



viu

**Universidad
Internacional
de Valencia**

Graph Neural Networks para la detección, gradación y prognosis de cáncer a partir de imagen histopatológica

Titulación:
Máster Universitario en
Inteligencia Artificial

Curso académico
2022 – 2023

Alumno/a: Armenteros Soto,
David
D.N.I: 77024949L

Director/a de TFM: Colomer
Granero, Adrián

Convocatoria:
Primera

Artificial Intelligence is the New Electricity.

Andrew Ng

Agradecimientos

Agradezco a mi tutor Adrián Colomer Granero por darme la oportunidad de trabajar en este proyecto y por acompañarme durante el desarrollo del mismo con sus consejos y explicaciones.

En especial, a mi familia por su apoyo no solo durante la elaboración de este trabajo sino en toda esta etapa académica y a todas las personas que me han apoyado a lo largo de los años.

Por último, agradecer a aquellos que, de forma directa o indirecta, han sido una fuente de inspiración y motivación.

Índice general

Índice de figuras	III
Índice de tablas	v
Resumen	1
Abstract	3
1. Introducción	5
1.1. Objetivos	6
1.2. Estructura	7
2. Estado del arte	9
3. Marco Teórico	13
3.1. Teoría de grafos	13
3.1.1. Definición y representación de grafos	13
3.1.2. Propiedades y medidas	14
3.1.3. Teoría espectral de grafos	16
3.1.4. Procesamiento de señales en grafos	18
3.1.5. Tipos de grafos	19
3.1.6. Tareas en grafos	20
3.2. Graph Embedding	21
3.3. Graph Neural Networks	23
3.3.1. Estructura general de las GNN	25
3.3.2. Filtros de grafo	28
3.3.3. Pooling de grafo	33
3.4. Graph Neural Networks en clasificación de imágenes	36
3.5. Graph Neural Networks en histopatología	39
3.5.1. Representación de grafos en patología digital	39
4. Material	43



5. Metodología	47
5.1. Perspectiva general del uso de GNN en histopatología	47
5.2. Representación de nodos y extracción de características de parches	49
5.2.1. U-Net: redes convolucionales para segmentación de imágenes biomédicas	49
5.3. Construcción de grafos	51
5.4. SegGini: Clasificación de imágenes completas mediante grafos de tejidos	55
6. Resultados y Discusión	59
7. Conclusiones	67
8. Limitaciones y Perspectivas de Futuro	69
Lista de Acrónimos	71
Bibliografía	75

Índice de figuras

3.1. Ejemplo de grafo con 4 nodos y 6 aristas (Fuente: Elaboración propia)	14
3.2. Representación de la señal en los dominios espacial y espectral (Fuente: Elaboración propia)	19
3.3. Proceso general de Graph Embedding (Fuente: Ma y Tang (2021) Cap. 2)	22
3.4. Codificación de nodos en el dominio de embedding (Fuente: Elaboración propia)	23
3.5. Operación de filtrado de grafos (Fuente: Ma y Tang (2021) Cap. 5)	24
3.6. Operación de pooling de grafos (Fuente: Ma y Tang (2021) Cap. 5)	25
3.7. Arquitectura general de GNN para tareas en nodos (Fuente: Ma y Tang (2021) Cap. 5)	26
3.8. Bloque de GNN para tareas en grafos (Fuente: Ma y Tang (2021) Cap. 5)	26
3.9. Arquitectura general de GNN para tareas en grafos (Fuente: Ma y Tang (2021) Cap. 5)	27
3.10. Flujo de trabajo estándar basado en grafos en patología digital (Fuente: David Ahmedt-Aristizabal (2021))	39
4.1. Descripción de las puntuaciones de Gleason en SICAPv2. Matriz de co-ocurrencia de los grados primario y secundario en cada biopsia (Fuente: Elaboración propia)	44
4.2. Descripción de la presencia de cáncer y grados ISUP en SICAPv2 (Fuente: Elaboración propia)	44
5.1. Arquitectura U-Net (Fuente: Ronneberger et al. (2015))	50
5.2. Parche para 16B0008045_Block_Region_0_3_5_xini_11089_yini_3611 (Fuente: Silva-Rodríguez et al. (2020b))	51
5.3. Parches asociados a la WSI 16B0008045 (Fuente: Elaboración propia)	52
5.4. Reconstrucción de la WSI 16B0008045 (Fuente: Elaboración propia)	53
5.5. Grafo de la WSI 16B0008045 (Fuente: Elaboración propia)	53
5.6. RAG de la WSI 16B0008045 (Fuente: Elaboración propia)	54
5.7. Ejemplo de RAG no conexo de la WSI 16B0022781 (Fuente: Elaboración propia)	54
5.8. Resumen de las distintas etiquetas para una muestra de WSI (Fuente: Anklin et al. (2021))	56

5.9. Resumen de la metodología SegGini (a) Construcción del grafo de tejido (b) Graph-head: WSS mediante clasificación de grafos (c) Node-head: WSS me- diante clasificación de nodos (Fuente: Anklin et al. (2021))	58
6.1. Entrenamiento y validación de la arquitectura U-Net para la obtención de repre- sentaciones latentes de parches de histologías (Fuente: Elaboración propia) . . .	61
6.2. Métricas de entrenamiento y validación para (a) Detección de cáncer (b) Clasifi- cación de grados ISUP (c) Clasificación de Gleason (suma) (d) Clasificación de Gleason (categorías) (e) Clasificación de Gleason primario (f) Clasificación de Gleason secundario	64

Índice de tablas

1.1. Relación entre la puntuación de Gleason y el grado ISUP.	5
6.1. Métricas en las distintas tareas de clasificación	63

Resumen

En el campo de la patología digital, el desafío de analizar y clasificar imágenes histológicas completas, conocidas como WSI, destaca prominentemente. Estas imágenes, con su alta resolución y complejidad intrínseca, plantean una serie de desafíos en su interpretación y análisis. A pesar de esto, con la evolución del procesamiento de imágenes y las capacidades computacionales actuales, se han ideado metodologías sofisticadas para abordar estos retos.

Este trabajo se sumerge en una propuesta concreta y novedosa para afrontar la clasificación de las WSI: Graph Neural Networks. Este enfoque se basa en la construcción de grafos a partir de parches individuales extraídos de estas WSI. Esta investigación se beneficia del acceso a la base de datos SICAPv2, que proporciona coordenadas precisas de los parches dentro de las WSI. En lugar de simplemente recomponer la WSI a partir de estos parches, el estudio propone transformar esta información en estructuras de grafo.

En estos grafos, los nodos representan cada uno de los parches, y su conectividad se establece a través de un enfoque de región de adyacencia de grafo (RAG). Estas interconexiones no solo sirven para representar la relación espacial entre los parches, sino que también se enriquecen con representaciones latentes que capturan las características de las imágenes. Las Graph Neural Networks, se entrena para procesar estas estructuras de grafo, comprendiendo y analizando las interacciones entre los nodos.

El objetivo principal es utilizar esta metodología para la detección, gradación y prognosis en histologías de próstata, identificando la presencia de cáncer y clasificando en grados ISUP y puntuación de Gleason. De esta manera, se busca no solo innovar en las técnicas de análisis de WSI, sino también mejorar la detección temprana de patologías, potenciando las decisiones clínicas y los tratamientos adecuados.

PALABRAS CLAVE: patología digital, histopatología, WSI, SICAPv2, grafo de adyacencia de región, Graph Neural Networks, cáncer de próstata, grados ISUP, puntuación Gleason

Abstract

In the field of digital pathology, the challenge of analysing and classifying whole slide images, known as WSI, stands out prominently. These images, with their high resolution and inherent complexity, pose a set of challenges in their interpretation and analysis. Despite this, with the evolution of image processing and current computational capabilities, sophisticated methodologies have been devised to address these challenges.

This work dives into a concrete and novel proposal to tackle the classification of WSI: Graph Neural Networks (GNN). This approach is based on the construction of graphs from individual patches extracted from these WSIs. This research benefits from access to the SICAPv2 database, which provides precise coordinates of the patches within the WSI. Instead of simply recomposing the WSI from these patches, the study proposes to transform this information into graph structures.

In these graphs, nodes represent each of the patches, and their connectivity is established through a region adjacency graph (RAG) approach. These interconnections not only serve to represent the spatial relationship between the patches but are also enriched with latent representations that capture the image features. The Graph Neural Networks are trained to process these graph structures, understanding and analyzing interactions between nodes.

The primary goal is to use this methodology for detection, grading, and prognosis in prostate histologies, identifying the presence of cancer and classifying into ISUP grades and Gleason scores. In this way, the aim is not only to innovate in WSI analysis techniques but also to improve the early detection of pathologies, enhancing clinical decisions and appropriate treatments.

KEYWORDS: digital pathology, histopathology, WSI, SICAPv2, region adjacency graph, Graph Neural Networks, prostate cancer, ISUP grades, Gleason score

Introducción

1

El **cáncer de próstata** es el segundo tipo de cáncer más común en hombres, con 1.4 millones de nuevos casos en la actualidad ([IAR, WCR](#)). Una vez que los estudios clínicos o análisis de sangre indican la posible presencia de este cáncer, la principal herramienta de diagnóstico es la biopsia de próstata. Se extraen pequeños fragmentos de tejido con una aguja, se procesan y tiñen con Hematoxilina y Eosina, para después ser guardadas en cristal. Posteriormente, un especialista examina la muestra bajo microscopio, identificando y clasificando las células malignas a través del sistema de **puntuación de Gleason** ([Gleason \(1992\)](#)). Este sistema categoriza las células cancerosas en grados del 3 al 5 según la diferenciación de la glándula del tejido. Por ejemplo, el grado de Gleason 3 incluye áreas glandulares bien diferenciadas, mientras que el grado 5 se refiere a células aisladas o agrupaciones de estas sin formación de lúmenes.

Los patólogos identifican la presencia de patrones específicos y diagnostican una combinación de grados de Gleason. Por ejemplo, una muestra con un grado predominante de 5 seguido de un grado 4 sería clasificada como $5 + 4 = 9$. El resultado varía entre 6 y 10 y es clave para determinar el pronóstico del cáncer y el tratamiento a seguir. Por otro lado, un resultado de $0 + 0 = 0$ indica la ausencia de cáncer en la muestra evaluada. Cabe señalar que clasificar las biopsias usando el sistema de Gleason puede resultar tedioso y mostrar variaciones en los juicios de distintos patólogos ([Gordetsky y Epstein \(2016\)](#)).

La **sociedad internacional de patología urológica** (ISUP) ofrece una perspectiva similar para clasificar el cáncer de próstata, centrada en la gravedad ([Epstein et al. \(2016\)](#)). El ISUP compara las células cancerosas con las normales y asigna un grado del 1 al 5 a las células tumorales. Un menor grado indica mayor similitud, mientras que un mayor grado señala una diferenciación más acentuada. Básandose en la puntuación Gleason, el ISUP agrupa la gravedad en categorías. Así, la relación entre la puntuación de Gleason y el grado ISUP viene reflejada en la Tabla 1.1.

Tabla 1.1: Relación entre la puntuación de Gleason y el grado ISUP.

Puntuación Gleason	Grado ISUP
2-6	1
7 (3+4)	2
7 (4+3)	3
8 (4+4/3+5/5+3)	4
9-10 (4+5/5+4/5+5)	5

Para respaldar la labor de los especialistas y mejorar la objetividad en este proceso, existen sistemas que utilizan algoritmos de visión por computador (CV) con el fin de analizar imágenes digitalizadas de las muestras y detectar áreas cancerosas. Los algoritmos de visión por computador han sido ampliamente utilizados en el análisis de **imágenes histológicas** de cáncer de próstata ([Doyle et al. \(2007\)](#), [Gertych et al. \(2015\)](#), [Toro et al. \(2017\)](#), [Ren et al. \(2017\)](#), [Lucas et al. \(2019\)](#), [Arvaniti et al. \(2018a\)](#), [Nir et al. \(2018\)](#), [Nir et al. \(2019\)](#)). Se han usado desde imágenes completas de la muestra (WSI) hasta microarrays de tejido (TMA) ([Remotti \(2013\)](#)). En cuanto a los propósitos, se aborda la detección de cáncer de próstata y la determinación de puntuaciones Gleason o grados ISUP.

Las técnicas tradicionales para lograr esto, involucran el uso de redes neuronales convolucionales (CNN), las cuales han mostrado resultados prometedores. Pese a ello, la meta principal de este estudio es investigar la aplicación y efectividad de las **redes neuronales basadas en grafos** (GNN) en la detección y clasificación de cáncer de próstata. Al explorar las GNN, buscamos nuevos enfoques que puedan ofrecer ventajas sobre las CNN, especialmente en términos de precisión y compresión de estructuras y relaciones en los datos. De esta manera, se aspira a proporcionar otras herramientas en el diagnóstico y tratamiento de esta enfermedad.

1.1. Objetivos

El objetivo central de este trabajo es explorar y evaluar el potencial de las redes neuronales de grafos (GNN) para la clasificación de cáncer de próstata a partir de imágenes histológicas. A continuación se detallan objetivos parciales que guiarán esta investigación:

- Analizar la evolución y fundamentos de las redes neuronales convolucionales (CNN) y cómo se contrastan con las propuestas de uso de redes neuronales de grafos (GNN).
- Definir, explicar y comprender la teoría subyacente a los grafos y mostrar cómo estas estructuras pueden ser utilizadas en este contexto.
- Explorar los distintos enfoques en términos de estructuras y relaciones en los datos que los grafos pueden ofrecer.
- Exponer y analizar los conceptos fundamentales relacionados con las redes neuronales de grafos, incluyendo operaciones de filtrado y pooling y el marco general de trabajo, entre otros.
- Contextualizar y conectar cómo las redes neuronales de grafos se relacionan y aplican en el estudio y análisis de imágenes histológicas.
- Diseñar y desarrollar un modelo de GNN específico para la tarea de detección y/o clasificación de cáncer de próstata.
- Evaluar empíricamente el rendimiento de la GNN diseñada.

- Investigar posibles mejoras o adaptaciones en la metodología GNN para implementaciones prácticas.
- Extraer conclusiones y conocimiento a partir de la evaluación y despliegue de la metodología del estudio realizado.

1.2. Estructura

Este documento está dividido en capítulos que detallarán aspectos relacionados con el uso de las GNN para la detección, gradación y prognosis de cáncer de próstata. Al inicio de cada capítulo se preceden los temas que se van a tratar en el mismo.

Este primero sirve como introducción general, estableciendo el contexto y exponiendo la relevancia del trabajo. Además, se definen los objetivos que guiarán el desarrollo del proyecto y nos irán acercando al objetivo principal.

El capítulo 2 realiza un análisis del estado de las investigaciones o estudios que ya existen sobre el tema y expone las razones fundamentales que hay detrás de la elección de las GNN, respondiendo a preguntas como ¿Por qué aprendizaje profundo en grafos?

El capítulo 3, se centra en la construcción de la teoría que sustenta las GNN, proporcionando una base sólida para entender las implementaciones y resultados que vendrán después.

En el capítulo 4, nos adentramos en la base de datos que contiene las imágenes histológicas de próstata. Exploramos su estructura enfocándonos en cómo se ha organizado la información y esclarecemos la proveniencia y recolección de los datos.

El capítulo 5, desgrana paso a paso la metodología adoptada para trabajar con las GNN en relación con las imágenes histológicas. Comenzamos con una perspectiva general y avanzamos hasta su aplicación para abordar el objetivo final.

En el capítulo 6, presentamos detalladamente los hallazgos obtenidos a través de nuestro estudio con las GNN. Mediante gráficos, tablas y análisis, proporcionamos una interpretación clara de lo que estos resultados significan en el contexto del análisis histológico de próstata.

El capítulo 7 recapitula las principales conclusiones derivadas de nuestro estudio con las GNN. Examinamos el impacto de la investigación en este ámbito, subrayando las contribuciones clave.

En el capítulo 8, discutimos las posibles limitaciones y las direcciones futuras que pueden surgir a partir de estas. Es esencial considerar nuevos desafíos y oportunidades que aún quedan por explorar.

Estado del arte

2

Los avances recientes en técnicas de aprendizaje profundo (DL) han llevado a qué los enfoques basados en redes neuronales de grafos se conviertan en una metodología destacada para analizar imágenes médicas, y en particular para problemas de clasificación de imágenes histológicas ([David Ahmedt-Aristizabal \(2021\)](#)).

Debido a la creciente disponibilidad de imágenes de alta resolución de muestras de tejido, la patología digital y la microscopía se han convertido en áreas atractivas para los algoritmos de aprendizaje profundo. A causa de la variabilidad en la patología y al proceso de diagnóstico, que a menudo consume mucho tiempo, los expertos clínicos han comenzado a beneficiarse de los métodos computacionales capaces de aprender características que representan de manera óptima los datos.

Existen numerosos estudios ([Deng et al. \(2020\)](#), [van der Laak et al. \(2021\)](#), [Srinidhi et al. \(2021\)](#)) que destacan cómo el aprendizaje profundo proporciona un apoyo fiable en la toma de decisiones en el diagnóstico y tratamiento, especialmente en la patología microscópica y digital, con un enfoque notable en el diagnóstico del cáncer. Esta revolución ha sido potenciada a causa de la superación de grandes retos, donde las técnicas empleadas han mostrado ser altamente efectivas en tareas como la detección, segmentación y clasificación de células y núcleos, así como en la identificación de enfermedades como el cáncer.

Los métodos de aprendizaje profundo, como las redes neuronales convolucionales, han demostrado ser un éxito en la extracción de representaciones a nivel de imagen. Sin embargo, son ineficientes al tratar con representaciones conscientes de las relaciones. Nuevas técnicas como las redes neuronales de grafos han tenido un impacto significativo en muchos dominios para manejar y describir relaciones ([Wu et al. \(2021\)](#)). Por definición, los grafos capturan relaciones entre entidades y, por lo tanto, se pueden usar para codificar información entre objetos. Las CNN tradicionales analizan áreas locales basadas en una estructura predefinida, lo que lleva a un rendimiento limitado y dificultad para interpretar patrones complejos. En cambio, las GNN ofrecen más flexibilidad para analizar datos no ordenados conservando relaciones. La adaptación de imágenes a grafos ha recibido una atención creciente, dando lugar a un nuevo campo de aprendizaje que busca aprender representaciones de grafos.

Los embeddings de grafos han aparecido en tareas de visión por computador, ya que permiten definir de manera efectiva las conexiones entre entidades o analizar imágenes estructuradas como grafos. Se han obtenido resultados interesantes en tareas de segmentación semántica, clasificación de imágenes, detección de objetos ([Qi et al. \(2018\)](#), [Chen et al. \(2019\)](#),

[Li et al. \(2020\)](#)), etc.

Las imágenes histológicas revelan detalles microscópicos de muestras de tejido, permitiendo a los patólogos diagnosticar enfermedades al observar cambios morfológicos en tejidos, relaciones espaciales entre regiones de tejido y células, densidad, entre otros factores. Los métodos basados en grafos, son capaces de modelar información a nivel celular y la microarquitectura general del tejido. Antes de la llegada del DL, se investigaron numerosos enfoques para procesar imágenes histopatológicas como grafos. Estos métodos usaban el aprendizaje automático clásico y eran menos precisos para la clasificación en comparación con las GNN.

¿Por qué aprendizaje profundo en grafos?

El aprendizaje profundo ha potenciado el análisis de imágenes médicas al permitir la identificación de características morfológicas y de textura en imágenes directamente desde los datos. Aunque las CNN han demostrado un buen rendimiento en el análisis histopatológico, no pueden capturar información compleja de vecindario ya que analizan áreas locales determinadas por el kernel convolucional. Para extraer información de interacción entre entidades, una CNN necesita alcanzar una profundidad suficiente apilando múltiples capas convolucionales, lo cual puede ser ineficiente ([Georgousis et al. \(2021\)](#)).

Las GNN operan sobre grafos, y se están volviendo cada vez más útiles para el diagnóstico y análisis médico. A continuación, se presentan algunas de las razones por las que cambiar el paradigma analítico de píxel a procesamiento de entidad-relación:

- Las CNN tradicionales no captan bien las relaciones entre imágenes. Al incorporar una GNN, podemos estimar las dependencias y mejorar la capacidad discriminativa de las características.
- Las CNN se han usado para clasificar imágenes en segmentos de tamaño fijo, usando reglas para promediar características o métricas de clases. Sin embargo, estos modelos ignoran el contexto global de la imagen, lo que puede causar inexactitudes. Se requieren arquitecturas capaces de lidiar con variaciones en las áreas de interés.
- Un sistema de predicción debe ser capaz de capturar características a múltiples escalas, algo que las CNN no logran. Transformando la imagen patológica en un grafo, se puede entender mejor la estructura y las relaciones entre las partes del tejido.
- Las representaciones de grafos pueden mejorar la interpretación final modelando relaciones entre diferentes regiones de interés. Se puede emplear para producir mapas interpretables de imágenes de tejido que muestran la contribución de cada región y sus alrededores al diagnóstico.
- Incorporando datos patológicos previos, es posible personalizar un grafo permitiendo mayor interpretabilidad en patología y facilitando el aprendizaje colaborativo.
- A diferencia de las CNN, las GNN permiten una integración más fluida de información heterogénea. En algunos casos, en el ámbito histopatológico, no solo se trata de la imagen

en sí, sino también de metadatos asociados, características genéticas y otros biomarcadores. Al representar estos datos en un grafo, las GNN pueden procesar y relacionar toda esta información de forma coherente, ofreciendo diagnósticos o perspectivas basadas en un contexto más amplio y enriquecido.

