbb{H}$ | 0.69 | 0.583 | 0.163 | 192.476 | 14.308 | 497.5 | 0.768 | - | 0.497 |

Tabla 12: Resultados de evaluación a nivel de muestra.

La Figura 38 muestra la matriz de confusión y curva ROC del método *B-0.2*. En general, todos los métodos obtuvieron resultados similares, como se puede ver en las métricas de clasificación de la Tabla 12. Al inspeccionar esta matriz de confusión, se observa el mismo comportamiento obtenido durante la clasificación de *patches*. La clase que se confunde más es la clase 0. Sin embargo, se obtiene una mejora en la clasificación de las clases 2+ y 3+. Es necesario mayor investigación para minimizar el error de clasificación de las clases negativas 0 y 1+.

### 4.4.2. Visualización de Mapas de Incertidumbre

Es posible construir mapas de clasificación e incertidumbre al proyectar en las muestras las predicciones de clase y estimaciones de incertidumbre epistémica de los modelos (ver Figura 39).

Estos mapas permiten visualizar cómo el modelo determina la clasificación de la muestra a partir de las predicciones individuales, entregando cierto grado de interpretabilidad a los resultados. Como por ejemplo las filas tres y cuatro de la Figura 39, considerando que las categorías se diferencian por el porcentajes de células con sobreexpresión sobre el total de la muestra, visualizar la proporción de *patches* predichos como clase 2+ vs. clase 3+, permite estimar un porcentaje a partir del mapa resultante.

Junto a esto, la visualización de los mapas de incertidumbre permite localizar las regiones de la muestra con mayor grado de incertidumbre. De esta forma es posible identificar patrones en los tejidos y asociarlos con la incertidumbre que obtienen los modelos.