- Las GNN son una alternativa o complemento a las CNN para identificar características en imágenes.

Marco Teórico

3

En este capítulo, se presenta un recorrido por el marco teórico que fundamenta las graph neural networks (GNN). Antes de adentrarnos en las aplicaciones y desafíos concretos de las GNN, es imprescindible entender la teoría de grafos, desde su definición y representación hasta las diversas tareas asociadas a ellos. A continuación, se explora el concepto de graph embedding. A partir de aquí, nos sumergimos en un estudio exhaustivo de las GNN, considerando su estructura general, los filtros, la operación de pooling y finalizando con su integración en tareas de visión por computador.

Lo descrito en este capítulo se encuentra recogido en [Ma y Tang \(2021\)](#), [David Ahmedt-Aristizabal \(2021\)](#), [Zhou et al. \(2020\)](#), [Hamilton et al. \(2017b\)](#), [Hamilton \(2020\)](#) y [You](#).

3.1. Teoría de grafos

En esta sección, presentaremos conceptos básicos de grafos y examinaremos las representaciones matriciales utilizadas para describirlos, como la matriz de adyacencia y la matriz laplaciana. Se estudiarán las propiedades fundamentales de estas representaciones. A continuación, abordaremos los grafos atribuidos, donde cada nodo tiene asociado distintas características. Nos centraremos en tareas representativas en grafos que han sido utilizadas como punto de partida para el aprendizaje profundo en este dominio. Finalmente, se analizará como estas tareas han impulsado el desarrollo de técnicas y modelos de aprendizaje específicamente diseñados para grafos.

3.1.1. Definición y representación de grafos

Definición 3.1.1 (Grafo). Un **grafo** se define como $G = (V, E)$, donde $V = \{v_1, \dots, v_N\}$ es un conjunto de vértices con $|V| = N$ nodos y $E = \{e_1, \dots, e_M\}$ un conjunto de M aristas.

Los nodos son entidades en un grafo y las aristas describen las conexiones entre los nodos. Una arista $e \in E$ que conecta dos nodos v_e^1 y v_e^2 se puede representar como el par (v_e^1, v_e^2) .

Una clasificación de grafos se puede realizar considerando si son **dirigidos** o **no dirigidos**. En un grafo dirigido, las aristas tienen una dirección definida, lo que implica que las conexiones entre nodos tienen un sentido específico. En este caso, la arista $e = (v_e^1, v_e^2)$ se dirige del nodo v_e^1 al nodo v_e^2 . Por otro lado, en un grafo no dirigido, las aristas no tienen dirección asociada, lo que significa que las conexiones entre nodos son bidireccionales. En este escenario, el orden

de los nodos no supone ninguna diferencia, es decir, $(v_e^1, v_e^2) = (v_e^2, v_e^1)$. Es importante señalar que, a menos de que se indique lo contrario, en esta sección se restringirá nuestra discusión a grafos no dirigidos.

Un nodo v_i es adyacente a otro nodo v_j si y solo si existe una arista entre ellos. Un grafo $G = (V, E)$ se puede representar de manera equivalente mediante una matriz de adyacencia, que describe la conectividad entre los nodos.

Definición 3.1.2 (Matriz de Adyacencia). Dado un grafo $G = (V, E)$, la **matriz de adyacencia** correspondiente se denota como $A \in \{0, 1\}^{N \times N}$. La entrada i, j -ésima de la matriz de adyacencia A , indicada como $A_{i,j}$, representa la conectividad entre los nodos v_i y v_j . Más específicamente, $A_{i,j} = 1$ si v_i es adyacente a v_j , en caso contrario $A_{i,j} = 0$.

Ejemplo 3.1.1. Supongamos que tenemos un grafo $G = (V, E)$ con $V = \{1, 2, 3, 4\}$ y $E = \{(1, 1), (1, 2), (2, 3), (3, 4), (4, 1), (3, 1)\}$ como el de la Figura 3.1.

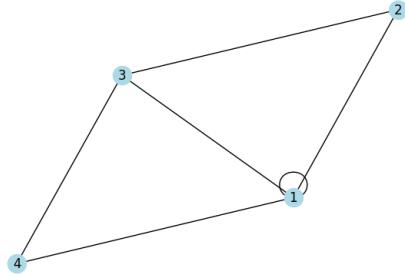


Figura 3.1: Ejemplo de grafo con 4 nodos y 6 aristas (Fuente: Elaboración propia)

La matriz de adyacencia correspondiente es:

$$A = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix}$$

Definición 3.1.3 (Matriz de Pesos). Dado un grafo $G = (V, E)$, la **matriz de pesos** correspondiente se denota como $W \in \mathbb{R}^{N \times N}$. La entrada i, j -ésima de la matriz de pesos W , indicada como $W_{i,j}$, refleja la importancia de la conexión entre los nodos v_i y v_j . Más específicamente, $W_{i,j}$ representa el peso de la arista que une v_i con v_j . Si no hay conexión entre v_i y v_j , entonces generalmente $W_{i,j} = 0$.

3.1.2. Propiedades y medidas

Definición 3.1.4 (Grado). En un grafo $G = (V, E)$, el **grado** de un nodo $v_i \in V$ es el número de nodos que son adyacentes a v_i .

$$d(v_i) = \sum_{v_j \in V} \mathbb{1}_{E(\{v_i, v_j\})}$$

donde $\mathbb{1}_{E(\cdot)}$ está definida como:

$$\mathbb{1}_{E(\{v_i, v_j\})} = \begin{cases} 1 & \text{si } (v_i, v_j) \in E, \\ 0 & \text{si } (v_i, v_j) \notin E. \end{cases}$$

El grado de un nodo v_i en G también se puede calcular a partir de su matriz de adyacencia. De esta forma, el grado de un nodo v_i se obtiene como:

$$d(v_i) = \sum_{j=1}^N A_{i,j}$$

Ejemplo 3.1.2. En el grafo de la Figura 3.1, el grado del nodo 1 es 4, ya que es adyacente a 4 nodos (1, 2, 3 y 4). Además la primera fila de la matriz de adyacencia tiene 4 elementos no nulos.

Definición 3.1.5 (Vecinos). Para un nodo v_i en un grafo $G = (V, E)$, el conjunto de sus **vecinos** $N(v_i)$ consiste en todos los nodos que son adyacentes a v_i . Es importante destacar que para un nodo v_i , la cantidad de nodos en $N(v_i)$ es igual a su grado, es decir, $d(v_i) = |N(v_i)|$.

Las conexiones entre nodos juegan un papel fundamental en los grafos. Antes de abordar en detalle esta propiedad, introduciremos conceptos básicos como recorrido o camino.

Definición 3.1.6 (Recorrido). Un **recorrido** en un grafo es una secuencia alternante de nodos y aristas, que comienza en un nodo y termina en otro. Cada arista del recorrido es incidente a los nodos inmediatamente anteriores y siguientes.

Un recorrido que comienza en el nodo v_i y termina en el nodo v_j se denota como $v_i - v_j$. El número de aristas que componen un recorrido indica la longitud del mismo. Cabe destacar que los recorridos no son únicos, ya que pueden existir distintos recorridos con diferentes longitudes.

Definición 3.1.7 (Sendero). Un **sendero** es un recorrido cuyas aristas son distintas.

Definición 3.1.8 (Camino). Un **camino** es un recorrido cuyos nodos son distintos.

Ejemplo 3.1.3. En el grafo de la Figura 3.1, $(2, (1,2), 1, (3,1), 3, (3,1), 1, (4,1), 4)$ es un recorrido 2-4 de longitud 4. Es un sendero pero no un camino ya que el nodo 1 se visita dos veces. En cambio, $(2, (1,2), 1, (3,1), 3, (3,4), 4)$ es tanto un sendero como un camino.

Teorema 3.1.1. Dado un grafo $G = (V, E)$ con matriz de adyacencia A , utilizamos A^n para denotar la n -ésima potencia de la matriz de adyacencia. El elemento i, j -ésimo de la matriz A^n indica al número de recorridos $v_i - v_j$ de longitud n .

Definición 3.1.9 (Subgrafo). Un **subgrafo** $G' = (V', E')$ de un grafo dado $G = (V, E)$ es un grafo formado por un subconjunto de nodos $V' \subset V$ y un subconjunto de aristas $E' \subset E$. Además, el subconjunto V' debe contener todos los nodos involucrados en las aristas del subconjunto E' .

Ejemplo 3.1.4. En el grafo de la Figura 3.1, el conjunto de nodos $V' = \{1, 2, 3\}$ y el conjunto $E' = \{(1, 2), (2, 3), (3, 1)\}$ forman un subgrafo G' del grafo original G .

Definición 3.1.10 (Componente conexa). Dado un grafo G , se dice que un subgrafo $G' = (V', E')$ es una **componente conexa** si existe al menos un camino entre cualquier par de nodos del grafo y los nodos en V' no son adyacentes a ningún vértice de $V \setminus V'$.

Definición 3.1.11 (Grafo conexo). Se dice que un grafo $G = (V, E)$ es **conexo** si tiene exactamente una componente conexa.

Ejemplo 3.1.5. El grafo de la Figura 3.1 es conexo.

Definición 3.1.12 (Camino más corto). Dado un par de nodos $v_s, v_t \in V$ de un grafo G , denotamos el conjunto de caminos desde el nodo v_s al nodo v_t como P_{st} . El **camino más corto** entre el nodo v_s y el nodo v_t se define como:

$$P_{st}^{SP} = \arg \min_{p \in P_{st}} |p|$$

donde p representa un camino de P_{st} de longitud $|p|$ y P_{st}^{SP} indica el camino más corto. Puede haber más de un camino más corto para cualquier par de nodos. El camino más corto entre un par de nodos describe información importante entre ellos. Indica características importantes del grafo. El diámetro de un grafo se define como la longitud del camino más corto más largo en el grafo.

Definición 3.1.13 (Diámetro). Dado un grafo conectado $G = (V, E)$, su **diámetro** se define de la siguiente manera:

$$\text{diametro}(G) = \max_{v_s, v_t \in V} \min_{p \in P_{st}} |p|$$

Ejemplo 3.1.6. El diámetro del grafo de la Figura 3.1 es 2.

3.1.3. Teoría espectral de grafos

La teoría espectral de grafos estudia las propiedades de un grafo mediante el análisis de autovalores y autovectores de su matriz Laplaciana. A continuación, se presentará la matriz Laplaciana de un grafo y se discutirán sus propiedades clave, autovalores y autovectores.

Definición 3.1.14 (Matriz Laplaciana). Dado un grafo $G = (V, E)$ con A como su matriz de adyacencia, su **matriz Laplaciana** se define como:

$$L = D - A$$

donde D es la matriz diagonal de grados $D = \text{diag}(d(v_1), \dots, d(v_N))$.

Definición 3.1.15 (Matriz Laplaciana Normalizada). Dado un grafo $G = (V, E)$ con A como su matriz de adyacencia, su **matriz Laplaciana normalizada** se define como:

$$L = D^{-\frac{1}{2}}(D - A)D^{-\frac{1}{2}} = I - D^{-\frac{1}{2}}AD^{-\frac{1}{2}}$$

donde D es la matriz diagonal de grados.

Cabe señalar que la matriz Laplaciana es simétrica ya que tanto la matriz diagonal de grados D como la matriz de adyacencia A lo son. Sea f un vector donde su i -ésimo elemento $f[i]$ está asociado al nodo v_i . Multiplicando L con f podemos obtener un vector h :

$$h = Lf = (D - A)f = Df - Af$$

El elemento i -ésimo de h se puede representar como:

$$\begin{aligned} h[i] &= d(v_i) \cdot f[i] - \sum_{j=1}^N A_{i,j} \cdot f[j] \\ &= d(v_i) \cdot f[i] - \sum_{v_j \in N(v_i)} A_{i,j} \cdot f[j] \\ &= \sum_{v_j \in N(v_i)} (f[i] - f[j]). \end{aligned} \tag{3.1}$$

Según la ecuación 3.1, $h[i]$ es la suma de las diferencias entre el nodo v_i y sus vecinos $N(v_i)$. A continuación, calculamos $f^T L f$:

$$\begin{aligned} f^T L f &= \sum_{v_i \in V} f[i] \sum_{v_j \in N(v_i)} (f[i] - f[j]) \\ &= \sum_{v_i \in V} \sum_{v_j \in N(v_i)} (f[i] \cdot f[i] - f[i] \cdot f[j]) \\ &= \sum_{v_i \in V} \sum_{v_j \in N(v_i)} \left(\frac{1}{2} f[i] \cdot f[i] - f[i] \cdot f[j] + \frac{1}{2} f[j] \cdot f[j] \right) \\ &= \frac{1}{2} \sum_{v_i \in V} \sum_{v_j \in N(v_i)} (f[i] - f[j])^2. \end{aligned} \tag{3.2}$$

De esta forma, $f^T L f$, es la suma de los cuadrados de las diferencias entre nodos adyacentes.

En otras palabras, mide lo diferentes que son los valores de los nodos adyacentes. El valor de $f^T L f$ es siempre no negativo para cualquier elección posible, lo cual indica que la matriz Laplaciana es **semidefinida positiva**.

Teorema 3.1.2. *Dado un grafo $G = (V, E)$, los autovalores de su matriz Laplaciana L son no negativos.*

Teorema 3.1.3. *Dado un grafo G , el número de autovalores 0 de su matriz Laplaciana L (multiplicidad del autovalor 0) es igual al número de componentes conexas del grafo.*

3.1.4. Procesamiento de señales en grafos

Los enfoques tradicionales de grafos se basan en vértices y aristas para representar datos. Sin embargo, métodos como las redes neuronales de grafos (GNN), permiten agregar información adicional a cada nodo. Este tipo de datos pueden tratarse como señales en grafos, las cuales capturan tanto la información de la estructura (conectividad entre nodos) como los datos (atributos en los nodos) enriqueciendo la representación del mismo.

Una **señal** se compone de un grafo $G = (V, E)$ y una función f definida en el dominio del grafo, que asigna valores reales a los nodos. Matemáticamente, esta función se puede expresar como:

$$f : V \rightarrow \mathbb{R}^{1 \times d}$$

donde d es la dimensión del vector de características asociado a cada nodo.

Además, este conocimiento puede representarse mediante una **matriz de características** $F \in \mathbb{R}^{N \times d}$ donde N denota el número de nodos y d la dimensión de las características.

Un grafo es **suave** si las características en los nodos conectados son similares. Una señal suave en el grafo tiene una baja frecuencia, ya que los valores de las características cambian lentamente a través de las aristas del grafo. La ecuación 3.2 se puede utilizar para medir la suavidad (frecuencia) de una señal f en un grafo, ya que es la suma del cuadrado de la diferencia entre todos los pares de nodos conectados. Concretamente, cuando una señal f es suave, $f^T L f$ es pequeño. El valor $f^T L f$ se denomina **suavidad** (o frecuencia) de la señal f .

Los operadores de convolución son las operaciones de propagación más utilizadas en las Graph Neural Networks. La idea principal de estos operadores es generalizar las convoluciones al dominio de los grafos. Nos permiten aplicar técnicas de procesamiento de señales en el contexto de los grafos. Desde una perspectiva clásica del procesamiento de señales, los avances en esta dirección categorizan las señales en dos dominios: el dominio espacial y el dominio espectral (dominio de frecuencia). El dominio espacial hace referencia a la representación de los datos en el propio grafo. El dominio espectral se basa en la **transformada de fourier**. Se construye sobre la teoría espectral de grafos que hemos presentado anteriormente.

En el **dominio espacial**, se definen convoluciones directamente en el grafo teniendo en cuenta su topología. Este enfoque permite procesar la información de manera local, considerando las conexiones entre los nodos vecinos. Sin embargo, uno de los desafíos principales es definir la operación de convolución considerando vecindarios de diferentes tamaños y mantener la invariancia local. La invariancia local es una propiedad deseable en las convoluciones porque permite que el modelo sea insensible a pequeñas variaciones en la posición y estructura local de los nodos en el grafo.

En el **dominio espectral**, se trabaja con una representación espectral de los grafos. En este enfoque, una señal f se transforma inicialmente al dominio espectral mediante la transformada de Fourier del grafo \mathcal{F} , luego se realiza una operación de convolución. Después de esta operación, la señal resultante se transforma de nuevo utilizando la transformada inversa \mathcal{F}^{-1} . Estas transformadas se definen como:

$$\mathcal{F}(f) = U^T f$$

$$\mathcal{F}^{-1}(f) = U f$$

En este caso, U es la matriz de autovectores de la matriz Laplaciana normalizada del grafo. Dadas las señales f y g , la operación de convolución se define como:

$$g \star f = \mathcal{F}^{-1}(\mathcal{F}(g) \odot \mathcal{F}(f)) = U(U^T g \odot U^T f)$$

donde $U^T g$ es el filtro en el dominio espectral. Si simplificamos el filtro utilizando una matriz diagonal aprendible g_w , entonces tenemos la función básica de los métodos espectrales:

$$g_w \star f = U g_w U^T f$$

Ejemplo 3.1.7. Sea G un grafo compuesto por 8 vértices y 14 aristas. Sea f la señal asignada a cada nodo que toma los valores $[0.2, 0.4, 0.6, 0.8, 0.1, 0.3, 0.5, 0.7]$. La Figura 3.2 muestra la representación gráfica de la señal en los dominios espacial y espectral.

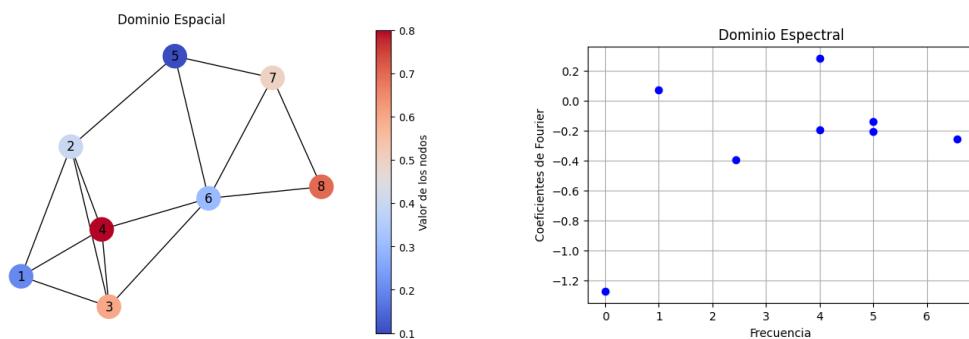


Figura 3.2: Representación de la señal en los dominios espacial y espectral (Fuente: Elaboración propia)

3.1.5. Tipos de grafos

En los apartados anteriores, se han introducido grafos simples y sus propiedades fundamentales. Sin embargo, existen numerosos contextos y aplicaciones reales que conducen a una amplia gama de representaciones y expresiones. Por ello, describiremos distintos tipos de grafos.

Homogéneos y heterogéneos

Hasta el momento, los grafos que hemos discutido son **homogéneos**. Los nodos y las aristas de los grafos homogéneos contienen el mismo tipo de información. No obstante, es posible que queramos modelar múltiples tipos de relaciones entre diferentes tipos de nodos. Los grafos heterogéneos permiten capturar esta diversidad de relaciones y entidades.

Definición 3.1.16 (Grafo heterogéneo). Un **grafo heterogéneo** G consiste en un conjunto de nodos $V = \{v_1, \dots, v_N\}$ y un conjunto de aristas $E = \{e_1, \dots, e_M\}$ donde cada nodo y cada arista están asociados a un tipo. Sea T_n el conjunto de tipos de nodos y T_e el conjunto de tipos de aristas. Existen dos funciones $\phi_n : V \rightarrow T_n$ y $\phi_e : E \rightarrow T_e$ que asignan cada nodo y cada arista a sus respectivos tipos.

Estáticos y dinámicos

Un grafo puede categorizarse como estático o dinámico en función de la evolución de las características almacenadas. Cuando las características de entrada o la topología del grafo no varían con el tiempo, el grafo se considera **estático**. Sin embargo, en muchos contextos, los grafos están en constante evolución a medida que se agregan nodos y aparecen nuevas aristas. Estos tipos de grafos en evolución se denominan como grafos dinámicos, donde cada nodo o arista está asociado a una marca de tiempo.

Definición 3.1.17 (Grafo dinámico). Un **grafo dinámico** G consiste en un conjunto de nodos $V = \{v_1, \dots, v_N\}$ y un conjunto de aristas $E = \{e_1, \dots, e_M\}$ donde cada nodo y/o arista está asociado a una marca de tiempo que indica el momento en el que surgió. Específicamente, tenemos dos funciones ϕ_v y ϕ_e que asignan a cada nodo y a cada arista sus marcas de tiempo de aparición.

En realidad, es bastante probable que no podamos registrar la marca de tiempo de cada nodo y/o arista ya que la evolución del grafo se revisa cada cierto tiempo. En cada marca de tiempo t , se registra el estado del grafo G_t . Nos referimos a este tipo de grafos como grafos dinámicos discretos.