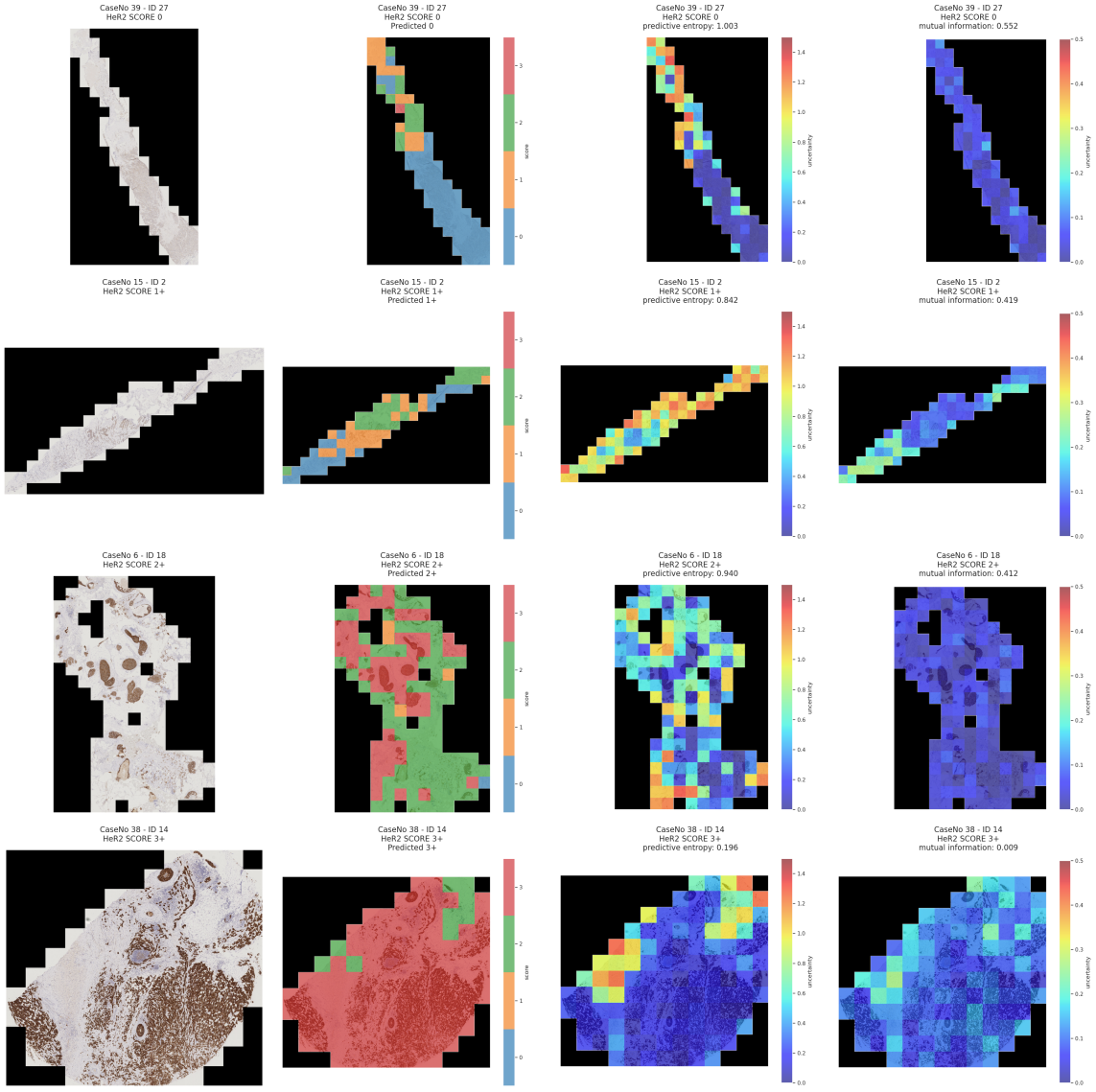



Figura 39: Predicciones agregadas destacadas.
Fuente: Elaboración propia.

Esta información puede dar paso a la construcción de nuevos algoritmos para la generación de *patches* que consideren la incertidumbre en la selección de regiones de interés, de esta forma, generar conjuntos de datos que permitan el entrenamiento de modelos con imágenes con información relevante que permita al modelo generar respuestas con un alto grado de confianza.

# CAPÍTULO 5
# CONCLUSIONES

En este trabajo, se abordó el problema de la clasificación de la sobreexpresión de la proteína HER2, también conocido como HER2 *scoring*, la cual es una tarea crucial realizada por patólogos, para el diagnóstico del cáncer de mamas. Con el objetivo de asistir a los patólogos en el diagnóstico del cáncer de mamas, se desarrolló una herramienta basada en algoritmos automáticos de clasificación con enfoque en la estimación de la incertidumbre, para una evaluación cuantitativa de la sobreexpresión.

Esto se logró con el diseño e implementación de un *framework* para el análisis de muestras de histopatología. El cual establece un flujo de trabajo para el procesamiento de imágenes de alta resolución (*digital slides*): detección de regiones de interés, generación de datos, entrenamiento y evaluación de modelos de clasificación. De esta forma, agilizar el desarrollo de nuevas soluciones para el problema de HER2 *scoring*.

Junto a esto, se implementan los modelos incertidumbre *stain separation MC-dropout* y *stain separation aleatoric* basados en el enfoque *bayesian deep learning*. Los cuales son modelos de clasificación de imágenes, especializados para imágenes de tejidos preparados para el diagnóstico de HER2 *scoring* con el procedimiento de IHC.

Lo anterior permite la evaluación de la incertidumbre asociada a los modelos de clasificación, la incertidumbre epistémica, y la incertidumbre asociada a los datos, la incertidumbre aleatoria. Dando paso a un análisis cuantitativo basado en las predicciones y estimación de confianza realizadas por los modelos.

En el desarrollo de esta memoria, se pudo establecer una interpretación de las medidas de incertidumbre mediante un caso de clasificación binaria. Lo que permitió la formulación de estrategias de agregación de predicciones individuales. Por otro lado, se corroboró el impacto de la deconvolución de color en el desempeño de la tarea de HER2 *scoring*, con el uso de la capa *stain separation layer*.