Existen otros tipos de grafos diseñados para diferentes tareas. No se van a enumerar todos los tipos, pero hay que tener en cuenta que estas categorías son ortogonales, lo que significa que pueden combinarse.

3.1.6. Tareas en grafos

En el ámbito del procesamiento de grafos, se plantean diversas tareas con el objetivo de analizar y extraer conocimiento de los datos representados en forma de grafo. Estas tareas se pueden agrupar en tres niveles:

- **Nodos:** Las tareas centradas en nodos se enfocan en comprender las características y propiedades de los nodos de un grafo. En estas tareas se incluyen la clasificación, regresión y agrupamiento de nodos.

En la clasificación de nodos, se busca asignar una categoría o etiqueta a cada nodo del grafo. Se busca aprender una función que utilice el grafo y los nodos etiquetados para predecir la categoría de los nodos sin etiquetar. Otras tareas incluyen la regresión de nodos, donde se busca predecir un valor continuo y la agrupación de nodos, donde se intenta recopilar los nodos en conjuntos disjuntos, de manera que nodos similares estén en el mismo grupo.

- **Aristas:** Las tareas centradas en aristas se centran en predecir las conexiones o aristas de un grafo. Resultan útiles en aplicaciones donde los grafos están incompletos, con conexiones faltantes u observaciones limitadas. Dentro de estas tareas encontramos la clasificación de aristas y la predicción de conexiones.

La tarea de predicción de enlace (o conexiones) tiene como objetivo predecir qué conexiones son más probables de existir en el grafo. Para ello, se asigna un puntuaje a cada una de las posibles conexiones faltantes y se estima la probabilidad de su aparición futura.

- **Grafos:** Las tareas centradas en grafos se dedican a analizar cualidades del grafo en su conjunto. Estas tareas buscan aprender representaciones o factores distintivos de los grafos.

Una de las tareas más representativas es la clasificación, donde cada grafo se considera como una muestra de datos independiente. Consiste en el aprendizaje de una función de asignación de etiquetas, utilizando un conjunto de grafos $\{(G_i, y_i)\}$ donde G_i es un grafo y y_i la etiqueta asociada. El objetivo es tratar de predecir la categoría de los grafos no etiquetados mediante esta función.

3.2. Graph Embedding

El **Graph Embedding** (incrustación de grafos) es una técnica que busca representar las estructuras complejas de los grafos en un espacio de baja dimensión. Para lograr este objetivo, se asigna cada nodo del grafo a una representación de baja dimensión, comúnmente conocida como **node embedding**.

En lugar de trabajar directamente con la representación original de nodos y aristas, el Graph Embedding captura las relaciones y características importantes del grafo preservando información clave. Este método permite que nodos con mayor similitud estén más cercanos en el espacio de baja dimensionalidad.

En este proceso, se plantean dos preguntas fundamentales: ¿qué información se debe preservar? y ¿cómo se preserva esta información? Existen diferentes algoritmos de Graph Embedding que proporcionan diferentes respuestas a estas preguntas. En relación a la primera pregunta, se tiene en cuenta distintos tipos de información, como la información del vecindario, el papel estructural, el estado, ... Para la segunda pregunta, los métodos varían en detalles técnicos, pero la mayoría de ellos comparten la idea de reconstruir la información del grafo a partir de las representaciones de los nodos en el embedding. La intuición es que buenas representaciones deben ser capaces de reconstruir la información que se desea preservar. Por lo tanto, estas representaciones se pueden aprender minimizando el error de reconstrucción.

El Graph Embedding ofrece un enfoque poderoso para reducir la complejidad de los grafos, al tiempo que captura la esencia de sus estructuras y relaciones subyacentes. Esto permite un aprendizaje eficiente y generalizable para una variedad de tareas, sin importar las características específicas del grafo. Además, al utilizar esta técnica, se elimina la necesidad de

realizar un procesamiento de características exhaustivo para cada tarea particular, ya que las características ya están codificadas en los embeddings.

En la Figura 3.3, se ilustra el proceso general de Graph Embedding. Está compuesto por cuatro componentes clave:

- Una **función** que asigna cada nodo desde el dominio del grafo al dominio de embedding.
- Un proceso de **extracción** de información, que se encarga de identificar y obtener la información clave que queremos preservar.
- Un proceso de **reconstrucción**, que utiliza la información extraída I , para reconstruir la información del grafo original. La información reconstruida se denota como I' .
- Un **objetivo** basado en la información extraída I y la información reconstruida I' . Normalmente, se busca optimizar el objetivo para aprender todos los parámetros que intervienen en la función y/o el proceso de reconstrucción.

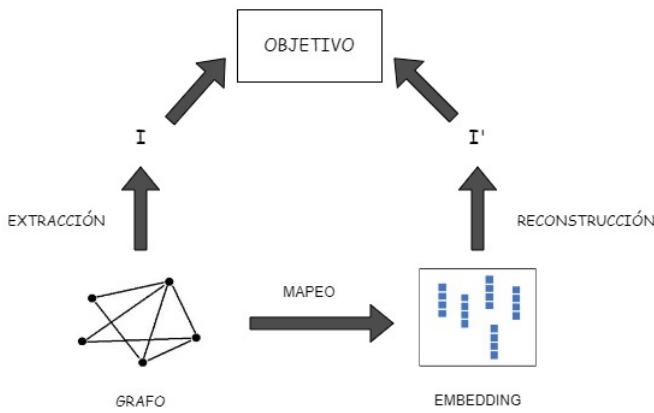


Figura 3.3: Proceso general de Graph Embedding (Fuente: Ma y Tang (2021) Cap. 2)

El proceso general de Graph Embedding descrito anteriormente también puede verse desde una perspectiva de codificación (encoder) y decodificación (decoder). El **encoder** es la función que asigna cada nodo del grafo a su representación en el espacio de embedding (Figura 3.4). El **decoder** alude al proceso de reconstrucción del grafo original utilizando las representaciones en el espacio de embedding.

El objetivo de este planteamiento es maximizar la similitud entre los nodos en el espacio de embedding y la similitud entre los mismos nodos en el grafo original. Para lograr esto, el proceso busca ajustar los parámetros del encoder y decoder de manera que las representaciones reflejen adecuadamente las relaciones y características del grafo.

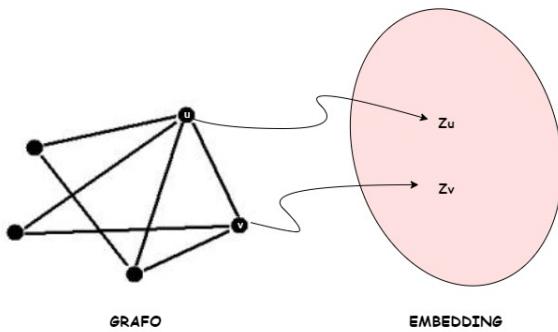


Figura 3.4: Codificación de nodos en el dominio de embedding (Fuente: Elaboración propia)

La elección clave de los métodos de Graph Embedding radica en cómo definir la similitud entre nodos: ¿qué características deben considerarse para que dos nodos tengan representaciones de embedding similares? ¿deberían dos nodos tener representaciones similares si están conectados entre sí, si comparten vecinos o si tienen roles estructurales similares? Las respuestas a estas preguntas dan lugar a distintos métodos que abordan la similitud entre nodos desde perspectivas diferentes.

Concretamente, una de las formas más populares de extraer y preservar la similitud de los nodos de un grafo es mediante técnicas que utilizan recorridos aleatorios. Si dos nodos tienden a co-ocurrir frecuentemente en esos recorridos, se consideran similares y se busca que sus representaciones en el espacio de embedding estén cercanas.

En grafos simples que son estáticos, no dirigidos y homogéneos (sección 3.1.5) destacan algoritmos como DeepWalk y Node2Vec. A pesar de que estos métodos constituyen una base esencial en el dominio de los grafos, es pertinente comentar que no profundizaremos en ellos. A pesar de su relevancia, nuestro foco no reside en su implementación. Sin embargo, es imperativo mencionarlos para ofrecer un marco teórico completo y no omitir conceptos fundamentales.

3.3. Graph Neural Networks

Las **redes neuronales de grafos** (GNN) son un conjunto de técnicas diseñadas para aplicar redes neuronales a datos con estructura de grafo. Frente a la limitación de las redes neuronales tradicionales para adaptarse a estas estructuras, las GNN se presentan como solución innovadora y efectiva a este desafío. La atención en este campo de estudio ha crecido significativamente debido al auge del aprendizaje profundo en ámbitos como la visión por computador y el procesamiento del lenguaje natural.

Las GNN se pueden entender como un proceso de aprendizaje de representaciones de grafos. En este contexto, el concepto de graph embedding surge como un elemento de este proceso. Las GNN aprovechan las representaciones del embedding para aprender directamente las propiedades y relaciones clave de los grafos. Mientras que algunas tareas se centran en aprender sobre nodos, otras buscan una comprensión del grafo en su totalidad, usando el

aprendizaje a nivel de nodo como paso previo. Nos referiremos a este proceso como operación de **filtrado de grafos**. Esta operación no cambia la estructura del grafo pero refina las características de los nodos. Se define como:

$$F^{(of)} = h(A, F^{(if)}) \quad (3.3)$$

donde $A \in \{0, 1\}^{N \times N}$ representa la matriz de adyacencia del grafo con N nodos y $F^{(if)} \in \mathbb{R}^{N \times d_{if}}$ y $F^{(of)} \in \mathbb{R}^{N \times d_{of}}$ denotan las matrices de características de entrada y salida donde d_{if} y d_{of} son sus dimensiones. Los índices if y of denotan la entrada y salida del filtro.

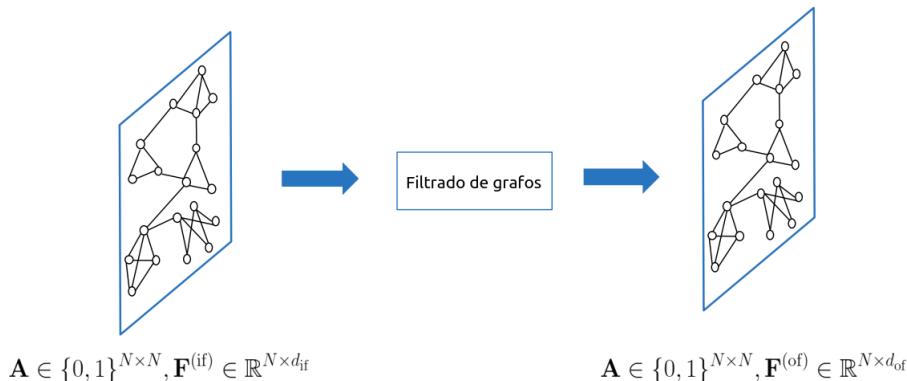


Figura 3.5: Operación de filtrado de grafos (Fuente: [Ma y Tang \(2021\)](#) Cap. 5)

Para tareas que se centran en nodos, la operación de filtrado es suficiente. Generalmente, se apilan múltiples operaciones de este tipo para determinar las características finales de un nodo. Sin embargo, al abordar tareas enfocadas en grafos completos, se necesitan operaciones adicionales para obtener características de todo el grafo a partir de las de sus nodos. De manera similar a cómo funcionan las CNN tradicionales, se proponen operaciones de agrupamiento que resuman las particularidades de los nodos y establezcan rasgos a nivel de grafo. Las CNN convencionales se aplican sobre estructuras de datos en mallas regulares, en cambio, un grafo presenta una estructura irregular. Esto subraya la necesidad de métodos de agrupación adaptados a redes neuronales que se enfocan en grafos. Estos métodos reciben un grafo como entrada y generan una versión más simplificada con menos nodos. La principal complicación radica en diseñar la estructura del nuevo grafo y determinar las propiedades de los nodos.

Una operación de **pooling** o **agrupamiento de grafos** se puede describir de la siguiente manera:

$$A^{(op)}, F^{(op)} = pool(A^{(ip)}, F^{(ip)}) \quad (3.4)$$

donde $A^{(ip)} \in \mathbb{R}^{N_{ip} \times N_{ip}}$, $F^{(ip)} \in \mathbb{R}^{N_{ip} \times d_{ip}}$ y $A^{(op)} \in \mathbb{R}^{N_{op} \times N_{op}}$, $F^{(op)} \in \mathbb{R}^{N_{op} \times d_{op}}$ son las matrices de adyacencia y características antes y después de la operación de pooling, respectivamente. Análogamente, los índices ip y op se usan para indicar la entrada y salida de la operación de agrupamiento. Cabe señalar que N_{op} denota el número de nodos del grafo simplificado y que $N_{op} < N_{ip}$.

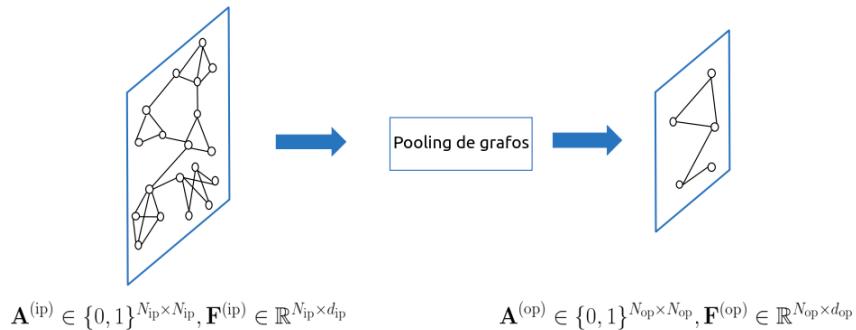


Figura 3.6: Operación de pooling de grafos (Fuente: [Ma y Tang \(2021\)](#) Cap. 5)

La arquitectura de un modelo de GNN típico se compone de operaciones de filtrado y, en algunos casos, de agrupamiento de grafos. En lo que sigue, presentaremos una visión general de cómo están estructuradas estas redes. Más adelante, se explorarán estas operaciones en profundidad, abordando métodos y aplicaciones específicas.

3.3.1. Estructura general de las GNN

En este apartado, presentaremos las bases estructurales de las GNN. Comenzaremos recordando ciertas notaciones que serán de utilidad. Un grafo se representa como $G = (V, E)$, A hace referencia a la matriz de adyacencia y las características vinculadas se denotan mediante $F \in \mathbb{R}^{N \times d}$. Cabe mencionar que, abordaremos superficialmente las tareas centradas en nodos ya que nuestro objetivo principal se situará en las tareas relacionadas con grafos.

Marco general para tareas en nodos

El esquema básico para tareas centradas en nodos puede describirse como una combinación de operaciones de filtrado y capas de activación no lineales. En este diseño, contamos con L capas de filtrado y $L - 1$ capas de activación. En la Figura 3.7, observamos que cada h_i y α_i hace referencia a la i -ésima capa de filtrado y a la capa de activación, respectivamente. Designamos el resultado de la i -ésima capa como $F(i)$. En particular, $F(0)$ se inicializa con la matriz de características F . Dado que la estructura del grafo permanece inalterada, tenemos $F(i) \in \mathbb{R}^{N \times d_i}$ donde d_i indica la dimensión de salida de la capa i -ésima. La i -ésima capa de filtrado se define como:

$$F^{(i)} = h_i(A, \alpha_{i-1}(F^{(i-1)})) \quad (3.5)$$

donde α_{i-1} es la función de activación después de la $(i-1)$ -ésima capa de filtrado. Usamos la notación α_0 para referirnos a la función identidad, ya que no aplicamos la activación a las características de entrada. La salida final $F^{(L)}$ se emplea como entrada de determinadas capas, en función de la tarea que se desee realizar.

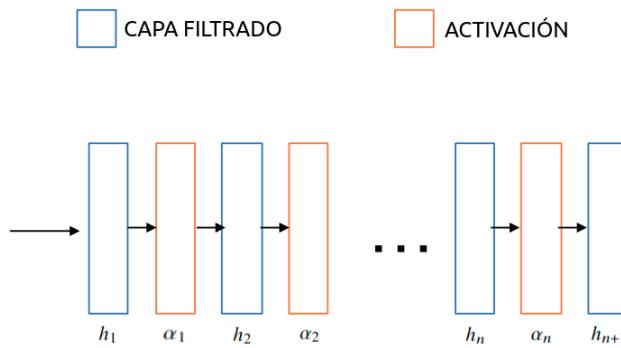


Figura 3.7: Arquitectura general de GNN para tareas en nodos (Fuente: [Ma y Tang \(2021\)](#) Cap. 5)

Marco general para tareas en grafos

La estructura general de GNN para tareas centradas en grafos se compone de tres tipos de operaciones: filtrado, activación y pooling. En este esquema, las capas de filtrado y activación persiguen metas similares a las del enfoque centrado en nodos. Por otro lado, la capa de pooling se encarga de sintetizar las características de los nodos y generar propiedades de un nivel superior que encapsulen la información completa. Generalmente, esta operación sigue a una serie de capas de filtrado y activación. Al finalizar, se genera un grafo más resumido con características de mayor jerarquía.

Estas capas pueden organizarse en un bloque específico (Figura 3.8), donde h_i , α_i y p simbolizan la i -ésima capa de filtrado, activación y pooling en dicho bloque.

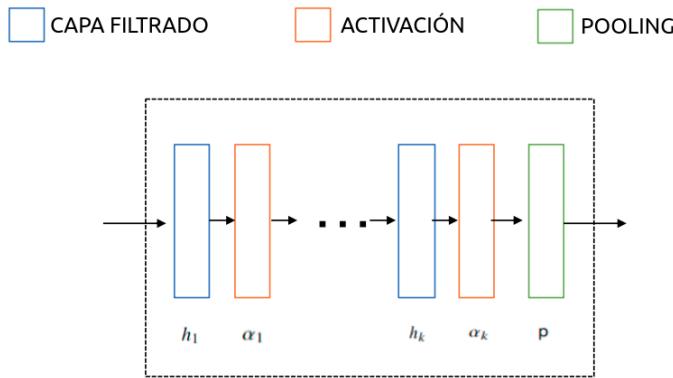


Figura 3.8: Bloque de GNN para tareas en grafos (Fuente: [Ma y Tang \(2021\)](#) Cap. 5)

Este bloque toma como entrada la matriz de adyacencia $A^{(ib)}$ y las características $F^{(ib)}$ del grafo $G_{ib} = \{V_{ib}, E_{ib}\}$. Como resultado, produce una nueva matriz de adyacencia $A^{(ob)}$ y características $F^{(ob)}$ para el grafo de salida $G_{ob} = \{V_{ob}, E_{ob}\}$.

El proceso de cálculo de un bloque se define formalmente de la siguiente manera:

$$\begin{aligned} F^{(i)} &= h_i(A^{(ib)}, \alpha^{i-1}(F^{(i-1)})) \quad i = 1, \dots, k \\ A^{(ob)}, F^{(ob)} &= p(A^{(ib)}, F^{(k)}) \end{aligned} \quad (3.6)$$

donde α^i es la función de activación para $i \neq 0$, α^0 es la función identidad y $F^{(0)} = F^{(ib)}$. Este proceso se puede resumir como:

$$A^{(ob)}, F^{(ob)} = B(A^{(ib)}, F^{(ib)}) \quad (3.7)$$

La arquitectura completa de GNN puede estar compuesta por uno o varios bloques. El proceso de cálculo de una arquitectura con L bloques se define formalmente como:

$$A^{(j)}, F^{(j)} = B^{(j)}(A^{(j-1)}, F^{(j-1)}) \quad j = 1, \dots, L \quad (3.8)$$

donde $F^{(0)} = F$ y $A^{(0)} = A$ son las características iniciales de los nodos y la matriz de adyacencia del grafo original, respectivamente.

Cada bloque produce una salida que alimenta directamente al siguiente bloque, como se ilustra en la ecuación 3.8. Si solo contamos con un bloque ($L = 1$), el diseño de la GNN es sencillo y directo, generando características a partir de su estructura original. Sin embargo, cuando tenemos más de un bloque ($L > 1$), la GNN trabaja en niveles. Comienza detallando características individuales de los nodos y, a medida que avanza, va consolidando y simplificando la información hasta obtener las características globales del grafo.

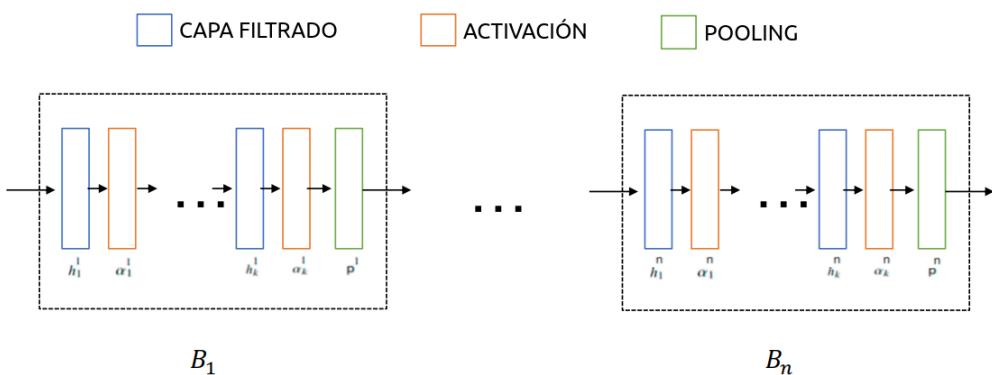


Figura 3.9: Arquitectura general de GNN para tareas en grafos (Fuente: [Ma y Tang \(2021\)](#) Cap. 5)

Este marco de trabajo brinda una visión clara de cómo operan las GNN cuando se enfocan en tareas relacionadas con grafos, y es vital para explotar su potencial en aplicaciones concretas.