Respecto a los resultados obtenidos en la tarea de HER2 *scoring*, los modelos y métodos de agregación evaluados obtuvieron resultados satisfactorios. Sin mayores variaciones entre los diferentes métodos evaluados. Siendo la clase 0 la más confundida, y en segundo lugar la 1+, mientras que las clases 2+ y 3+ suelen ser clasificadas correctamente. Estos resultados se pueden correlacionar con las medidas de incertidumbre epistémica obtenidas para las cuatro clases.

Junto a esto, se elaboraron mapas de clasificación e incertidumbre, permitiendo visualizar como el tejido en la *digital slide* es clasificado por los algoritmos de *deep learning*, permitiendo localizar y caracterizar regiones de mayor incertidumbre en los tejidos. Dado cierto grado de interpretabilidad a los resultados obtenidos.

Las contribuciones de este trabajo son las siguientes:

El diseño e implementación de `her2bdl`, un *framework* para el entrenamiento y evaluación de modelos la clasificación de imágenes y estimación de incertidumbre, especializado, pero sin limitarse al problema de *HER2 scoring*. El cual facilita el desarrollo, entrenamiento y evaluación de nuevas soluciones, además de integrar herramientas para la colaboración, replicación los experimentos y divulgación de resultados.

El diseño de las meta-arquitecturas basada en submodelos *MC-Dropout model* y *aleatoric model*, para simplificar el diseño de nuevas arquitecturas bayesianas basadas en *Monte-Carlo dropout* y *predictive variance*, permitiendo el uso arquitecturas conocidas de *deep learning*. Junto a esto, la propuesta del algoritmo *2-steps predictive distribution*, para realizar una predicción eficiente de la *predictive distribution*.

Finalmente, la implementación de la capa *stain separation layer*, una capa de parámetros no entrenables, incorporando la descomposición de los colores (DAB, Hematoxilina y Eosin) como parte de la arquitectura, dando paso a la implementación de modelos especializados en la clasificación de *digital slides*: *stain separation MC-dropout* y *stain separation aleatoric*.

## 5.1. Discusión

A medida que la AI se integra cada vez más en la sociedad, las decisiones tomadas por estos sistemas pueden afectar la vida de los humanos, y en el peor de los casos, volverse una amenaza. La estimación de la incertidumbre puede jugar un rol clave en este aspecto. El uso de modelos probabilísticos que representan la incertidumbre, como los expuestos en esta memoria, permiten cierto grado interpretación a los resultados del modelo. Los expertos pueden utilizar esta medición de la incertidumbre para fundamentar sus decisiones, y a través de trabajo conjunto entre humanos y máquinas, mitigar el riesgo de desplegar estos algoritmos en el mundo real.

El problema abordado en esta memoria, el diagnóstico de la sobreexpresión de la proteína HER2, es uno de estos escenarios, donde los errores cometidos por el algoritmo afectan a la salud del paciente. Por lo que se beneficia de la inclusión de una medida de incertidumbre en la elaboración de sistemas automáticos de clasificación. Los resultados logrados pueden extenderse a una variedad de problemas de clasificación, por ejemplo, la identificación de signos en el diagnóstico de la retinopatía diabética en imágenes oculares. En especial, donde exista un riesgo asociado a la clasificación tomada por el algoritmo.

## 5.2. Trabajo Futuro

El trabajo realizado en esta memoria da paso una diversas aristas para la mejora de la solución propuesta, o para un siguiente paso de investigación, entre estas se destacan las siguientes:

1. Enviar evaluación a *HER2 Challenge Contest*: Si bien el envío de los resultados se encontraba fuera del alcance de esta memoria, los resultados obtenidos pueden ser enviados a los organizadores de *HER2 Challenge Constest* para su evaluación post-concurso.

2. Mejorar la selección de patches: Durante el entrenamiento de los modelos, la generación de *patches* resultó en una serie de imágenes con poca información relevante para la determinación de la clase. Esto afecta negativamente el entrenamiento de los modelos. Se deben proponer soluciones que apunten a una mejor selección de *patches*, considerando los resultados de la agregación. Un enfoque posible es utilizar la incertidumbre como un factor para la selección de *patches*.

3. Mejorar la localización y detección de objetos: Durante el procesamiento asistido de las *digital slides*, el algoritmo usado requiere de la intervención del experto para seleccionar los objetos localizados válidos. La propuestas puede ser mejorada, por ejemplo, con el uso de algoritmos basados en *machine learning*.