3.3.2. Filtros de grafo

En el área de diseño de **filtros**, es común identificar dos enfoques predominantes: filtros de grafo orientados al dominio espacial y filtros de grafo orientados al dominio espectral. Cabe recordar que estos dominios fueron expuestos en el apartado 3.1.4. Los filtros orientados al dominio espacial se enfocan en la estructura intrínseca del grafo, considerando las relaciones entre los nodos. Aquellos orientados al dominio espectral recurren a la teoría espectral. A pesar de sus diferencias, existe una notable interconexión entre ambos enfoques. De hecho, algunos filtros se pueden utilizar en ambos dominios. Dado que nuestro estudio se centra en el dominio espacial, pondremos el foco en este subconjunto de filtros. Aunque el dominio espectral no es el centro de nuestra atención, abordaremos algunos aspectos debido a su relación con estrategias utilizadas en ambos dominios.

Similar a las CNN, las GNN buscan aprender representaciones abstractas de características a través de un proceso de paso de mensajes. En este proceso, los nodos agregan vectores de características de su vecindario para calcular un nuevo vector en la siguiente capa. Desde un perspectiva técnica, una GNN básica se compone de dos operaciones: **AGGREGATE**, que reúne representaciones de nodos vecinos y **COMBINE**, que combina la representación del nodo central con las de su vecindario para generar una representación actualizada.

Las operaciones *AGGREGATE* y *COMBINE* en la capa l -ésima de la GNN se pueden definir de la siguiente manera:

$$h_{N_v}^{(t)} = \text{AGGREGATE}^{(l)}(\{h_u^{l-1}, \quad \forall u \in N_v\}) \quad (3.9)$$

donde $h_{N_v}^{(t)}$ es la característica agregada del nodo del vecindario y h_u^{l-1} es la característica del nodo en el vecindario $N(\cdot)$ del nodo v .

$$h_v^{(t)} = \text{COMBINE}^{(l)}(h_v^{t-1}, h_{N_v}^{(t)}) = \sigma(W^t \cdot [h_v^{t-1} \parallel h_{N_v}^{(t)}]), \quad (3.10)$$

donde $h_v^{(t)}$ es la representación del nodo en la l -ésima iteración, $h_v^{(0)} = f_v$ donde f_v es el vector de características inicial para el nodo, σ denota la función sigmoide logística y \parallel la concatenación de vectores.

A continuación, examinaremos con mayor profundidad los filtros de grafos más usados en tareas de visión por computador, concretamente en análisis de patología digital. Los diferentes filtros usan agregadores distintos para obtener información de los vecinos de cada nodo usando diversas estrategias.

3.3.2.1. ChebNet

La operación de convolución para las redes convolucionales de grafo (GCN) basadas en el espectro se define en el dominio de Fourier y consiste en el cálculo de la descomposición de valores y vectores propios de la matriz laplaciana del grafo. En la definición 3.1.15 presentamos la matriz laplaciana normalizada. Esta matriz se puede representar de esta otra forma:

$$L = I - D^{-\frac{1}{2}}AD^{-\frac{1}{2}} = U\Lambda U^T$$

donde las columnas de U están formadas por los vectores propios y Λ es una matriz diagonal formada por sus autovalores. Esta operación se define como la multiplicación de la señal f con el filtro $g_\theta = \text{diag}(\theta)$ parametrizado por θ :

$$g_\theta * f = Ug_\theta(\Lambda)U^T f \quad (3.11)$$

[Defferrard et al. \(2016\)](#) propusieron una CNN espectral de Chebyshev, conocida como **ChebNet**. Esta aproximación usa polinomios de Chebyshev truncados (términos acotados) para simular los filtros espectrales. De esta manera, se evita el cálculo de los vectores propios del laplaciano y se reduce el coste computacional. La operación se define como:

$$g_\theta * f \approx \sum_{m=0}^{M-1} \theta_m T_m(\tilde{L})f \quad (3.12)$$

donde se usa el **polinomio de Chebyshev** $T_m(x)$ de orden m evaluado en $\tilde{L} = \frac{2L}{\lambda_{\max} - \lambda_{\min}}$.

3.3.2.2. GCN

Las **redes convolucionales de grafo** (GCN) se enfocan tanto en dominios espectrales como espaciales. Las **GCN** basadas en espectro se fundamentan en la transformada de Fourier del grafo y la matriz Laplaciana normalizada del mismo. Las GCN basadas en el dominio espacial definen una operación de convolución con base en las relaciones espaciales que existen entre los nodos del grafo.

Una GCN espectral es una GNN con un agregador de pooling promedio (mean-pooling). [Kipf y Welling \(2016\)](#) introdujeron las GCN mediante una aproximación localizada de ChebNet de primer orden. Se limita el filtro de convolución a $K = 1$ y utiliza una aproximación adicional de $\lambda \approx 2$ para evitar el sobreajuste y limitar el número de parámetros. Por lo tanto, la ecuación 3.12 se puede simplificar a:

$$g_\theta * f \approx \theta'_0 f + \theta'_1 f(L - I)f = \theta'_0 f + \theta'_1 D^{-\frac{1}{2}}AD^{-\frac{1}{2}}f \quad (3.13)$$

En la ecuación 3.13, θ'_0 y θ'_1 son dos parámetros sin restricciones. Una GCN asume que $\theta = \theta'_0 = -\theta'_1$, lo que lleva a la siguiente definición de convolución de grafo:

$$g_\theta * f \approx \theta(I + D^{-\frac{1}{2}}AD^{-\frac{1}{2}})f \quad (3.14)$$

La definición de una señal $f \in \mathbb{R}^{N \times C}$ con C canales de entrada y \mathfrak{F} filtros para mapas de características se generaliza de la siguiente manera:

$$Z = \tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}X\Theta \quad (3.15)$$

donde $\Theta \in \mathbb{R}^{C \times \mathfrak{F}}$ es la matriz formada por los parámetros del conjunto de filtros, y $Z \in \mathbb{R}^{N \times \mathfrak{F}}$

es la matriz de señal obtenida por convolución. Desde una perspectiva espacial (Gilmer et al. (2017)), la ecuación 3.15 se reformula como una capa de paso de mensaje que actualiza la representación del nodo x_i^k de la siguiente manera:

$$\begin{aligned} m_i^{k+1} &= \sum_{j \in N(i) \cup i} \frac{x_j^k}{|N(j)||N(i)|} \\ x_i^{k+1} &= \sigma(W^k m_i^{k+1}) \end{aligned} \quad (3.16)$$

donde m_i^k es la salida de una iteración de paso de mensaje, $|N(j)|$ y $|N(i)|$ denotan el grado del nodo j e i , W^k denota una matriz de pesos entrenable y σ es una función no lineal.

3.3.2.3. GraphSAGE

GraphSAGE es una GCN espacial que utiliza un agregador max-pooling como método de embedding de nodos. En Hamilton et al. (2017a) se presenta una extensión de las GCN para el aprendizaje de representación no supervisada inductiva utilizando funciones de agregación entrenables, en lugar de simples convoluciones aplicadas a vecindarios. Este enfoque propone un algoritmo de entrenamiento por lotes para GCN, buscando ahorrar memoria aunque ello implique una menor eficiencia temporal.

Se introducen tres operaciones de agregación: la media elemento a elemento, una LSTM y max-pooling. El agregador de media es una aproximación de la operación convolucional del marco GCN (Gilmer et al. (2017)). Una LSTM se modifica para trabajar en un conjunto no ordenado en el que se permutan los vecinos del nodo. En el agregador de max-pooling, el estado oculto de cada vecino se pasa a través de una capa completamente conectada, y luego se aplica una operación de max-pooling al conjunto de vecinos del nodo. Estas operaciones de agregación se denotan como:

$$h_{N_v}^t = \max\{\sigma(W_{\text{pool}} h_u^{t-1} + b_{\text{pool}}), \forall u \in N_v\}$$

donde N_v es el conjunto de vecindarios del nodo v , W_{pool} y b_{pool} son los parámetros a aprender, y $\max\{\cdot\}$ es el máximo elemento a elemento. Por lo tanto, siguiendo la ecuación 3.16 de paso de mensaje, la representación del nodo se actualiza según:

$$\begin{aligned} m_i^{k+1} &= \text{MEAN}_{j \in N(i) \cup i}(x_j^k) \\ x_i^{k+1} &= \sigma(W^k m_i^{k+1}) \end{aligned} \quad (3.17)$$

3.3.2.4. GAT

Basándose en el concepto de auto-atención, las **redes de atención de grafos** (GAT) (Veličković et al. (2018)), modifican las convoluciones tradicionales al incluir un mecanismo de atención durante la propagación. **GAT** es un filtro espacial de GCN que integra capas de auto-

atención con convoluciones de grafos y aprenden pesos específicos para cada vecino. [Velickovic et al. \(2018\)](#) diseñaron GAT al incorporar una capa de atención, denotada por a , que es una red neuronal parametrizada por un vector de pesos $\vec{a} \in \mathbb{R}^{2F^i}$. La capa calcula los coeficientes en los mecanismos de atención del par de nodos (i, j) según:

$$\alpha_{i,j} = \frac{\exp(\text{LeakyReLU}(\vec{a}^T [W\vec{h}_i \parallel W\vec{h}_j])))}{\sum_{k \in N_i \setminus \{j\}} \exp(\text{LeakyReLU}(\vec{a}^T [W\vec{h}_i \parallel W\vec{h}_k])))}, \quad (3.18)$$

donde \parallel representa la operación de concatenación. La capa de atención toma como entrada un conjunto de características del nodo $h = \{\vec{h}_1, \vec{h}_2, \dots, \vec{h}_N\}$, con $\vec{h}_i \in \mathbb{R}^{\mathfrak{F}}$, donde N es el número de nodos del grafo de entrada y \mathfrak{F} el número de características para cada nodo. Esto produce un nuevo conjunto de características de nodo $h' = \{\vec{h}'_1, \vec{h}'_2, \dots, \vec{h}'_N\}$, con $\vec{h}'_i \in \mathbb{R}^{\mathfrak{F}}$ como salida. Para generar características de nivel superior, se aplica una transformación lineal parametrizada por una matriz de pesos $W \in \mathbb{R}^{\mathfrak{F} \times \mathfrak{F}}$ a cada nodo y luego se emplea un mecanismo de atención, resultando en:

$$e_{ij} = a(W\vec{h}_i, W\vec{h}_j) \quad (3.19)$$

que indica la importancia de las características del nodo j para el nodo i . Las características de salida de cada nodo se pueden obtener aplicando una σ no lineal, de acuerdo a:

$$h'_i = \sigma \left(\sum_{j \in N_i} \alpha_{ij} Wh_j \right) \quad (3.20)$$

Esta capa utiliza atención multi-cabeza para estabilizar el proceso de aprendizaje. Se aplican K cabezas de atención diferentes para calcular características independientes en paralelo para luego concatenarlas. Los coeficientes de atención se utilizan para actualizar la representación del nodo según la siguiente formulación de paso de mensaje:

$$\begin{aligned} m_i^{k+1} &= \sum_{j \in N(i)} x_j^k \\ x_i^{k+1} &= \sigma \left(\alpha_{i,j}^k W^k x_j^k + m_i^{k+1} \right) \end{aligned} \quad (3.21)$$

3.3.2.5. GIN

La **red de isomorfismo de grafos** (GIN) ([Xu et al. \(2018\)](#)) es una GCN espacial que agrega información del vecindario sumando las representaciones de los nodos vecinos. Los filtros basados en isomorfismo de grafos están diseñados para interpretar grafos con diferentes nodos y aristas. La representación del nodo i se actualiza posteriormente mediante una red neuronal de perceptrón multicapa (MLP):

$$\begin{aligned} m_i^{k+1} &= \sum_{j \in N(i)} x_j^k \\ x_i^{k+1} &= F((1 + \epsilon) \cdot x_i^k + m_i^{k+1}) \end{aligned} \quad (3.22)$$

donde F es un MLP y ϵ un parámetro que puede ser aprendido o fijo. Las funciones de agregación y lectura de **GIN** son inyectivas, por lo que están diseñadas para lograr la máxima capacidad discriminativa.

3.3.2.6. Otros filtros de grafos

En patología digital, se toman en cuenta otras arquitecturas de GNN para la evaluación de grafos de entidades, entre ellas se encuentran:

- **Red neuronal de grafo de borde (EGNN):** Se incluyen las características de los bordes al utilizar la estructura del grafo en la red.
- **Filtrado espacial robusto (RSF):** Estos modelos espaciales son más versátiles al tratar con grafos heterogéneos, ya que las entradas del grafo pueden ser fácilmente incorporadas en la función de agregación.
- **GraphSAGE Adaptativo:** Filtros con la habilidad de aprender de manera más efectiva las características de incrustación entre nodos. Utilizan un patrón aprendible para agregar de forma adaptativa características de incrustación de múltiples niveles para cada nodo.
- **Red de conocimiento de salto (JK-Net):** Este método fue sugerido para adaptar y utilizar distintos rangos de vecindad para cada nodo, optimizando la representación de sus características.
- **GCN espacial mejorada por características (FENet):** Este filtro está diseñado para estudiar grafos no isomórficos, a diferencia de los isomórficos que poseen exactamente la misma matriz de adyacencia. El mecanismo de mejora de características selecciona adaptativamente la representación del nodo de diferentes capas de convolución. El modelo toma un agrupamiento por suma para capturar toda la información estructural de la representación completa del grafo.
- **Red Neuronal de ondícula a múltiples escalas (MS-GWNN):** Este filtro espectral aprovecha la propiedad de localización de las ondículas del grafo para realizar análisis a múltiples escalas con una variedad de parámetros de escala en paralelo. De esta manera, MS-GWMM ofrece alta eficiencia y buena interpretabilidad en la convolución de grafos.

3.3.3. Pooling de grafo

Los filtros de grafo afinan las características de los nodos sin modificar la estructura del grafo. Tras esta operación, cada nodo tiene una nueva representación de características. Estas operaciones de filtrado suelen ser suficientes para tareas centradas en nodos. Sin embargo, para tareas centradas en grafos, es necesario una representación completa de este. Para lograrlo, es esencial resumir la información de los nodos.

Existen dos tipos principales de información cruciales para generar representaciones de grafo: las características de los nodos y la estructura del grafo. Se espera que el graph embedding conserve ambas.

De forma similar a las CNN, se proponen capas de **pooling** o agrupación de grafo para generar representaciones a este nivel. Las primeras versiones de estas capas eran generalmente planas, es decir, creaban la representación a nivel de grafo directamente a partir de las representaciones de nodos en un solo paso. Posteriormente, se desarrollaron métodos de agrupación jerárquicos para resumir la información reduciendo gradualmente el grafo original. En el pooling jerárquico, suele haber varias capas de agrupación, seguidas de un conjunto de filtros, como se muestra en la Figura 3.9.

Una operación de pooling toma un grafo como entrada y produce uno nuevo reducido. Esto proceso se precisa en la ecuación 3.6 como:

$$A^{(op)}, F^{(op)} = \text{pool}(A^{(ip)}, F^{(ip)}) \quad (3.23)$$

A continuación, se describen las operaciones de pooling plano y jerárquico.

3.3.3.1. Pooling plano

Una capa de **pooling plano** genera de forma directa una representación a nivel de grafo a partir de representaciones de nodos. Esta operación, no crea un nuevo grafo, sino que genera un nuevo nodo. Por lo tanto, en lugar de utilizar la ecuación 3.23, el proceso de agrupación se puede resumir de la siguiente manera:

$$f_G = \text{pool}(A^{(ip)}, F^{(ip)})$$

donde $f_G \in \mathbb{R}^{1 \times d_{op}}$ es la representación del grafo.

Seguidamente, presentaremos algunas de las capas de pooling plano más representativas. Las operaciones de agrupación máxima y promedio en las CNN pueden adaptarse a las GNN. Específicamente, la operación de **pooling máximo** se puede expresar como:

$$f_G = \max(F^{(ip)})$$

La operación *máx* se aplica de la siguiente manera:

$$f_G[i] = \max(F_{:,i}^{(ip)})$$

donde $F_{:,i}^{(ip)}$ denota el i-ésimo canal de $F^{(ip)}$. De manera similar, la operación de **pooling promedio** del grafo aplica la operación de promedio de la siguiente manera:

$$f_G = \text{ave}(F^{(ip)})$$

En este contexto, se propone una operación de agrupación basada en atención, denominada **pooling global de atención**. Se utiliza una medida de atención que evalúa la importancia de cada nodo para resumir sus propias representaciones y generar la representación del grafo. La medida de atención para el nodo v_i se calcula como:

$$s_i = \frac{\exp(h(F_i^{(ip)}))}{\sum_{v_j \in V} \exp(h(F_j^{(ip)}))}$$

donde h asigna $F_i^{(ip)}$ a un escalar, que luego se normaliza mediante softmax. Con la puntuación aprendida, la representación del grafo se puede resumir como:

$$f_G = \sum_{v_i \in V} s_i \cdot \tanh(F_i^{(ip)} \Theta_{ip})$$

donde Θ_{ip} son parámetros a aprender y la función de activación $\tanh()$ puede reemplazarse por la identidad.

3.3.3.2. Pooling jerárquico

Las operaciones de pooling plano suelen pasar por alto la información estructural jerárquica del grafo al resumir las representaciones de los nodos en una representación del grafo. Las operaciones de **pooling jerárquico** buscan conservar esta información estructural al ir reduciendo el grafo gradualmente. Se pueden categorizar las operaciones de agrupamiento jerárquico según la manera en que reducen el grafo.

Una técnica reduce el grafo mediante un muestreo, es decir, seleccionando los nodos más relevantes para formar el grafo reducido. Otro enfoque agrupa nodos del grafo original para crear supernodos que actúan como nodos en el grafo reducido. La principal diferencia entre estos dos métodos es que el basado en muestreo conserva nodos del grafo original, mientras que el basado en supernodos genera nuevos nodos para el grafo reducido. A continuación, detallaremos algunos métodos representativos en estas dos categorías.

Pooling jerárquico basado en reducción

Para simplificar el grafo de entrada, se selecciona un conjunto de nodos N_{op} basados en ciertos criterios de importancia. Luego, con estos nodos, se define la estructura del grafo y las características de los nodos del grafo simplificado.

En este tipo de agrupación, hay tres componentes clave:

- Establecer el criterio de reducción.
- Formar la estructura del grafo reducido.
- Definir las características de los nodos del grafo reducido.

La capa **gPool**, fue la primera en utilizar la estrategia de reducción. En gPool, la importancia de los nodos se calcula a partir de las características de entrada $F^{(ip)} \in \mathbb{R}^{N_{ip} \times d_{ip}}$ como:

$$y = \frac{F^{(ip)} \times p}{\|p\|}$$

donde $p \in \mathbb{R}^{d_{ip}}$ se aprende para proyectar las características de entrada en medidas de importancia. Una vez obtenidas las medidas y , se seleccionan los k nodos más importantes:

$$idx = rank(y, N_{op})$$

N_{op} es el número de nodos del grafo reducido y idx representa los índices de los nodos seleccionados. La estructura del nuevo grafo se induce como:

$$A^{(op)} = A^{(ip)}(idx, idx)$$

donde $A^{(ip)}(idx, idx)$ realiza la extracción de filas y columnas de $A^{(ip)}$ con los índices seleccionados.

En gPool, el cálculo de la importancia se basa solamente en las características de entrada. Para considerar la estructura del grafo, se utiliza el filtro GCN. De tal manera que, la importancia se calcula como:

$$y = \alpha(GCN(A^{(ip)}, F^{(ip)})) \quad (3.24)$$

donde α es una función de activación, como *tanh*. En la ecuación 3.24, y es un vector en lugar de una matriz. Esta operación es conocida como **SAGPool**.

Pooling jerárquico basado en supernodos

El pooling basado en reducción selecciona un subconjunto de nodos según su importancia. Sin embargo, al hacerlo, se pierde información de los nodos no seleccionados. El pooling basado en supernodos simplifican el grafo agrupando los nodos del grafo original en clústeres. Cada clúster es tratado como supernodo, es decir, son nodos del nuevo grafo.

Para conseguir esto, un enfoque basado en supernodos generalmente implica:

- Crear supernodos a partir de los clústeres de nodos.
- Establecer una estructura entre supernodos.

- Definir las características de cada supernodo.

El algoritmo **diffpool** ofrece una forma de generar estos supernodos. Utiliza una matriz que define cómo se asignan los nodos del grafo original a los supernodos. Esta matriz se aprende utilizando un filtro GCN:

$$S = \text{softmax}(\text{GCN}(A^{(ip)}, F^{(ip)})) \quad (3.25)$$

donde $S \in \mathbb{R}^{N_{ip} \times N_{op}}$. Cada columna de esta matriz representa un supernodo. Al aplicar la función softmax en cada fila, se normalizan las probabilidades de forma que la suma de cada fila sea 1. El elemento ij-ésimo indica la probabilidad de asignar el nodo i al supernodo j.

Utilizando la matriz S, podemos obtener la estructura del grafo simplificado a partir del original:

$$A^{(op)} = S^T A^{(ip)} S \in \mathbb{R}^{N_{op} \times N_{op}}$$

De manera similar, las características de los supernodos se derivan combinando linealmente las características de los nodos del original:

$$F^{(op)} = S^T F^{(inter)} \in \mathbb{R}^{N_{op} \times N_{op}}$$

donde $F^{(inter)} \in \mathbb{R}^{N_{ip} \times N_{op}}$ son las características intermedias obtenidas mediante filtros GCN:

$$F^{(inter)} = \text{GCN}(A^{(ip)}, F^{(ip)}) \quad (3.26)$$

Es posible aplicar múltiples filtros GCN aunque solo se muestre uno en la ecuación 3.26. En resumen, el proceso de diffpool se describe como:

$$A^{(op)}, F^{(op)} = \text{diffpool}(A^{(ip)}, F^{(ip)})$$

3.4. Graph Neural Networks en clasificación de imágenes

Los grafos están presentes en distintas tareas del área de visión por computador. Sirven como herramienta para establecer relaciones en cuestiones como reconocimiento basado en esqueleto, respuesta a preguntas visuales, ... En esta sección, abordaremos el papel que juegan los grafos en tareas de clasificación de imágenes con GNN.

La clasificación de imágenes se encarga de asignar imágenes a categorías específicas. Las GNN optimizan este proceso, destacando en contextos como clasificación sin muestras previas (zero-shot), con pocas muestras (few-shot) y multi-etiqueta.

Un clasificador de CNN se divide en una fase de extracción de características con capas convolucionales y de agrupación y una fase final de clasificación donde se determina la probabilidad de que una muestra pertenezca a una categoría. La lógica detrás de la clasificación

de imágenes con GNN es comparable a cómo funciona la clasificación con CNN. Sin embargo, si bien la idea subyacente es similar, existen detalles y particularidades distintivas en cada enfoque, partiendo de que las CNN procesan datos estructurados y GNN manejan grafos.

Clasificación de imágenes Zero-Shot

Tradicionalmente, se asume que disponemos de numerosas imágenes de cada categoría para entrenar los modelos de clasificación. Estos clasificadores sólo pueden reconocer imágenes de las categorías con las que fueron entrenados. Para reconocer imágenes de una nueva categoría, se necesitan multitud de ellas y es indispensable reentrenar el modelo con estos nuevos datos.

La tarea de **clasificación de imágenes sin muestras previas** (Zero-Shot) consiste en aprender un clasificador para una nueva categoría sin imágenes de entrenamiento, basándose únicamente en información sobre esa categoría, como su descripción o su relación con otras. En algunos estudios, se utilizan GNN para aprender estos clasificadores, propagando información de otras categorías a través de un grafo de conocimiento que describe las relaciones entre ellas.

La ventaja principal de este enfoque es que permite aprender sobre nuevas categorías sin requerir datos nuevos. Esto es especialmente útil en escenarios donde los datos de entrenamiento pueden ser escasos o simplemente no existir para ciertas categorías.

Dado un conjunto de n categorías, solo las primeras m tienen imágenes de entrenamiento, las demás no cuentan con ninguna. A pesar de ello, cada categoría tiene una descripción que se traduce en un espacio semántico, y todas las categorías están interconectadas en un grafo de conocimiento.

Cuando tenemos una imagen, es posible identificar sus rasgos principales mediante CNN. De esta forma, las categorías con muchas imágenes permiten crear un clasificador eficiente. El reto surge al intentar desarrollar clasificadores para las categorías sin imágenes. La solución propuesta es aprovechar las descripciones traducidas semánticamente y las relaciones en el grafo de conocimiento. En lugar de intentar entrenar una red neuronal convencional, que requeriría muchas más imágenes de muestra, se utiliza una especializada en grafos, la cual se nutre tanto de la descripción semántica de las categorías como de su interrelación en el grafo.

Esta técnica no está exenta de desafíos. El rendimiento del clasificador puede verse afectado por la calidad y precisión del grafo de conocimiento. Si las relaciones entre las categorías no están bien definidas, esto puede llevar a predicciones erróneas. Además, la construcción de descripciones semánticas precisas y útiles para cada categoría es esencial. Si estas descripciones no capturan adecuadamente las relaciones, el clasificador podría no funcionar como se espera.

Clasificación de imágenes Few-Shot

Mientras que la clasificación sin muestras se centra en aprender clasificadores para categorías sin imágenes de entrenamiento, la **clasificación con pocas muestras** (Few-Shot) se ocupa de categorías que disponen de un número muy limitado de imágenes etiquetadas.

En este escenario, todas las categorías cuentan con algunas imágenes etiquetadas. Esto significa que es posible aprender un clasificador para cada una, aunque la fiabilidad varíe dependiendo del número de muestras disponibles. Los modelos desarrollados para categorías con muchas imágenes son robustos y fiables. No obstante, aquellos para categorías con escasas imágenes, podrían no ser lo suficientemente precisos.

Para mejorar estos clasificadores, se puede usar un enfoque similar al descrito anteriormente, utilizando GNN para refinarlos. Este refinamiento se realiza a partir de los clasificadores previamente aprendidos.

Una técnica útil consiste en simular conjuntos de entrenamiento falsos, tomando un número limitado de muestras de categorías con muchas imágenes, y entrenando clasificadores con ellos. Estos, junto con los clasificadores originales, alimentan el modelo GNN.

Esta GNN no usa incrustaciones de palabras, sino que se basa en estos clasificadores falsos. Una vez entrenados, mejora la clasificación de categorías con menos imágenes. Estas relaciones se mapean usando grafos de conocimiento. Estos grafos se crean a partir de la similitud entre los clasificadores antes de refinarlos.

Clasificación de imágenes con múltiples etiquetas

La **clasificación de imágenes con múltiples etiquetas** busca identificar un conjunto de objetos en una imagen. Una forma básica de hacerlo es considerar cada etiqueta como un problema de clasificación binaria, determinando si un objeto está o no en la imagen. Sin embargo, algunos objetos suelen aparecer juntos en el mundo real. Reconocer estas relaciones entre objetos mejora la precisión del modelo.

Algunos enfoques proponen aprender las relaciones entre objetos usando un grafo basado en la frecuencia con la que aparecen en las imágenes. Con este grafo, aplican una red neuronal basada en grafos para aprender clasificadores que consideran las interdependencias de los objetos.

Para ello, se construye el grafo considerando los objetos como nodos. Las conexiones, o aristas, entre ellos se basan en cuántas veces coexisten en las imágenes. Una vez construido, se utiliza para entrenar la red neuronal basada en grafos. Los clasificadores obtenidos se emplean para determinar la presencia de cada objeto en la imagen.

3.5. Graph Neural Networks en histopatología

En la sección 3.4, exploramos la aplicación de las redes neuronales de grafos en tareas de clasificación de imágenes. Esta discusión sirve como preámbulo y calentamiento para comprender las posibilidades de estas redes en tareas de visión por computador. Ahora, nos adentraremos en el objetivo principal de nuestro estudio, la histopatología. Profundicemos en cómo estas redes están marcando la diferencia en este campo.

3.5.1. Representación de grafos en patología digital

Al convertir imágenes histológicas de pacientes en grafos que codifican el contexto espacial de células y tejidos, se mejora la interpretabilidad en diversas tareas. Las representaciones de grafos junto con los modelos basados en GNN, permiten a los patólogos entender y justificar los resultados obtenidos. Las GNN son versátiles, pudiendo adaptarse para diversas predicciones, ya sea a nivel de nodo o de grafo.

Un flujo de trabajo estándar con grafos implica varias fases, incluyendo definición de nodos y topología, y elección de arquitectura GNN. La Figura 3.10 muestra un marco tradicional que engloba las siguientes fases del proceso:

1. **Detección de entidades:** La imagen WSI se convierte inicialmente en uno o varios grafos. Las entidades de estos grafos pueden ser núcleos, segmentos o regiones de tejido.
2. **Representación de nodos:** Las características de los nodos incluyen atributos diseñados manualmente o a través de aprendizaje profundo para describir estas entidades.
3. **Representación de grafos:** Las conexiones del grafo representan relaciones inherentes, ya sean espaciales o semánticas, entre las entidades.
4. **Modelo:** Las representaciones de grafos se procesan utilizando GNN y sus variantes, como GCN, GraphSAGE, GAT y GIN. Esta fase también incorpora diferentes estrategias de agrupación de grafos, ya sea pooling plano o jerárquico.
5. **Interpretación del grafo:** Se utilizan herramientas para interpretar el modelo GNN.

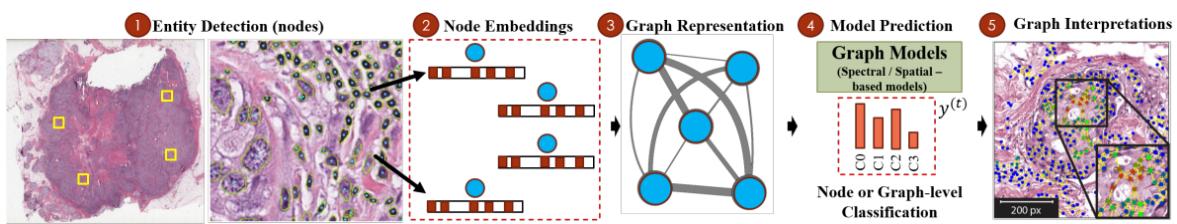


Figura 3.10: Flujo de trabajo estándar basado en grafos en patología digital (Fuente: [David Ahmedt-Aristizabal \(2021\)](#))

Los métodos basados en grafos comúnmente emplean datos que se encuentran en dos formatos principales. Por un lado, están las **WSI** (imágenes completas), que son imágenes de alta

resolución formadas por la combinación de múltiples fragmentos. Por otro lado, encontramos los **TMA** (microarrays de tejidos).

Estas imágenes histológicas se representan como un grafo, donde los nodos y las aristas simbolizan entidades biológicas e interacciones entre ellas. Estas entidades pueden ser biológicas, como núcleos y regiones de tejido, o definidas en base a parches de imagen. El proceso de construcción de un grafo de entidades en patología digital se compone de los siguientes pasos:

- **Definición de nodos:** Las WSI contienen regiones que no son tejidos. Para identificar las áreas de tejido, se emplea un proceso de segmentación. Una forma habitual de representación son los grafos celulares , donde los nodos son **células**. Estos grafos requieren que los modelos sean entrenados y afinados para la detección o segmentación de dichas células. Existen métodos específicos, como Hover-Net, CIA-Net, UNet ([Ronneberger et al. \(2015\)](#)) y cGAN, diseñados para segmentar núcleos en diferentes órganos. Otra alternativa es la representación de nodos a través de **parches** de tamaño fijo, tomados directamente de la WSI y descartando áreas sin tejido. Pueden seleccionarse parches clave usando criterios de color. También se han empleado modelos preentrenados para identificar regiones tumorales. De esta forma, regiones de tejido relevantes pueden emplearse como nodos.
- **Representación de nodos:** La representación de los nodos puede incluir características, como propiedades morfológicas y topológicas. En los grafos celulares, se incorporan características derivadas del modelo utilizado para identificar núcleos de células. En los grafos basados en parches, se usan redes neuronales profundas para aprender una representación de características de los parches. En los métodos que emplean parches, se han adoptado CNN, modelos preentrenados como MobileNetV2, DenseNet y ResNet y modelos de segmentación encoder-decoder, para extraer características.
- **Definición de aristas:** Las aristas del grafo reflejan las interacciones entre células o tejidos, determinando la probabilidad de que dos entidades cercanas interactúen. Esta topología a menudo se define heurísticamente usando un umbral de proximidad predefinido, una regla del vecino más cercano o un modelo probabilístico. La estructura del grafo también puede determinarse mediante la construcción de un **grafo de adyacencia de región** (RAG) ([Potjer \(1996\)](#)).

Una vez construido el grafo, la siguiente fase consiste en determinar la arquitectura o modelo que mejor se adapte a nuestras necesidades. Esto implica seleccionar las operaciones de filtrado y pooling, ya vistas en los apartados [3.3.2](#) y [3.3.3](#). Una vez definidas, se aborda la fase de entrenamiento donde el modelo aprende a analizar la información del grafo.

Existen diferentes enfoques en función del nivel de supervisión:

- **Aprendizaje supervisado:** Se basa en el uso de datos ya etiquetados.

- **Aprendizaje débilmente o parcialmente supervisado:** Se refiere a datos que están parcialmente etiquetados.
- **Aprendizaje semi-supervisado:** Fusiona la información de un pequeño conjunto de datos etiquetados con un conjunto más amplio sin etiquetar. A estos últimos se les asignan etiquetas artificiales, permitiendo al modelo aprender tanto de ejemplos etiquetados como de aquellos que no lo están.
- **Aprendizaje auto-supervisado:** Se centra en descubrir estructuras y patrones intrínsecos en los datos. Tras una representación inicial, el modelo se refina mediante un conjunto de datos etiquetado. Existen algunas técnicas destacadas como la codificación predictiva contrastiva (CPC), el autoencoder de textura (Deep Ten) y los autoencoders variacionales (VAE).

Material

4

En este capítulo, presentamos una revisión detallada de **SICAPv2**. Esta base de datos, descrita en [Silva-Rodríguez et al. \(2020a\)](#), constituye el material esencial para nuestro estudio en curso. Se encuentra disponible en [Silva-Rodríguez et al. \(2020b\)](#).

La base de datos SICAPv2 es, según los autores, la colección pública más grande de biopsias de próstata H&E con anotaciones a nivel local de grados de Gleason. SICAPv2 amplía la base de datos presentada anteriormente en [Esteban et al. \(2019\)](#).

[Silva-Rodríguez et al. \(2020a\)](#) identificaron cuatro principales bases de datos de imágenes de tejido de cáncer de próstata. La base de datos más grande fue publicada en [Weinstein et al. \(2013\)](#), contando con 720 biopsias. Sin embargo, la falta de anotaciones tanto a nivel local como de biopsia de grados Gleason limita el uso de estos datos. La base de datos proporcionada por [Arvaniti et al. \(2018b\)](#) incluye anotaciones a nivel de píxel de 886 áreas de biopsias. Desafortunadamente, estas áreas no reflejan los patrones heterogéneos de las estructuras locales de cáncer y lesiones benignas, por lo que carecen de relevancia clínica para el diagnóstico de grados Gleason a nivel de diapositiva. Limitaciones similares se encuentran en conjuntos de datos utilizados en otros artículos. Por este motivo, SICAPv2 supone una gran aportación en cuanto a la publicación de una extensa base de datos con etiquetas a nivel de biopsia y anotaciones de Gleason a nivel de pixel. Por primera vez, se indica la presencia de regiones glandulares cribiformes.

SICAPv2 contiene 155 biopsias de 95 pacientes que dieron su consentimiento. Estas muestras de tejido se cortaron, tiñeron y digitalizaron con un escáner Ventana iScan Coreo a una magnificación de 40x para obtener imágenes de todo el corte (WSI). Expertos patólogos del Hospital Clínico de Valencia analizaron las muestras y asignaron una puntuación de Gleason para cada biopsia . En situaciones de incertidumbre, la clasificación se llevo a cabo por consenso. Los grados de Gleason en las biopsias varían entre regiones no cancerosas, y grados 3, 4 y 5 ([Figura 4.1](#)).



Figura 4.1: Descripción de las puntuaciones de Gleason en SICAPv2. Matriz de co-ocurrencia de los grados primario y secundario en cada biopsia (Fuente: Elaboración propia)

Es importante considerar no sólo la distribución de las puntuaciones de Gleason, sino también su correspondencia con los grados ISUP. Para tener una visión más completa, podemos traducir la Figura 4.1 en otra basada en el sistema de gradación ISUP. Esto queda representado en la Figura 4.2:

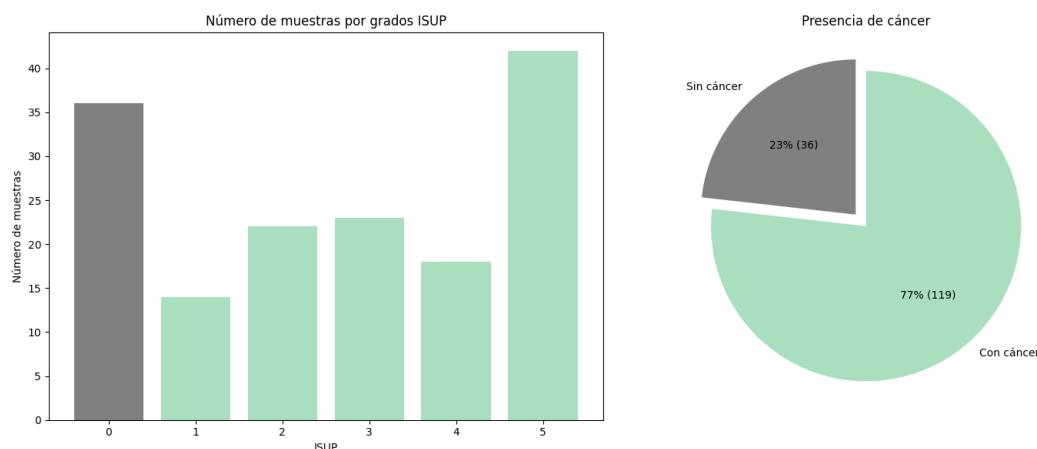


Figura 4.2: Descripción de la presencia de cáncer y grados ISUP en SICAPv2 (Fuente: Elaboración propia)

Estas muestras fueron anotadas localmente utilizando un software basado en las bibliotecas OpenSeadragon. Para procesar las grandes WSI, se redujo su resolución y se dividieron en fragmentos de 512x512, optimizando los valores para la detección de patrones cancerosos. Tras un proceso de selección y descarte, se obtuvieron varios parches con diferentes puntuaciones. Además, se obtuvo una máscara de la presencia de tejido en los parches. Para desarrollar un modelo capaz de predecir el grado de Gleason, se excluyeron los parches con menos del 20 % de tejido. Después de este procedimiento, la base de datos contiene 4417 parches no cancerosos, 1635 etiquetados como GG3, 3622 como GG4 y 665 como GG5. Cabe señalar que si un parche contenía más de un grado anotado, se asignaba el mayor grado como

etiqueta.

A continuación, se describe la estructura y contenido de directorios y tablas dentro de la base de datos SICAPv2:

■ Directorios:

- *images*: Contiene parches de histologías de próstata extraídos de una ampliación de 10x. Los parches tienen un tamaño de 512x512 píxeles y se superponen en un 50 % entre ellos. Solo se seleccionan los parches que contienen más del 20 % de tejido. El nombre de cada archivo sigue el siguiente formato:

`<Slide_ID>_Block_Region_<region_y_x>_xini_<xini>_yini_<yini>`

Cada nombre incluye un prefijo que indica el ID de la diapositiva, la posición de la región del parche mediante *region_y_x* y las posiciones *xini* y *yini*.

- *masks*: Contiene máscaras con anotaciones codificadas de las imágenes del directorio *images*. Las anotaciones en las máscaras se codifican con los valores: 0 (no canceroso), 1 (grado gleason 3), 2 (grado gleason 4) y 3 (grado gleason 5).
- *partition*: Contiene tablas que describen la partición propuesta de validación cruzada basada en pacientes de la base de datos, las etiquetas de los grados Gleason y la verdad del patrón cribiforme.

■ Tablas:

- *wsi_labels*: Tabla que contiene información sobre las WSI. Incluye los siguientes campos: *slide_id* (identificador de la imagen), *patient_id* (identificador del paciente asociado), *Gleason_primary* y *Gleason_secondary* (puntuación de Gleason, que incluye los grados primario y secundario).

Metodología

5

La metodología representa un elemento primordial de este proyecto. En este capítulo, se desarrollarán las estrategias y enfoques que se han tomado, apoyados en las bases teóricas expuestas con antelación y destacando, en particular, los contenidos de las secciones [3.3](#) y [3.5](#).

La metodología se centra en la aplicación de GNN para el diagnóstico de cáncer de próstata. Los detalles técnicos, como el contenido y la estructura del grafo, se precisarán a lo largo de este capítulo.

5.1. Perspectiva general del uso de GNN en histopatología

Con los avances en técnicas de escaneo TMA y WSI, junto con el acceso a grandes conjuntos de imágenes digitales de tejido, los métodos de aprendizaje profundo han avanzado significativamente en áreas como la localización de tumores y predicciones relacionadas con el cáncer. Para obtener una visión global del uso de GNN en histopatología, examinaremos tanto el enfoque basado en parches de tejido como el basado en características celulares ([David Ahmedt-Aristizabal \(2021\)](#)).

Representación mediante grafos de células:

Un enfoque para la representación de imágenes histológicas implica la transformación en grafos utilizando las células como puntos clave. Cada célula identificada se representa como un vértice en el grafo y se trazan conexiones entre ellas según su proximidad en el espacio. Luego, se procesa esta estructura para clasificar la imagen según su escala de Gleason.