4. Explorar nuevas funciones de pérdida: En particular, la evaluación de HER2 *Challenge Contest* propone un sistema de puntaje basado en el impacto de la clasificación errónea, las clases confundidas y la confianza estimada. Dado que el objetivo del concurso se mide con estas métricas, utilizar una función de pérdida basado en los criterios de evaluación de *HER2 Challenge Constest*, puede dar lugar a modelos con un mejor desempeño.

5. Abordar las tareas de clasificación binaria y segmentación de la membrana con BDL: En la literatura, el HER2 *scoring* también es abordado desde el problema de clasificación binaria (positivo/negativo), y desde la segmentación de la membrana con sobreexpresión. Futuras investigaciones pueden tomar el *framework* diseñado y extenderlo para analizar la incertidumbre y su impacto en el diagnóstico de HER2, desde la perspectiva de estas dos tareas.

# ANEXOS

| Layer | Type | Shape | Param # |
|---|---|---|---|
| Encoder | EfficientNet B0 | (None, 62720) | 4049571 |
| Classifier | C876 + mc-dropout | (None, 2) | 16099650 |

| Layer | Type | Shape | Param # |
|---|---|---|---|
| Encoder | EfficientNet B0 | (None, 62720) | 4049571 |
| Aleatoric_classifier | c876 + variance | (None, 3) | 16099715 |

Tabla 13: Arquitecturas EfficientNetB0 - MC-Dropout Model (izquierda) y Aleatoric Model (derecha).

| Layer | Type | Shape | Param # |
|---|---|---|---|
| Encoder | Stain Separation Encoder | (None, 10816) | 303352 |
| Classifier | C876 + mc-dropout | (None, 4) | 2812356 |

| Layer | Type | Shape | Param # |
|---|---|---|---|
| Encoder | Stain Separation Encoder | (None, 10816) | 303352 |
| Aleatoric_classifier | c876 + variance | (None, 5) | 2812421 |

Tabla 14: Arquitecturas Stain Separation - MC-Dropout Model (izquierda) y Aleatoric Model (derecha).

| Layer | Type | Shape | Param # |
|---|---|---|---|
| input_1 | InputLayer | [(None, 240, 240, 3)] | 0 |
| stain_separator | Stain_Separation_Layer | (None, 240, 240, 3) | 0 |
| scaler | Lambda | (None, 240, 240, 3) | 0 |
| stain_depthwiseConv2D | DepthwiseConv2D | (None, 118, 118, 24) | 600 |
| depthwiseConv2D_batchnorm | BatchNormalization | (None, 118, 118, 24) | 96 |
| depthwiseConv2D_activation | Activation | (None, 118, 118, 24) | 0 |
| block0_conv2d_a | Conv2D | (None, 118, 118, 32) | 6912 |
| block0_batchnorm_a | BatchNormalization | (None, 118, 118, 32) | 128 |
| block0_activation_a | Activation | (None, 118, 118, 32) | 0 |
| block0_conv2d_b | Conv2D | (None, 116, 116, 32) | 9216 |
| block0_batchnorm_b | BatchNormalization | (None, 116, 116, 32) | 128 |
| block0_activation_b | Activation | (None, 116, 116, 32) | 0 |
| block0_maxpool | MaxPooling2D | (None, 58, 58, 32) | 0 |
| block1_conv2d_a | Conv2D | (None, 58, 58, 64) | 18432 |
| block1_batchnorm_a | BatchNormalization | (None, 58, 58, 64) | 256 |
| block1_activation_a | Activation | (None, 58, 58, 64) | 0 |
| block1_conv2d_b | Conv2D | (None, 56, 56, 64) | 36864 |
| block1_batchnorm_b | BatchNormalization | (None, 56, 56, 64) | 256 |
| block1_activation_b | Activation | (None, 56, 56, 64) | 0 |
| block1_maxpool | MaxPooling2D | (None, 28, 28, 64) | 0 |
| block2_conv2d_a | Conv2D | (None, 28, 28, 128) | 73728 |
| block2_batchnorm_a | BatchNormalization | (None, 28, 28, 128) | 512 |
| block2_activation_a | Activation | (None, 28, 28, 128) | 0 |
| block2_conv2d_b | Conv2D | (None, 26, 26, 128) | 147456 |
| block2_batchnorm_b | BatchNormalization | (None, 26, 26, 128) | 512 |
| block2_activation_b | Activation | (None, 26, 26, 128) | 0 |
| block2_maxpool | MaxPooling2D | (None, 13, 13, 128) | 0 |
| feature_reduction_conv2d_k1 | Conv2D | (None, 13, 13, 64) | 8256 |
| feature_reduction_activation | Activation | (None, 13, 13, 64) | 0 |
| head_flatten | Flatten | (None, 10816) | 0 |

Tabla 15: Arquitecturas Stain Separation Encoder Model.

| Layer | Type | Shape | Param # |
|---|---|---|---|
| input_2 | InputLayer | [(None, 10816)] | 0 |
| head_dense_1 | Dense | (None, 256) | 2769152 |
| head_batchnorm_1 | BatchNormalization | (None, 256) | 1024 |
| head_activation_1 | Activation | (None, 256) | 0 |
| head_mc_dropout_1 | Dropout | (None, 256) | 0 |
| head_dense_2 | Dense | (None, 128) | 32896 |
| head_batchnorm_2 | BatchNormalization | (None, 128) | 512 |
| head_activation_2 | Activation | (None, 128) | 0 |
| head_mc_dropout_2 | Dropout | (None, 128) | 0 |
| head_dense_3 | Dense | (None, 64) | 8256 |
| head_batchnorm_3 | BatchNormalization | (None, 64) | 256 |
| head_activation_3 | Activation | (None, 64) | 0 |
| head_mc_dropout_3 | Dropout | (None, 64) | 0 |
| head_classifier | Dense | (None, 4) | 260 |

| Layer | Type | Shape | Param # |
|---|---|---|---|
| input_2 | InputLayer | [(None, 10816)] | 0 |
| head_dense_1 | Dense | (None, 256) | 2769152 |
| head_batchnorm_1 | BatchNormalization | (None, 256) | 1024 |
| head_activation_1 | Activation | (None, 256) | 0 |
| head_dense_2 | Dense | (None, 128) | 32896 |
| head_batchnorm_2 | BatchNormalization | (None, 128) | 512 |
| head_activation_2 | Activation | (None, 128) | 0 |
| head_dense_3 | Dense | (None, 64) | 8256 |
| head_batchnorm_3 | BatchNormalization | (None, 64) | 256 |
| head_activation_3 | Activation | (None, 64) | 0 |
| head_logits_classifier | Dense | (None, 4) | 260 |
| variance | Dense | (None, 1) | 65 |
| logits_variance | Concatenate | (None, 5) | 0 |

Tabla 16: Arquitecturas C876 - MC-Dropout Model (izquierda) y Variance Model (derecha).