Cabe destacar que las técnicas en este contexto son muy diversas. A modo ilustrativo, con la ayuda de un modelo GCN y utilizando una estrategia auto-supervisada conocida como codificación predictiva contrastiva (CPC) se identifica el riesgo de padecer esta enfermedad. La técnica se basa en características extraídas de la forma y textura de las células, así como en el aprendizaje derivado del CPC. Mediante herramientas avanzadas como la convolución GraphSAGE, se procesa el grafo para captar la distribución y características de las células en un nivel más amplio. Este método solo representa una de las estrategias posibles, existen muchas más disponibles.

Representación mediante grafos de parches:

Otros enfoques opta por transformar imágenes patológicas en grafos de parches. En estos, los nodos simbolizan áreas o parches específicos de la imagen y las aristas representan las conexiones entre ellos, reflejando su relación intrínseca. La identificación de estos parches se realiza a través de distintos métodos, como el análisis de color, la densidad celular o técnicas de atención. Una vez identificados, se emplean distintas estrategias para extraer las características de estos parches y construir un vector que representará a los nodos en el grafo. Posteriormente, se aplica un modelo de GNN para clasificarlos. Es vital entender que hay una distinción entre los grafos basados en tejidos, que capturan áreas biológicamente significativas, y los grafos de parches que interconectan zonas con múltiples entidades biológicas.

Dentro de este panorama, los patólogos enfrentan el desafío de anotar las imágenes de forma precisa. Ante esta necesidad, [Anklin et al. \(2021\)](#) presentaron un enfoque de segmentación semántica basado en grafos (SegGini). Este modelo integra relaciones tanto locales como globales entre regiones de tejido para segmentaciones contextuales con etiquetas inexactas e incompletas. Se fundamenta en el uso de superpíxeles, considerando que un único parche puede incluir varias estructuras distintas. De estos, se extraen características morfológicas y espaciales para configurar los nodos, estableciendo la estructura del grafo a partir de conexiones espaciales entre superpíxeles.

Utilizando un grafo de parches, se usa un modelo GIN que aprende características contextuales e interacciones entre tejidos para segmentación semántica. SegGini asigna una etiqueta de clase a cada nodo. Las características de nodos resultantes son procesadas por un modelo principal para segmentar patrones Gleason y finalmente, clasificar a nivel de imagen.

Dado que nuestro principal objetivo es la clasificación de imágenes completas WSI, nos enfrentamos al desafío de trabajar con un conjunto de datos que, en lugar de ofrecernos imágenes completas, nos brinda parches individuales de estas. A pesar de este escenario inicial, SICAPv2 nos proporciona el identificador de cada parche y las coordenadas y detalles asociados, lo que nos permite discernir la imagen completa WSI a la que pertenece cada fragmento y su posicionamiento dentro de ella. Esto nos brinda la posibilidad de reconstruir las WSI, pero, en lugar de simplemente reensamblar las imágenes, optamos por un enfoque basado en la construcción de grafos mediante parches. A través de estos grafos, podremos representar y comprender las interrelaciones entre los diferentes fragmentos de la imagen, permitiendo una interpretación más coherente y detallada del tejido en cuestión. En las siguientes secciones, profundizaremos en la mecánica y metodología detrás de este enfoque, delineando cómo definiremos los nodos, cómo extraeremos las características relevantes de cada parche, y qué modelo será el más adecuado para nuestra tarea de clasificación. Todo ello, guiado por la premisa de transformar nuestra limitación inicial en una fortaleza para un diagnóstico más preciso.

5.2. Representación de nodos y extracción de características de parches

En los enfoques basados en parches, las redes neuronales son la herramienta principal para aprender representaciones de características. Un desafío es que, si la entidad es más grande que el tamaño de parche especificado, se deben procesar múltiples parches. Algunos autores han agregado características de parches vecinos y las han combinado para obtener una representación de nodo central.

En esta línea, se han adoptado CNN específicas como VGG, ResNet y MobileNet-v2, preentrenadas en ImageNet, o ConvNext ([Medina \(2022\)](#)) ofreciendo ventajas de modelos ya consolidados. Paralelamente, se ha trabajado en modelos de segmentación encoder-decoder, como U-Net ([Ronneberger et al. \(2015\)](#)), diseñado desde cero con el objetivo de comparar, contrastar y ampliar nuestro espectro de métodos de representación.

De entre todas las redes preentrenadas disponibles, se seleccionaron aquellas que demostraron una óptima adaptación al problema. Se tomó en cuenta tanto las referencias disponibles como la eficiencia en cuanto a dimensión del espacio latente. Simultáneamente, optamos por una arquitectura U-Net debido a su reconocido desempeño en ofrecer métricas de reconstrucción superiores frente a otras configuraciones encoder-decoder. Mientras que las redes preentrenadas se describen ampliamente en la literatura y su aplicación es directa, nos centraremos más detenidamente en desglosar el proceso de afinamiento y adaptación de la arquitectura U-Net en nuestra metodología.

5.2.1. U-Net: redes convolucionales para segmentación de imágenes biomédicas

Las redes convolucionales han superado los estándares más avanzados en muchas tareas de visión por computador. A pesar de que estas redes ya llevan bastante tiempo existiendo, su éxito se vio limitado por al tamaño de los conjuntos de entrenamiento y de las propias redes.

Por lo general, las redes convolucionales se han utilizado principalmente en la clasificación de imágenes. No obstante, existen tareas más complejas en el ámbito biomédico como localización y clasificación a nivel de píxel. Realmente, utilizaremos las tareas de segmentación y reconstrucción de imágenes como un pretexto, con el objetivo principal de capturar las características de las histologías. De esta manera, se obtendrán representaciones latentes de calidad que posteriormente serán incorporadas como datos esenciales en estructuras de grafos.

Algunas técnicas más tradicionales, enfrentan ciertas limitaciones. En primer lugar, el proceso puede resultar lento dado que se trabaja individualmente con cada parche, ocasionando redundancias a causa de la superposición de estos fragmentos. Adicionalmente, el tamaño del parche influye directamente en la cantidad de capas de pooling requeridas. En otras palabras, es complicado encontrar un equilibrio entre el tamaño, precisión y contexto que la red puede procesar. En ocasiones, es difícil acceder a miles de imágenes de entrenamiento. Por este motivo, estos métodos recurren a la ampliación de datos mediante deformaciones elásticas a

5.2. REPRESENTACIÓN DE NODOS Y EXTRACCIÓN DE CARACTERÍSTICAS DE PARCHES

las imágenes de entrenamiento. Estas deformaciones suelen ser la variación más común en los tejidos y se pueden simular de manera eficiente. Gracias a SICAPv2, no hemos tenido que enfrentarnos a este desafío, ya que contamos con una enorme variedad de conjunto de datos. Además, el modelo en U que se presentará a continuación reduce la necesidad de una enorme cantidad de datos de entrenamiento.

En [Ronneberger et al. \(2015\)](#), se expone una arquitectura más sofisticada basada en [Long et al. \(2014\)](#). El modelo que se propone busca mejorar las redes convolucionales tradicionales. En lugar de simplemente reducir la resolución de las imágenes a través del proceso, añadimos capas que aumentan la resolución nuevamente, reemplazando las operaciones de pooling con operaciones de sobremuestreo. Estas capas adicionales, se combinan con características previamente reducidas para proporcionar una salida más precisa. La belleza de este diseño es que tiene un flujo que primero reduce (contracción) y luego aumenta (expansión), tomando una forma de U .

La arquitectura de la red se muestra en la Figura 5.1. Esta consta de un camino de contracción (lado izquierdo) y un camino expansivo (lado derecho). El camino de contracción se asemeja a la arquitectura típica de una red convolucional. Se compone de la aplicación repetida de dos convoluciones 3×3 , seguidas cada una por una unidad lineal rectificada (ReLU) y una operación de agrupación máxima (max pooling) 2×2 con paso 2 para la reducción de dimensiones. Con cada paso de reducción, duplicamos el número de canales de características. Cada etapa en el camino expansivo implica un sobremuestreo del mapa de características seguido de una convolución transpuesta 2×2 (up-sampling) que reduce a la mitad el número de canales de características, una concatenación con el mapa de características correspondiente del camino de contracción, y dos convoluciones 3×3 , seguidas cada una por una ReLU. En la capa final, se utiliza una convolución 1×1 para mapear cada vector de características al número deseado de clases.

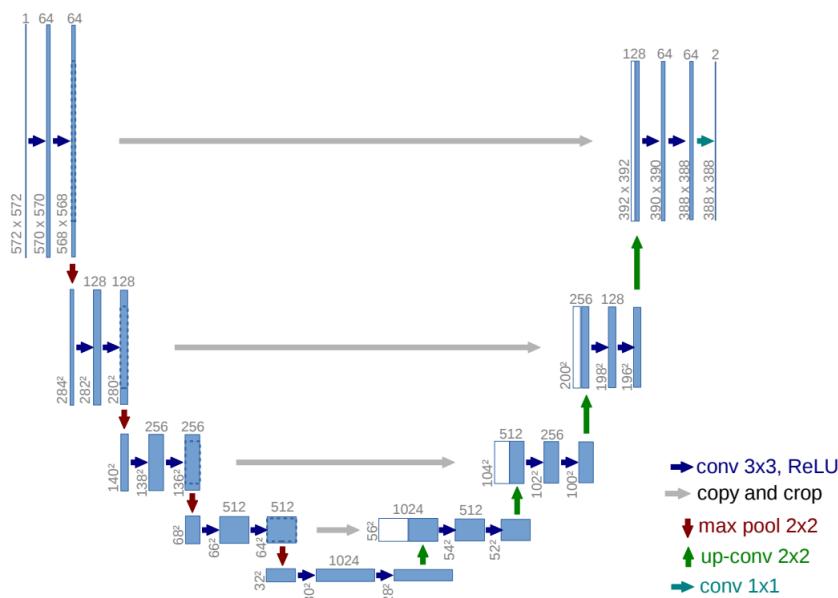


Figura 5.1: Arquitectura U-Net (Fuente: [Ronneberger et al. \(2015\)](#))

En la Figura 5.1, cada bloque azul se corresponde con un mapa de características multi-canal. El número de canales se indica en la parte superior del recuadro. El tamaño se indica en el borde inferior izquierdo. Los recuadros blancos representan mapas de características copiados. Las flechas indican las distintas operaciones

5.3. Construcción de grafos

Ante el desafío de clasificar imágenes completas (WSI), se ha adoptado un enfoque basado en la construcción de grafos mediante parches individuales. Basándonos en SICAPv2, esta decisión se fortalece. Tal y como se comentó en el capítulo 4, esta base de datos no solo proporciona parches, sino que revela sus coordenadas dentro de la imagen original, facilitando así nuestra tarea. En lugar de simplemente unir estos fragmentos para recomponer la WSI, vemos una oportunidad para representar las interacciones entre los parches a través de grafos. Este enfoque enriquecerá nuestra capacidad de diagnóstico, aprovechando la información intrínseca de cada parche (Figura 5.2).

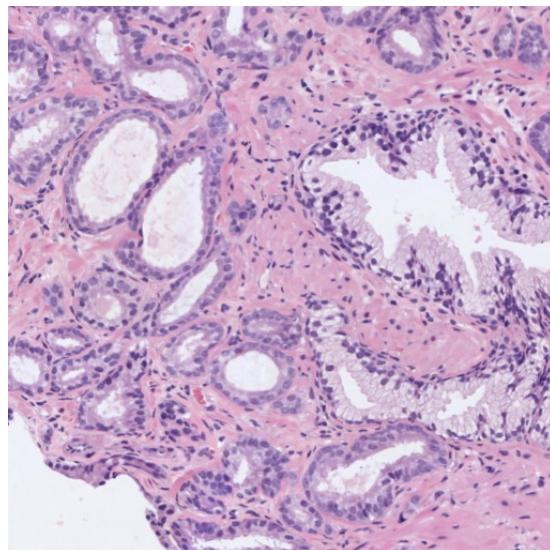


Figura 5.2: Parche para 16B0008045_Block_Region_0_3_5_xini_11089_yini_3611 (Fuente: Silva-Rodríguez et al. (2020b))

A continuación, vamos a explicar un proceso básico de reconstrucción de imágenes a partir de parches, para más adelante profundizar en la construcción de grafos de manera similar.

Dado un conjunto de parches de histologías de próstata, examinamos y extraemos información importante de estas imágenes, como el identificador y las coordenadas dentro de la imagen completa. La clave para la reconstrucción es reunir todos los parches asociados a ese mismo identificador (Figura 5.3). Una vez que los tenemos todos, los colocamos en el espacio de acuerdo a sus coordenadas originales (Figura 5.4).

Es importante recordar que no incluimos todos los parches originales, sino únicamente aquellos que se proporcionan en SICAPv2 y que, por tanto, tienen más del 20 % de tejido. Este proceso nos ayuda a tener una visión general de cómo sería la imagen completa si juntáramos todos los parches en su lugar correcto.

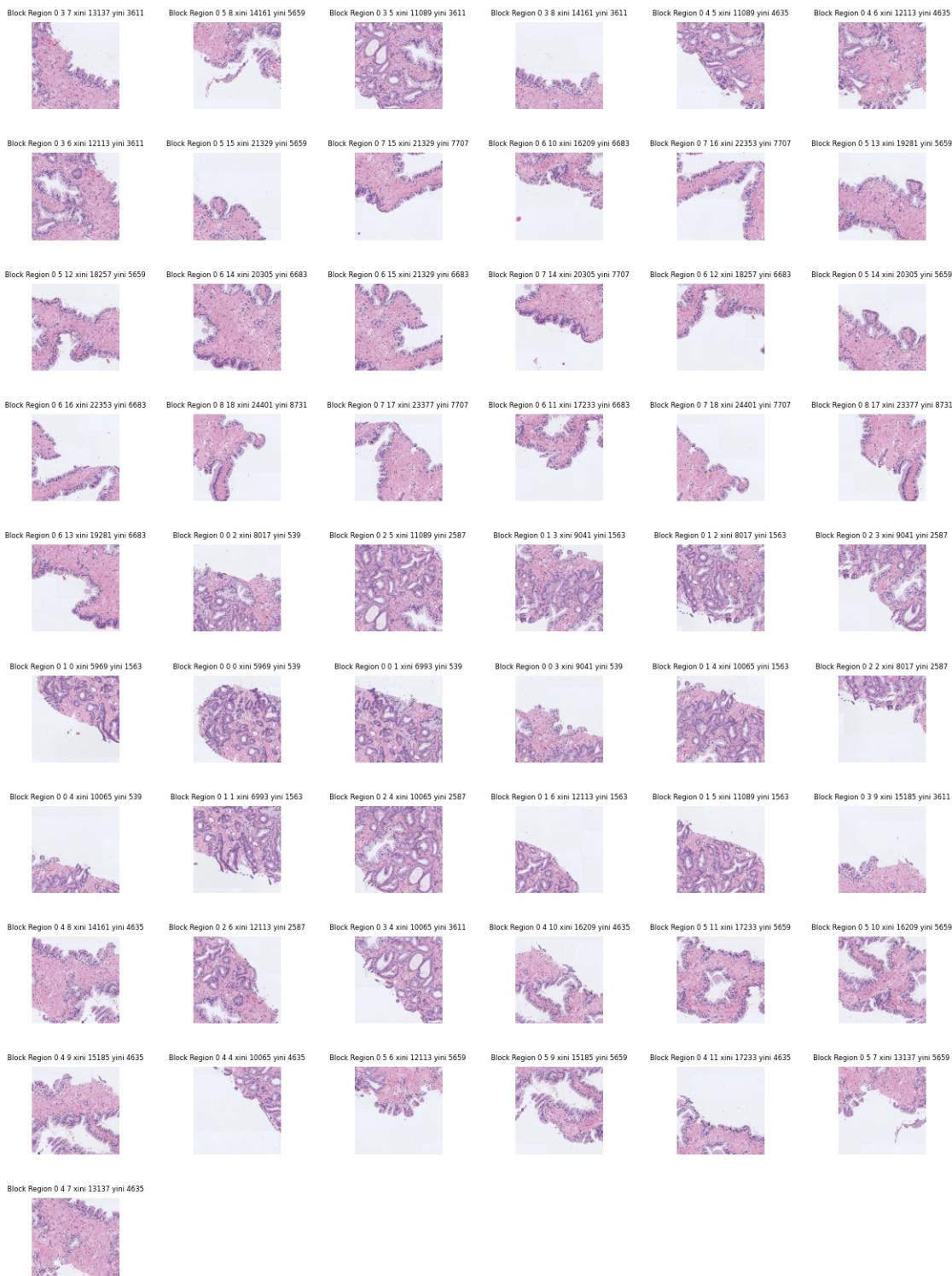


Figura 5.3: Parches asociados a la WSI 16B0008045 (Fuente: Elaboración propia)

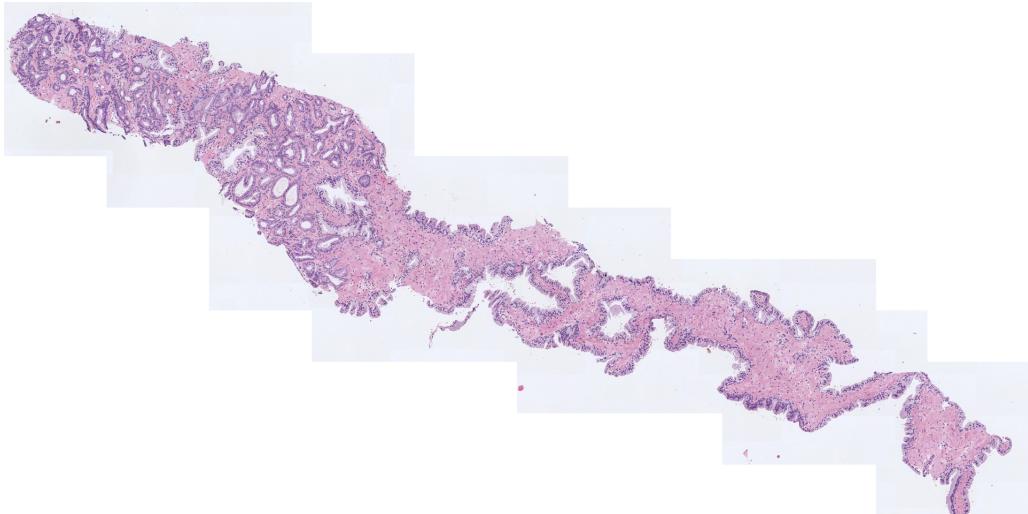


Figura 5.4: Reconstrucción de la WSI 16B0008045 (Fuente: Elaboración propia)

La reconstrucción de imágenes es solo un comienzo de nuestra exploración. Sirve como punto de partida para la construcción de grafos. A partir de aquí, explicaremos cómo estos parches se combinan para obtener una representación como grafo.

La construcción de grafos es una fase crucial del proyecto. Para cada imagen completa asociada a un identificador, creamos un grafo único. Cada uno de los parches de la imagen se convierten en un nodo dentro de este grafo. Lo que hace que estos nodos sean realmente interesantes son las características que les asignamos. Para ello, utilizamos las representaciones latentes calculadas previamente para enriquecer cada nodo con información valiosa.

Pero la pregunta es: ¿cómo conectamos estos nodos entre sí? Tenemos dos enfoques distintos. Uno de ellos, bastante directo, implica que todos estén conectados entre sí (Figura 5.5). Esto significa que existe una arista entre cada par de nodos. Las aristas llevan consigo información sobre la distancia entre los parches en la imagen. El segundo enfoque, llamado **grafo de adyacencia de región** (RAG), es más selectivo (Figura 5.6). Aquí, solo conectamos nodos que son vecinos cercanos en la imagen completa. Esto significa que dos parches solo estarán conectados si son contiguos en la imagen.



Figura 5.5: Grafo de la WSI 16B0008045 (Fuente: Elaboración propia)

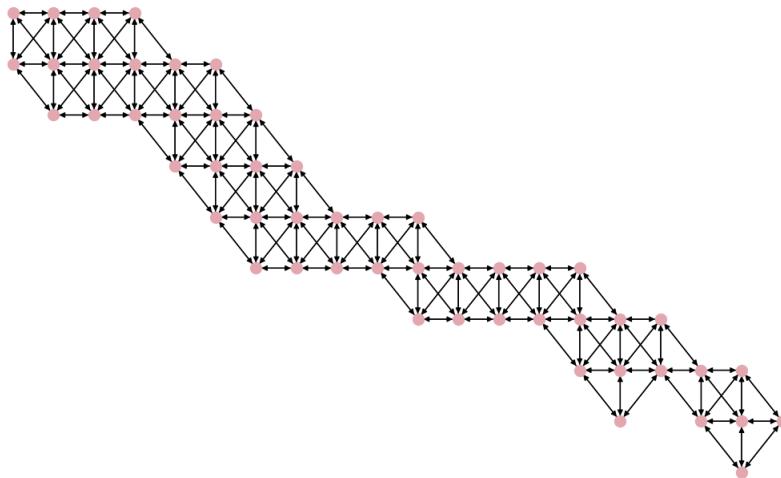


Figura 5.6: RAG de la WSI 16B0008045 (Fuente: Elaboración propia)

Partiendo de la sección 3.1.5, podemos asegurar que ambos enfoques generan grafos que se clasifican como no dirigidos, lo que significa que las relaciones entre los nodos son bidireccionales. Además, estos grafos son homogéneos, ya que todos los nodos comparten el mismo tipo de información y estáticos ya que su topología no varía con el tiempo o en diferentes momentos de la tarea. Esta clasificación es relevante, ya que determinará qué tipos de modelos de redes neuronales de grafos pueden aplicarse a estos grafos.