| CaseNo_Object | y_true | y_pred | $\mathbb{H}$ | $\mathbb{I}$ |
|---|---|---|---|---|
| 15_2 | 1 | 1 | 0.968 | 0.405 |
| 15_3 | 1 | 1 | 0.963 | 0.344 |
| 19_2 | 3 | 3 | 0.05 | 0.011 |
| 19_3 | 3 | 3 | 0.149 | 0.036 |
| 19_4 | 3 | 3 | 0.19 | 0.066 |
| 19_5 | 3 | 3 | 0.28 | 0.117 |
| 19_6 | 3 | 3 | 0.255 | 0.102 |
| 1_1 | 0 | 3 | 1.258 | 0.754 |
| 1_3 | 0 | 1 | 0.863 | 0.56 |
| 24_1 | 1 | 1 | 0.856 | 0.428 |
| 24_3 | 1 | 2 | 0.832 | 0.382 |
| 24_6 | 1 | 1 | 0.756 | 0.232 |
| 25_1 | 2 | 2 | 0.831 | 0.356 |
| 25_3 | 2 | 2 | 0.8 | 0.354 |
| 35_2 | 3 | 3 | 0.408 | 0.168 |
| 35_4 | 3 | 3 | 0.237 | 0.123 |
| 35_5 | 3 | 3 | 0.092 | 0.018 |
| 38_14 | 3 | 3 | 0.394 | 0.244 |
| 38_2 | 3 | 3 | 1.063 | 0.738 |
| 39_27 | 0 | 0 | 1.159 | 0.639 |
| 39_30 | 0 | 0 | 1.021 | 0.486 |
| 39_4 | 0 | 1 | 1.049 | 0.412 |
| 47_1 | 1 | 1 | 1.159 | 0.492 |
| 47_4 | 1 | 2 | 1.091 | 0.524 |
| 49_1 | 2 | 2 | 0.704 | 0.428 |
| 49_10 | 2 | 2 | 0.933 | 0.358 |
| 49_3 | 2 | 2 | 1.091 | 0.521 |
| 49_8 | 2 | 2 | 0.74 | 0.375 |
| 49_9 | 2 | 2 | 0.847 | 0.405 |
| 55_1 | 2 | 2 | 1.081 | 0.631 |
| 55_4 | 2 | 2 | 1.079 | 0.5 |
| 6_11 | 2 | 2 | 0.916 | 0.324 |
| 6_18 | 2 | 2 | 0.939 | 0.412 |
| 70_5 | 0 | 0 | 1.335 | 0.579 |
| 70_7 | 0 | 1 | 1.257 | 0.564 |
| 73_1 | 0 | 2 | 1.165 | 0.314 |
| 73_8 | 0 | 3 | 1.297 | 0.443 |
| 88_1 | 1 | 2 | 1.119 | 0.517 |
| 9_20 | 3 | 3 | 0.362 | 0.18 |
| 9_5 | 3 | 3 | 0.052 | 0.012 |
| 9_6 | 3 | 3 | 0.128 | 0.022 |

Tabla 17: Resultados de evaluación de conjunto de datos *test*.

# REFERENCIAS BIBLIOGRÁFICAS

[Abadi *et al.*, 2015] Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., y Zheng, X. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

[BBC, 2015] BBC (2015). Google pide perdón por confundir a una pareja negra con gorilas. `https://shorturl.at/mopsV` - Visitado 31-07-2020.

[Biewald, 2020] Biewald, L. (2020). Experiment tracking with weights and biases. Software available from wandb.com.

[Bishop, 2006] Bishop, C. M. (2006). *Pattern recognition and machine learning*. springer.

[Chollet, 2015] Chollet, F. (2015). Keras.

[Combalia y Vilaplana Besler, 2018] Combalia, M. y Vilaplana Besler, V. (2018). Monte-carlo sampling applied to multiple instance learning for whole slide image classification. En *International conference on Medical Imaging with Deep Learning: Amsterdam, 4-6th July 2018*, pp. 1–3.

[Cregger *et al.*, 2006] Cregger, M., Berger, A. J., y Rimm, D. L. (2006). Immunohistochemistry and quantitative analysis of protein expression. *Archives of pathology & laboratory medicine*, 130(7):1026–1030.

[Der Kiureghian y Ditlevsen, 2009] Der Kiureghian, A. y Ditlevsen, O. (2009). Aleatory or epistemic? does it matter? *Structural safety*, 31(2):105–112.

[DICOM, 2020] DICOM (2020). *DICOM Whole Slide Imaging (WSI)*. `http://dicom.nema.org/Dicom/DICOMWSI` - Visitado 18-02-2020.

[Elson *et al.*, 2007] Elson, J., Douceur, J. J., Howell, J., y Saul, J. (2007). Asirra: A captcha that exploits interest-aligned manual image categorization. En *Proceedings of 14th ACM Conference on Computer and Communications Security (CCS)*. Association for Computing Machinery, Inc.

[Eriksson y Hu, 2018] Eriksson, D. y Hu, F. (2018). *Whole-slide image preprocessing in Python*. `https://developer.ibm.com/articles/an-automatic-method-to-identify-tissues-from-big-whole-slide-images-pt1/` - Visitado 18-02-2021.

[Farahani *et al.*, 2015] Farahani, N., Parwani, A. V., y Pantanowitz, L. (2015). Whole slide imaging in pathology: advantages, limitations, and emerging perspectives. *Pathol Lab Med Int*, 7(23-33):4321.

[Fawcett, 2006] Fawcett, T. (2006). An introduction to roc analysis. *Pattern recognition letters*, 27(8):861–874.

[Ficarra *et al.*, 2011] Ficarra, E., Di Cataldo, S., Acquaviva, A., y Macii, E. (2011). Automated segmentation of cells with ihc membrane staining. *IEEE Transactions on Biomedical Engineering*, 58(5):1421–1429.

[Gal, 2016] Gal, Y. (2016). Uncertainty in deep learning. *University of Cambridge*, 1(3).

[Gal y Ghahramani, 2016] Gal, Y. y Ghahramani, Z. (2016). Dropout as a bayesian approximation: Representing model uncertainty in deep learning. En *international conference on machine learning*, pp. 1050–1059. PMLR.

[Gonzalez y Woods, 2002] Gonzalez, R. y Woods, R. (2002). Digital image processing.

[Goode *et al.*, 2013] Goode, A., Gilbert, B., Harkes, J., Jukic, D., y Satyanarayanan, M. (2013). Openslide: A vendor-neutral software foundation for digital pathology. *Journal of pathology informatics*, 4.

[Goodfellow *et al.*, 2016] Goodfellow, I., Bengio, Y., y Courville, A. (2016). *Deep Learning*. MIT Press. `http://www.deeplearningbook.org`.

[Gutierrez y Schiff, 2011] Gutierrez, C. y Schiff, R. (2011). Her2: biology, detection, and clinical implications. *Archives of pathology & laboratory medicine*, 135(1):55–62.

[Higham y Higham, 2019] Higham, C. F. y Higham, D. J. (2019). Deep learning: An introduction for applied mathematicians. *SIAM Review*, 61(4):860–891.

[Jordan *et al.*, 1999] Jordan, M. I., Ghahramani, Z., Jaakkola, T. S., y Saul, L. K. (1999). An introduction to variational methods for graphical models. *Machine learning*, 37(2):183–233.

[Kendall y Gal, 2017] Kendall, A. y Gal, Y. (2017). What uncertainties do we need in bayesian deep learning for computer vision? En *Advances in neural information processing systems*, pp. 5574–5584.

[Kumar *et al.*, 2017] Kumar, N., Verma, R., Sharma, S., Bhargava, S., Vahadane, A., y Sethi, A. (2017). A dataset and a technique for generalized nuclear segmentation for computational pathology. *IEEE transactions on medical imaging*, 36(7):1550–1560.

[Lagree *et al.*, 2021] Lagree, A., Mohebpour, M., Meti, N., Saednia, K., Lu, F.-I., Slodkowska, E., Gandhi, S., Rakovitch, E., Shenfield, A., y Sadeghi-Naini, A. (2021). A review and comparison of breast tumor cell nuclei segmentation performances using deep convolutional neural networks. *Scientific Reports*, 11(1):1–11.

[LeCun *et al.*, 2010] LeCun, Y., Kavukcuoglu, K., y Farabet, C. (2010). Convolutional networks and applications in vision. En *Proceedings of 2010 IEEE international symposium on circuits and systems*, pp. 253–256. IEEE.

[Moasser, 2007] Moasser, M. M. (2007). The oncogene her2: its signaling and transforming functions and its role in human cancer pathogenesis. *Oncogene*, 26(45):6469–6487.

[Perez *et al.*, 2006] Perez, E. A., Suman, V. J., Davidson, N. E., Martino, S., Kaufman, P. A., Lingle, W. L., Flynn, P. J., Ingle, J. N., Visscher, D., y Jenkins, R. B. (2006). Her2 testing by local, central, and reference laboratories in specimens from the north central cancer treatment group n9831 intergroup adjuvant trial. *Journal of Clinical Oncology*, 24(19):3032–3038.