Una de las diferencias visuales más notables entre ambos enfoques radica en su conectividad. En el primer enfoque, los grafos resultantes son siempre conexos. Esto significa que, sin importar cuál sea el par de parches, siempre encontraremos un camino que los conecta, ya que todas las conexiones posibles están consideradas. En contraste, en el enfoque del grafo de adyacencia de región (RAG), existe la posibilidad de que ciertos parches seleccionados estén muy alejados entre sí y no haya una conexión directa entre ellos. En este caso, el grafo resultante puede no ser conexo, lo que significa que algunos parches pueden quedar aislados del resto de la estructura (Figura 5.7). Esta diferencia en la conectividad entre los dos enfoques es un aspecto a considerar, ya que afecta directamente la estructura y a las propiedades del grafo.

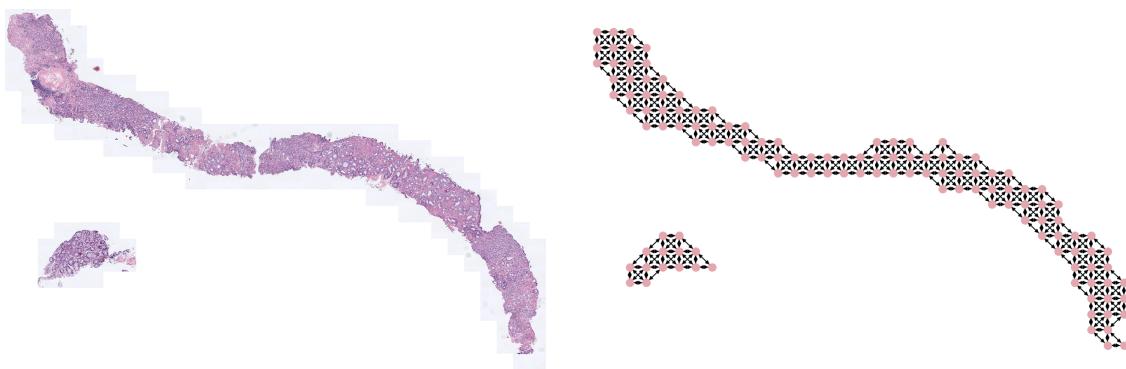


Figura 5.7: Ejemplo de RAG no conexo de la WSI 16B0022781 (Fuente: Elaboración propia)

Asimismo, estos grafos también incluyen características a nivel de grafo completo, como la puntuación de gleason primario y secundario y el grado ISUP. Estas puntuaciones proporcionan información sobre la naturaleza de las muestras y serán de utilidad para nuestra tarea de clasificación. Añadir que la elección entre estos dos enfoques dependerá del modelo GNN que se utilice, ya que diferentes modelos pueden requerir un enfoque específico.

5.4. SegGini: Clasificación de imágenes completas mediante grafos de tejidos

Segmentar y/o clasificar imágenes histológicas es crucial para el diagnóstico y toma de decisiones precisas y rápidas de los patólogos. Aunque existen técnicas asistidas para destacar áreas segmentadas en diapositivas histológicas escaneadas, requieren conjuntos de datos extensos con píxeles anotados. Esta obtención es laboriosa, lenta y costosa, y en muchos casos, prácticamente imposible. En [Anklin et al. \(2021\)](#), se presenta **SegGini**, un método de segmentación basado en grafos. Este método puede trabajar con anotaciones débiles e incompletas, siendo capaz de segmentar imágenes de cualquier tamaño, desde matrices de tejidos hasta imágenes completas.

Aunque a primera vista el artículo [Anklin et al. \(2021\)](#) parece estar exclusivamente enfocado en la segmentación, es tentador descartar métodos que por su título o presentación, se inclinen hacia esta tarea. Sin embargo, es imperativo subrayar que en la metodología de SegGini se alberga una fase de clasificación basada en grafos. En otras palabras, dentro del proceso de segmentación se encuentran pasos intermedios o componentes que se dedican a clasificar. Por tanto, si bien nuestra tarea no se alinea directamente con el propósito final de [Anklin et al. \(2021\)](#), es viable aprovechar las herramientas que nos interesen. Es más, vale la pena señalar que, aunque el título de [Anklin et al. \(2021\)](#) sugiere lo contrario, en [David Ahmedt-Aristizabal \(2021\)](#), se introduce abarcando tanto la clasificación como la segmentación.

En [Chan et al. \(2021\)](#), se revisan métodos **WSS** que emplean diferentes enfoques de aprendizaje para segmentación. A pesar de su éxito, estos métodos enfrentan desafíos en imágenes histológicas debido a la presencia de objetos de gran detalle y a menudo fronteras ambiguas entre componentes del tejido. Aunque se propusieron algunos métodos WSS para histología, muchos se limitan a segmentaciones específicas o a trabajar con imágenes de tamaño fijo. Por tanto, es necesario un método WSS que funcione con imágenes histológicas grandes y variadas, y que utilice tanto el contexto local como el global. La mayoría de los métodos se centran en tareas de clasificación binaria y, aunque algunos manejan múltiples clases, requieren anotaciones detalladas que van más allá de las necesidades clínicas estándar. Lo ideal sería que un método WSS pudiera aprender de anotaciones generales y no detalladas.

SegGini crea representaciones de grafos mediante parches de tejido de una imagen histológica, donde cada nodo simboliza una región. Luego, adopta un enfoque de clasificación a nivel de nodo para segmentar la imagen. Esta particularidad sugiere que, más allá de la segmentación, SegGini tiene el potencial de ser utilizado para una clasificación completa y detallada de

5.4. SEGGINI: CLASIFICACIÓN DE IMÁGENES COMPLETAS MEDIANTE GRAFOS DE TEJIDOS

las imágenes histológicas, ofreciendo un enfoque más global. Esta dualidad de segmentación y clasificación en su enfoque lo convierte en una herramienta sumamente versátil.

En ambas tareas, se usan conjuntos de datos de cáncer de próstata, que incluyen tanto TMA como WSI. En el caso de la segmentación, se clasifican patrones benignos (B), grado de gleason 3 (GG3), grado 4 (GG4) y grado 5 (GG5) mediante anotaciones a nivel de pixel inexactas. En la tarea de clasificación, se recurre a anotaciones de la imagen (Figura 5.8 (c)).

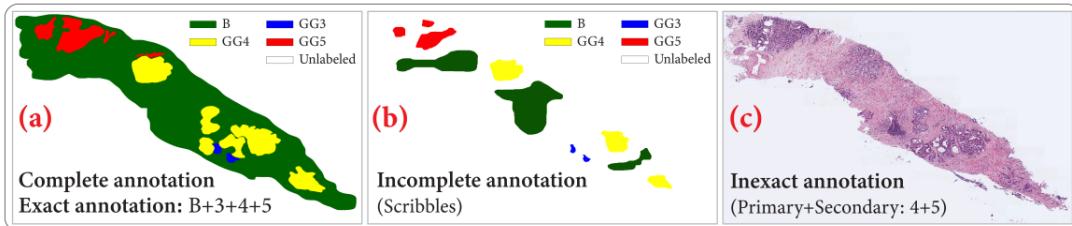


Figura 5.8: Resumen de las distintas etiquetas para una muestra de WSI (Fuente: [Anklin et al. \(2021\)](#))

En esta sección, describiremos la metodología propuesta SegGini (Figura 5.9). Primero, una imagen histológica de entrada es normalizada para reducir variabilidades de apariencia ([Vahadane et al. \(2016\)](#)). Luego, se extrae la máscara de tejido y los distintos parches para convertir la imagen en un **grafo de tejido** (TG). Definimos un TG como un grafo $G = (V, E)$ donde los nodos V codifican regiones de tejidos en forma de superpíxeles y las aristas E representan interacciones entre estos. Un superpixel es una agrupación de píxeles adyacentes en una imagen que comparten características similares, como el color o la textura. El concepto detrás de los superpíxeles es simplificar la representación de la imagen al agrupar píxeles que tienen características similares, lo que a menudo lleva a una representación más coherente y semánticamente significativa.

Para la construcción del **TG** hemos seguido las explicaciones y procedimientos previamente detallados donde se incluyen la representación de nodos, extracción de características y construcción de la topología del grafo basado en la proximidad espacial.

Dado un TG, aspiramos a aprender representaciones discriminativas de los nodos que se beneficien del contexto, del microentorno del tejido y de las interacciones. Estas representaciones de nodo se utilizan para realizar tareas de clasificación y/o segmentación. Para ello, empleamos graph neural networks, una familia de redes que pueden operar con estos datos en forma de grafo. En particular, se utilizan capas de isomorfismo de grafo (GIN), una arquitectura de GNN potente y rápida que funciona de la siguiente manera.

Para cada nodo $v \in V$, GIN utiliza un operador de suma para agregar las características de los vecinos del nodo $N(v)$. Luego, actualiza las características del nodo h_v combinándolas con las características actuales del nodo h_v a través de un perceptrón multicapa (MLP). Después de T capas GIN, es decir, adquiriendo contexto hasta T -saltos, las características intermedias del nodo $h_v^{(t)}$, $t = 1, \dots, T$ se concatenan para definir las representaciones de nodo contextualizadas. Formalmente, una GNN con normalización por lote (BN) se describe para $v, u \in V$ como:

$$\begin{aligned}
 h_v^{(t+1)} &= \text{MLP} \left(\text{BN} \left(h_v^{(t)} + \sum_{u \in N(v)} h_u^{(t)} \right) \right), t = \{0, \dots, T-1\} \\
 h &= \text{Concat} \left(\{h_v^{(t)}\}_{t=1}^T \right)
 \end{aligned} \tag{5.1}$$

La esencia aquí es que el modelo está utilizando información contextual de nodos vecinos en el grafo para enriquecer la representación de cada nodo, permitiendo una mejor segmentación y clasificación en tareas posteriores.

En la segmentación semántica supervisada, las representaciones contextualizadas de los nodos h_v , que corresponden a una imagen X , son procesadas por SegGini para asignar una etiqueta de clase a cada nodo. SegGini tiene la capacidad de incluir anotaciones múltiples, es decir, etiquetas de imagen inexactas Y_X y anotaciones incompletas Y_S . Luego, se asignan etiquetas Y_G al grafo y las etiquetas Y_V de los nodos se extraen de Y_S al asignar la clase más frecuente de cada nodo. Esto se basa en la idea de que las regiones del tejido son homogéneas en términos semánticos. Existen dos procesos llamados **Graph-head** ((Figura 5.9 (b)) y **Node-head** (Figura 5.9 (c)) que utilizan las etiquetas Y_G y Y_V respectivamente.

Dentro de la metodología de SegGini, nuestro enfoque se dirigirá hacia el proceso Graph-head, y más específicamente, al módulo de clasificación de grafos que contiene. Aunque el Graph-head también incluye un módulo de atribución de características, nuestra tarea no requiere de este aspecto. La razón de este enfoque selectivo es que, a diferencia del propósito integral de SegGini, nosotros no buscamos segmentar, sino clasificar de manera eficiente. Por lo tanto, adaptaremos y utilizaremos solo las partes de la metodología que se alineen con nuestro objetivo final.

El clasificador F_Φ predice Y_G para G . F_Φ incluye una operación de lectura promedio para producir un embedding de grafo de tamaño fijo h_G a partir de las representaciones de nodo h_v y un MLP para mapear h_G a Y_G . F_Θ y F_Ψ se entrena en un conjunto de grafos G con una función de pérdida adecuada.

Aunque no es fundamental para nuestra tarea, es interesante destacar que el módulo de atribución de características tiene la capacidad de determinar la relevancia de cada nodo en relación con la clasificación de cada clase. Este proceso se realiza mediante GraphGrad-CAM, que es una adaptación del conocido Grad-CAM, pero optimizado para trabajar con redes neuronales de grafos (GNN). En términos sencillos, esta herramienta nos permite visualizar y entender cuán significativos son ciertos nodos en el proceso de toma de decisiones del modelo. Es como tener una lupa que muestra qué partes del grafo son realmente esenciales para identificar una clase específica.

Para añadir un poco más de contexto, es útil mencionar la componente Node-head. Este módulo está diseñado para agilizar la segmentación de imágenes enfocándose en la clasificación de nodos. Funciona introduciendo h_v en un MLP y F_Ψ predice la etiqueta de cada nodo. De igual forma, F_Θ y F_Ψ se entrena en un conjunto de grafos G con una función de pérdida adecuada. Sin embargo, es importante enfatizar que nuestra atención se centra en otros

5.4. SEGGINI: CLASIFICACIÓN DE IMÁGENES COMPLETAS MEDIANTE GRAFOS DE TEJIDOS

aspectos del modelo, y la descripción del Node-head es meramente ilustrativa. Es una pieza interesante de la metodología, pero no una que necesitemos para nuestra tarea.

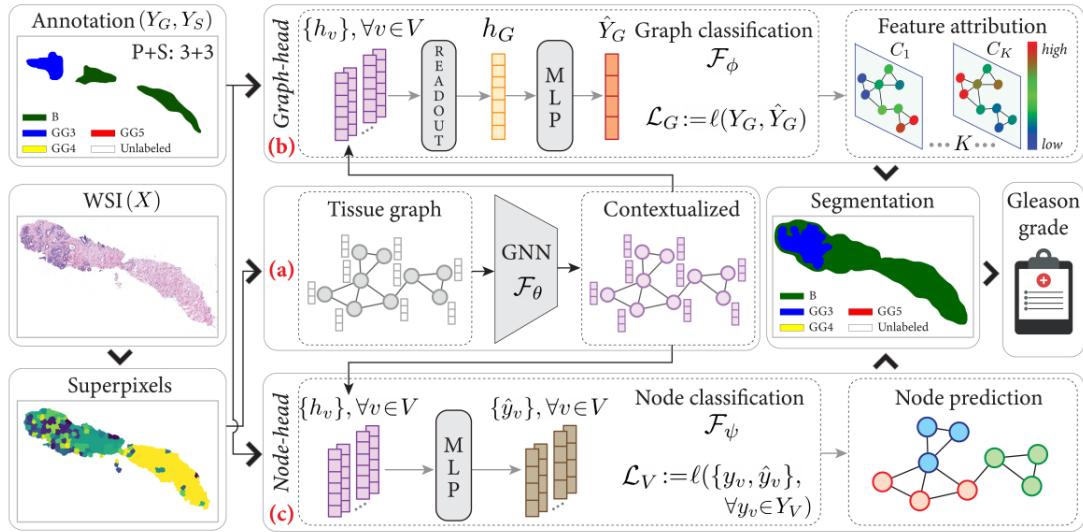


Figura 5.9: Resumen de la metodología **SegGini** (a) Construcción del grafo de tejido (b) Graph-head: WSS mediante clasificación de grafos (c) Node-head: WSS mediante clasificación de nodos (Fuente: [Anklin et al. \(2021\)](#))

Resultados y Discusión

6

Tras haber expuesto el marco teórico (capítulo 3), analizado el material (capítulo 4) y detallado la metodología (capítulo 5) ha llegado el momento de transformar ese conocimiento en aplicaciones prácticas y conclusiones. En este capítulo, detallaremos los descubrimientos que surgen de nuestro estudio sobre el uso de GNN para la detección de cáncer de próstata.

Antes de adentrarnos en los resultados, es fundamental entender el entorno y las herramientas que han hecho posible esta investigación. El proyecto ha sido desarrollado en python 3 y está disponible para su consulta en [Graph Neural Networks en cáncer de próstata](#). El cuaderno principal del proyecto, que expone la totalidad de la experimentación y análisis realizado, se encuentra en [GNN_SICAPv2.ipynb](#). En cuanto al entorno de ejecución, se ha optado por una máquina virtual de Google Colab T4 de alta capacidad de RAM con 51 GB de RAM de sistema y 15 GB de GPU.

A medida que avanzamos en este capítulo, es esencial aclarar que no abordaremos cada detalle minuciosamente, sino que nos centraremos en los aspectos más relevantes. Inicialmente, se llevan a cabo una serie de tareas preliminares. Estas tareas incluyen describir el problema en cuestión para comprender el propósito y la dirección del trabajo y acciones fundamentales como la descarga de SICAPv2 ([file_utils.py](#)). Posteriormente, generamos conjuntos de datos de imágenes que serán de gran importancia en nuestro análisis ([slide_dataset.py](#)). Una vez establecidos, se explora la base de datos y se procede a la construcción de WSI ([images_utils.py](#)).

Llegado a este punto, surge un dilema metodológico. Tenemos la posibilidad de trabajar directamente con los parches proporcionados en SICAPv2 o considerar la opción de generar nuestros propios parches a partir de la WSI reconstruida. Durante las primeras fases experimentales, observamos una superposición del 50 % entre los parches de SICAPv2. Esta superposición conduce a una redundancia de datos que compromete la capacidad del modelo GNN para generalizar de manera efectiva.

Tomando esto en cuenta, se optó por utilizar los parches originales exclusivamente para la reconstrucción de las imágenes WSI completas. Tras obtener estas WSI reconstruidas, se procedió a generar nuevos parches de dimensiones más reducidas. Esta decisión se basó en la necesidad de obtener imágenes más manejables. De esta manera, se registran resultados ligeramente mejorados, confirmando que la superposición de parches no ayuda a la generalización del modelo.

Tras superar el reto relacionado con los parches, emerge un nuevo desafío en la fase de

extracción de características ([feature_extractor.py](#)). Se presentan dos enfoques a la hora de obtener representaciones latentes: optar por redes preentrenadas o usar una arquitectura U-Net. A su vez, en la etapa de construcción de grafos, contamos con dos métodos: RAG y el enfoque que implica la conexión de todos los nodos ([graph_utils.py](#)). Tras determinar la estrategia de construcción, se genera el conjunto de grafos ([graph_dataset.py](#)). Este último paso es necesario para la posterior aplicación de modelos GNN ([models.py](#)), los cuales son puestos a prueba mediante funciones de entrenamiento ([train_model.py](#)) y evaluación ([test_model.py](#)). Estas funciones abordan las distintas tareas de clasificación y detección del cáncer en histologías de próstata.

Antes de la presentación de resultados, es fundamental abordar una limitación del conjunto de datos. SICAPv2 muestra un notorio desbalanceo, ciertas categorías apenas cuentan con 1, 4 o 7 imágenes. Por ello, surge la necesidad de explorar estrategias de aumento de grafos ([graph_augmentation.py](#)). Estas técnicas buscan diversificar el conjunto de entrenamiento mediante la creación de versiones modificadas de los grafos originales. El campo de graph augmentation es muy amplio y podría constituir otro tema de estudio. A pesar de su amplitud, se ha optado por métodos sencillos, tales como la introducción de ruido gaussiano y eliminación aleatoria de aristas, para ofrecer una primera aproximación al problema y evaluar si estos ajustes brindan una mejora. Con esto se pretende reconocer y abordar el problema intentando mitigar el sobreajuste que pueda surgir debido a ello. Señalar que esta implementación preliminar, es solo una pequeña fracción de un tema con muchas ramificaciones.

Tras haber delineado las distintas fases, es momento de transitar a la exposición y discusión de resultados obtenidos. El modo de proceder es considerar todas las combinaciones posibles entre enfoques, siempre y cuando tengan sentido, para ver qué resultados obtenemos.

Comenzamos diciendo que, dado que la metodología SegGini considera grafos de adyacencia de región, consideraremos este enfoque como el válido para la construcción de grafos. En la fase de obtención de representaciones latentes, el uso de redes preentrenadas es directo. En estas redes, la estrategia consiste en aplicar transfer learning suprimiendo las capas finales hasta llegar a la que nos interesa. En esta etapa, se llevaron a cabo pruebas con varias redes, como VGG, ResNet, MobileNet-v2 y ConvNext ([Medina \(2022\)](#)). Nuestra elección final recayó sobre ConvNext ya que esta red se preentrenó con un conjunto de imágenes específicas de próstata, a diferencia de las otras que fueron entrenadas con el conjunto de datos ImageNet. Esto no implica que las otras redes no sean eficaces; de hecho, son comúnmente citadas y utilizadas en la mayoría de los estudios en este campo ([David Ahmedt-Aristizabal \(2021\)](#)). Sin embargo, optamos por ConvNext en busca de aportar una perspectiva diferente y experimentar con ella. En términos generales, todas estas redes ofrecen resultados similares en el contexto de nuestro trabajo.