[Pezoa, 2017] Pezoa, R. (2017). A computational method for segmentation of histopathological images with her2 overexpression. *Universidad Técnica Federico Santa María*, 1:5–17.

[Qaiser *et al.*, 2018] Qaiser, T., Mukherjee, A., Reddy Pb, C., Munugoti, S. D., Tallam, V., Pitkaaho, T., Lehtimaki, T., Naughton, T., Berseth, M., y Pedraza, A. (2018). Her2 challenge contest: A detailed assessment of automated her2 scoring algorithms in whole slide images of breast cancer tissues. *Histopathology*, 72(2):227–238.

[Rączkowski *et al.*, 2019] Rączkowski, Ł., Możejko, M., Zambonelli, J., y Szczurek, E. (2019). Ara: accurate, reliable and active histopathological image classification framework with bayesian deep learning. *Scientific reports*, 9(1):1–12.

[Rajpurkar *et al.*, 2017] Rajpurkar, P., Irvin, J., Zhu, K., Yang, B., Mehta, H., Duan, T., Ding, D., Bagul, A., Langlotz, C., y Shpanskaya, K. (2017). Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning. *arXiv preprint arXiv:1711.05225*.

[Rakha *et al.*, 2015] Rakha, E., Pinder, S., Bartlett, J., Ibrahim, M., Starczynski, J., Carder, P., Provenzano, E., Hanby, A., Hales, S., y Lee, A. (2015). Updated uk recommendations for her2 assessment in breast cancer. *Journal of clinical pathology*, 68(2):93–99.

[Ruifrok y Johnston, 2001] Ruifrok, A. y Johnston, D. (2001). Quantification of histochemical staining by color deconvolution. *Analytical and quantitative cytology and histology*, 23(4):291–299.

[Russakovsky *et al.*, 2015] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., y Fei-Fei, L. (2015). ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252.

[Saha y Chakraborty, 2018] Saha, M. y Chakraborty, C. (2018). Her2net: A deep framework for semantic segmentation and classification of cell membranes and nuclei in breast cancer evaluation. *IEEE Transactions on Image Processing*, 27(5):2189–2200.

[Sokolova y Lapalme, 2009] Sokolova, M. y Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. *Information processing & management*, 45(4):427–437.

[Srivastava *et al.*, 2014] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., y Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958.

[Sun *et al.*, 2019] Sun, S., Cao, Z., Zhu, H., y Zhao, J. (2019). A survey of optimization methods from a machine learning perspective. *IEEE transactions on cybernetics*, 50(8):3668–3681.

[Sánchez *et al.*, 2018] Sánchez, C., Domínguez, F., Galindo, H., Camus, M., Oddo, D., Villarroel, A., Razmilic, D., Navarro, M. E., Pérez, A., y Medina, L. (2018). Survival of patients with advanced her2+ breast cancer. analysis of a cancer center database. *Revista medica de Chile*, 146(10):1095–1101.

[Tan y Le, 2019] Tan, M. y Le, Q. (2019). Efficientnet: Rethinking model scaling for convolutional neural networks. En *International Conference on Machine Learning*, pp. 6105–6114. PMLR.

[Wang y Xu, 2019] Wang, J. y Xu, B. (2019). Targeted therapeutic options and future perspectives for her2-positive breast cancer. *Signal transduction and targeted therapy*, 4(1):1–22.

[Weiss *et al.*, 2016] Weiss, K., Khoshgoftaar, T. M., y Wang, D. (2016). A survey of transfer learning. *Journal of Big data*, 3(1):1–40.

[Wolff *et al.*, 2018] Wolff, A. C., Hammond, M. E. H., Allison, K. H., Harvey, B. E., Mangu, P. B., Bartlett, J. M., Bilous, M., Ellis, I. O., Fitzgibbons, P., Hanna, W., Jenkins, R. B., Press, M. F., Spears, P. A., Vance, G. H., Viale, G., McShane, L. M., y Dowsett, M. (2018). Human epidermal growth factor receptor 2 testing in breast cancer: American society of clinical oncology/college of american pathologists clinical practice guideline focused update. *Journal of Clinical Oncology*, 36(20):2105–2122. PMID: 29846122.