En relación a la arquitectura U-Net, hemos implementado un modelo basado en la sección [5.2.1](#), y más concretamente inspirado en la Figura [5.1](#). Este modelo, por defecto, sigue la configuración original de la U-Net ilustrada. Sin embargo, hemos introducido una mayor flexibilidad en su diseño que nos permite ajustar las dimensiones de las capas. Los resultados obtenidos al usar esta arquitectura son excelentes en términos de reconstrucción de imágenes. Este mode-

lo, con su configuración por defecto, nos brinda un espacio latente de dimensión $1024 \times 16 \times 16$, tanto para parches de SICAPv2 como para los generados, siempre y cuando tengan un tamaño de 256×256 . Para el entrenamiento de la U-Net, es interesante comentar que, a pesar de contar con más de 18000 parches, únicamente hemos requerido aproximadamente 1000 para entrenar el modelo satisfactoriamente. El proceso se llevó a cabo a lo largo de 15 épocas, utilizando un tamaño de lote de 16. Para optimizar, seleccionamos el algoritmo Adam con una tasa de aprendizaje de 0.001. Con el fin de medir la calidad de esta reconstrucción, además de evaluar la pérdida convencional, hemos empleado métricas como el PSNR (relación señal-ruido máxima) y el SSIM (índice de similitud estructural) (Figura 6.1).

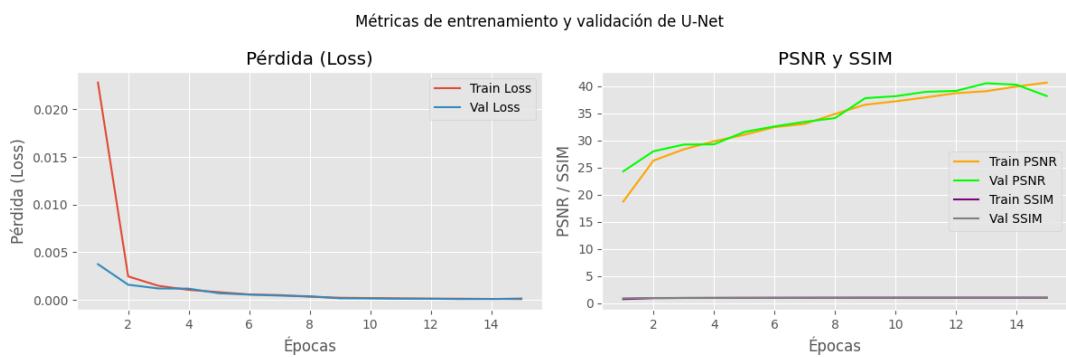


Figura 6.1: Entrenamiento y validación de la arquitectura U-Net para la obtención de representaciones latentes de parches de histologías (Fuente: Elaboración propia)

Para una mayor claridad en los resultados de la Figura 6.1, describiremos de manera sencilla y general las métricas PSNR y SSIM, sin entrar en tecnicismos. El **PSNR** compara la imagen original con la reconstruida, cuantificando el ruido o el error presente. Un PSNR más alto sugiere que la imagen reconstruida se asemeja a la original. A diferencia del PSNR, que se enfoca en los errores pixel a pixel, el **SSIM** evalúa cómo han cambiado estructuras, luminancia y contraste entre la imagen original y la reconstruida. Un valor cercano a 1 nos indica que las imágenes no solo se parecen en términos de color y brillo, sino que también conservan patrones y estructuras. La elección de estas métricas no es fortuita. Mientras el PSNR nos brinda una perspectiva cuantitativa sobre la calidad de la reconstrucción, el SSIM añade una visión cualitativa sobre posibles cambios en la estructura y apariencia. Ambas, en conjunto, garantizan una evaluación holística y confiable de nuestras imágenes reconstruidas.

Una vez mostrados los resultados del extractor de características, llega el momento de hablar de la implementación del modelo clasificador de grafos. Este se basa, en gran medida, en la metodología detallada en la sección 5.4. Este modelo pone su foco en lo que llamamos el proceso de graph head (Figura 5.9 (a) y (b)).

A continuación, comentaremos varios aspectos a tener en cuenta. Uno de los objetivos al implementar el graph head de SegGini ha sido proporcionar un alto grado de adaptabilidad. Esta configuración permite que, en lugar de tener que realizar cambios complejos en el código, se puedan modificar distintas funcionalidades simplemente ajustando determinados parámetros o activando ciertas banderas. Este modelo permite definir una dimensión de entrada. El

valor de la entrada vendrá determinado por la dimensión del espacio latente del extractor de características escogido. En caso de tener más de una dimensión, como ocurre con la U-Net ($1024 \times 16 \times 16$), será necesario un aplanamiento o pooling de la información. Existen parámetros y banderas tanto para esto último, como para el uso de normalización por lotes, dropout, número de capas GIN, número de capas de salida, número de neuronas por capa, ...

Otro de los aspectos destacados en nuestra implementación es la multifuncionalidad. Se ha diseñado el proceso de entrenamiento y validación para que sea versátil. Esto significa que podemos entrenar el modelo para predecir solo la puntuación de gleason primaria, solo el gleason secundario y ambos al mismo tiempo. También se pueden predecir los grados ISUP y detectar la presencia de cáncer. En cada tarea se buscan los mejores resultados, seleccionando el optimizador, tasa de aprendizaje y criterio de pérdida adecuados.

La versatilidad de nuestra implementación nos permite abordar múltiples tareas de clasificación, y debido a ello, presentaremos los resultados según la tarea específica llevada a cabo. Es razonable comenzar por una visión general, como es la detección de cáncer, y avanzar hacia aspectos más específicos, como la predicción de la puntuación ISUP o grados Gleason. La clasificación de grados Gleason es un proceso multifacético que puede abordarse desde varias perspectivas. Una de las formas es a través de la evaluación de la suma, donde, en lugar de diferenciar entre las etiquetas, consideramos su suma total como una única puntuación; por ejemplo, una combinación de '3+5' se clasificaría directamente con una puntuación de Gleason de 8. Otra forma distingue entre las etiquetas compuestas, donde '3+5' y '5+3' se tratan como categorías separadas debido a la primacía de cada cifra en la suma. Además, se pueden predecir de manera independiente las puntuaciones primaria y secundaria de Gleason. Este contexto induce a las tareas presentadas en la Tabla 6.1. Por último, comentar que para cada una de las tareas, se ha realizado una comparativa entre distintos enfoques y arquitecturas, incluyendo el uso de redes preentrenadas y la arquitectura U-Net para la extracción de características, seleccionando aquella configuración que mejores resultados arroje.

Para evitar redundancia en nuestra exposición y no reiterar los mismos detalles técnicos en cada tarea de clasificación, es importante subrayar desde ahora algunas decisiones clave. En la mayoría de experimentos, las redes preentrenadas se han utilizado junto con una operación de pooling promedio. En el caso de U-Net, hemos consolidado una representación latente de dimensiones $128 \times 16 \times 16$, que se aplana para su posterior análisis. La red preentrenada ConvNext ha destacado frente a los demás métodos de extracción de características. En cuanto al modelo GNN, se han empleado capas GIN, adaptando el número de capas ocultas, tasa de dropout y el uso de normalización por lotes. Estas configuraciones, ya establecidas, serán las que se mantengan en las siguientes tareas.

Para la evaluación del rendimiento de los modelos en las distintas tareas de clasificación, se han seleccionado meticulosamente métricas estandarizadas como la exactitud (accuracy), la precisión (precision), la sensibilidad (recall), la puntuación F1 y el valor de kappa de Cohen. La exactitud mide la proporción de predicciones correctas sobre el total, proporcionando una visión general del rendimiento. La precisión se enfoca en los verdaderos positivos frente a todas las predicciones positivas, siendo crucial para minimizar falsos positivos. El recall mide la

capacidad del modelo para identificar y clasificar correctamente los casos positivos. Esta métrica es crucial en situaciones en las que no detectar o clasificar un caso positivo correctamente tiene consecuencias significativas. La puntuación F1 es el promedio armónico de la precisión y el recall, brindando un equilibrio entre ambos en un solo número. El valor de kappa de Cohen (k), específicamente, ajusta la precisión considerando la posibilidad de que el acierto sea aleatorio, lo que ofrece una medida más robusta en escenarios desequilibrados. Utilizamos el kappa de Cohen no solo por su amplia aplicación en estudios médicos (Cohen (1968)), sino también para permitir comparaciones directas con resultados reportados en otras investigaciones (Silva-Rodríguez et al. (2020a), Arvaniti et al. (2018a)), facilitando la contextualización del nuestro trabajo en el campo.

En la Tabla 6.1 se muestran los mejores resultados para cada tarea:

Tabla 6.1: Métricas en las distintas tareas de clasificación

Tarea	Accuracy	Precision	Recall	F1	k
Detección de cáncer	0.9500	1.0000	0.9375	0.9677	0.8571
Clasificación de grados ISUP	0.6897	0.7060	0.6897	0.6631	0.5804
Clasificación de Gleason (suma)	0.6842	0.7595	0.6842	0.6454	0.6017
Clasificación de Gleason (categorías)	0.4828	0.7220	0.4828	0.4378	0.3983
Clasificación de Gleason primario	0.7368	0.7701	0.7368	0.7229	0.6192
Clasificación de Gleason secundario	0.5517	0.5414	0.5517	0.5338	0.3850

En la Figura 6.2 se observa el entrenamiento y validación para cada una de las tareas de la Tabla 6.1.

En Silva-Rodríguez et al. (2020a) se presenta una arquitectura basada en CNN para el diagnóstico del cáncer de próstata. El sistema se desarrolla para mejorar la precisión en la clasificación de la puntuación de Gleason en biopsias de próstata a nivel de parche, la detección de patrones cribriformes y la puntuación de Gleason de la biopsia completa. El núcleo de la metodología es un modelo CNN específicamente diseñado para este estudio, llamado FSConv. El modelo base consta de tres capas convolucionales seguidas de operaciones de agrupamiento máximo para la reducción de la dimensionalidad de las características extraídas.

De manera análoga a nuestro estudio, el enfoque innovador de Silva-Rodríguez et al. (2020a) reside en su enfoque de procesamiento de imagen dividido en parches. El artículo también aborda la comparación entre su modelo propuesto y otros enfoques de la literatura. El proceso para predecir la puntuación de Gleason en WSI requiere una clasificación de parche como primer paso. A continuación, por cada píxel, se calcula la probabilidad de pertenecer a cada clase Gleason basándose en las probabilidades de los parches más cercanos y se asigna la mayor. Una vez que el modelo reconstruye el mapa de probabilidad de la imagen completa, se utiliza un perceptrón multicapa (MLP) para la tarea de clasificación de la WSI.

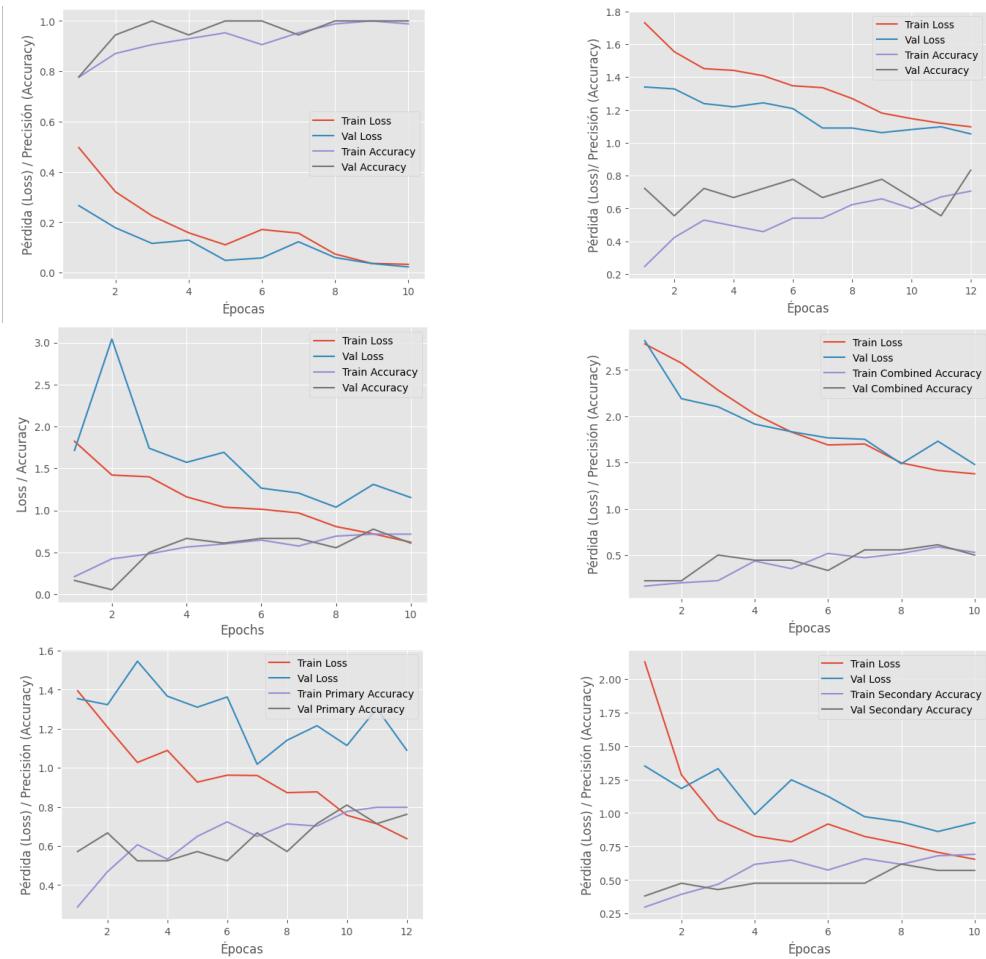


Figura 6.2: Métricas de entrenamiento y validación para (a) Detección de cáncer (b) Clasificación de grados ISUP (c) Clasificación de Gleason (suma) (d) Clasificación de Gleason (categorías) (e) Clasificación de Gleason primario (f) Clasificación de Gleason secundario

Continuando con la discusión, procederemos a examinar y comentar los distintos experimentos y resultados obtenidos en [Silva-Rodríguez et al. \(2020a\)](#). Posteriormente, estos resultados serán puestos en perspectiva con los alcanzados en nuestro trabajo, proporcionando así un terreno para la discusión y comparación. Antes de comenzar, mencionar que en los experimentos de [Silva-Rodríguez et al. \(2020a\)](#) se ha dividido la base de datos mediante validación cruzada para optimizar y probar algoritmos y se han empleado métricas como la precisión y el valor F1 para las distintas tareas.

En la tarea de clasificación de imágenes histológicas completas, [Silva-Rodríguez et al. \(2020a\)](#) proporciona una perspectiva detallada de la clasificación a nivel de parche, proceso que precede a la clasificación de WSI. Por ello, aunque nuestra investigación se centra en esta última tarea, resulta conveniente comentar los resultados de [Silva-Rodríguez et al. \(2020a\)](#) tanto a nivel de parche como de imagen completa.

Para la tarea de clasificación de puntuación de Gleason en parches, el estudio compara distintos enfoques y destaca uno frente al resto. La arquitectura FSConv propuesta junto con GMP (global max pooling) es la que alcanza un mayor desempeño. Este buen rendimiento se

atribuye a la reducción en el número de pesos a optimizar, simplificando el modelo y mejorando su capacidad de generalización para nuevas imágenes. Este modelo superó a otras arquitecturas como VGG19 y ResNet junto con GAP (global average pooling) y FC (fully-connected layers), mostrando más eficacia en la distinción entre grado de Gleason 3 y 4, tarea más desafiante para los patólogos, llegando a unos valores de precisión y F1-score de 0,7622 y 0,7125, respectivamente, en validación y 0,67 y 0,65, respectivamente, en evaluación. Además, este modelo fue validado en dos bases de datos externas para probar su capacidad de generalización. En [Arvaniti et al. \(2018b\)](#) se obtuvieron valores de 0,5861 y 0,5702 y en [Gertych et al. \(2015\)](#) de 0,5136 y 0,4753.

Una vez realizada la tarea a nivel de parche con el modelo FSConv, se generan mapas de probabilidad para cada grado Gleason, los cuales fueron validados cualitativamente en la práctica clínica por patólogos expertos. Se ilustraron casos de prueba donde el modelo identificó correctamente regiones de grados variados. Para la clasificación final, se empleó un MLP con los porcentajes de cada grado y se comparó el rendimiento mediante matrices de confusión y métrica de kappa de Cohen. El modelo se puso en práctica en la base de datos [Arvaniti et al. \(2018b\)](#) obteniendo un valor de kappa de Cohen de 0,8177. Los resultados obtenidos en validación fueron bastante similares, manteniendo un valor de kappa de 0,8168. Además, se han obtenido las métricas de la Tabla 6.1 a partir de la matriz de confusión de [Silva-Rodríguez et al. \(2020a\)](#) (Figura 18 (b)) donde toman los valores 0,5161, 0,6667, 0,70 y 0,6333, respectivamente.

Llegados a este punto, resulta oportuno contrastar los resultados obtenidos en nuestro trabajo con los de [Silva-Rodríguez et al. \(2020a\)](#). Mientras que [Silva-Rodríguez et al. \(2020a\)](#) se centra puramente en la clasificación de Gleason, nuestra labor extiende el espectro evaluativo a la detección de cáncer, clasificación de grados ISUP y clasificación de Gleason primario y secundario de manera independiente. Esta ampliación, podría enriquecer el corpus de conocimientos existente con nuevas perspectivas en la detección y prognosis del cáncer de próstata.

En esta línea, es gratificante informar que en la tarea de detección de cáncer, los valores altos en las métricas de evaluación empleadas reflejan una alta efectividad en su identificación. En lo que respecta a la clasificación de grados ISUP, los resultados obtenidos revelan una correspondencia estrecha con la clasificación de Gleason por suma, lo que sugiere que las similitudes inherentes entre estas categorizaciones pueden ser aprovechadas para afinar aún más las estrategias de clasificación. Por otro lado, es alentador observar que nuestro modelo predice con una mayor efectividad los grados de Gleason primarios frente a los secundarios. Este hallazgo abre la puerta a exploraciones futuras enfocadas en desentrañar y mejorar el rendimiento en la clasificación del grado Gleason secundario. Entender las razones que hay detrás del empeoramiento de esta clasificación puede ser crucial para desarrollar metodologías más afinadas. Estos esfuerzos podrían allanar el camino hacia una predicción más acertada del cómputo global de Gleason.

Centrándonos en la clasificación de Gleason por suma, la cual es directamente comparable con [Silva-Rodríguez et al. \(2020a\)](#), encontramos que las métricas de exactitud, precisión, sensibilidad, valor F1 muestran una congruencia notable con sus hallazgos. Sin embargo, el punto de divergencia surge al considerar el valor kappa de Cohen. A pesar de que nuestros

resultados reflejan un desempeño competente, el valor kappa obtenido es modestamente inferior al reportado en el estudio de referencia. Cabe señalar que esta comparación se interpreta de forma selectiva ya que en [Silva-Rodríguez et al. \(2020a\)](#) se evalua el rendimiento del modelo con base de datos externas ([Arvaniti et al. \(2018b\)](#)) y nuestra evaluación se realiza con SICAPv2.

En cuanto a la tarea de clasificación de Gleason por categorías, los resultados obtenidos se alinean con las expectativas previas, considerando las peculiaridades inherentes de los datos manejados ([Anklin et al. \(2021\)](#)). Un factor significativo que contribuye al valor más bajo observado en algunas métricas, es el desequilibrio en la distribución de las categorías en el conjunto de datos. Algunas categorías están menos representadas que otras, lo que puede sesgar el aprendizaje del modelo y afectar su capacidad para generalizar. Cuando la clasificación se realiza agrupando categorías bajo ISUP o mediante la suma de grados Gleason, cada categoría adquiere una mayor representación. Esto facilita el proceso de aprendizaje, permitiendo una mejor identificación de las características distintivas de cada grupo y, como resultado, mejora el rendimiento global. Esta distribución más equilibrada, señala la importancia de considerar más representatividad en cada categoría.

Concluimos nuestro análisis con la certeza de que nuestra investigación aporta valor al campo de clasificación de cáncer de próstata. Este trabajo no solo replica los hallazgos anteriores mediante una metodología basada en redes neuronales de grafos, sino que también los extiende al introducir resultados en tareas previamente no abordadas en SICAPv2 como la detección de cáncer y predicción de ISUP y Gleason de manera individualizada. A pesar de los retos presentados, el modelo ha demostrado cierta capacidad para abordar distintas tareas, sentando así las bases para futuras investigaciones que continuarán refinando la precisión y generalización de las técnicas de diagnóstico.

Conclusiones

7

A lo largo de este trabajo, hemos abordado la tarea de clasificar imágenes histológicas para la detección y gradación de cáncer de próstata. Se ha evaluado el desempeño de diferentes metodologías, ajustando sus parámetros para adaptarse a distintas tareas. Ahora, estableceremos las conclusiones derivadas de los resultados obtenidos, ofreciendo una síntesis de las principales lecciones aprendidas y reflexiones.

Basándonos en el estudio realizado, nos proponemos a delinean las conclusiones clave que emergen de este trabajo. Estas conclusiones no solo sirven como resumen de los aspectos principales del trabajo, sino que también se presentan como una verificación directa del cumplimiento de nuestros objetivos iniciales, tal como se definieron en el capítulo 1. Así, nos preparamos para presentar una serie de conclusiones fundamentadas:

1. Se ha realizado una comparativa básica entre las redes neuronales convolucionales (CNN) y las redes neuronales de grafos (GNN) mediante un recorrido por el estado del arte y explicando su evolución. Destacamos que, aunque las CNN son extremadamente eficientes para procesar imágenes, gracias a su habilidad para detectar patrones y características visuales, las GNN se presentan como una alternativa potente al ser capaces de capturar relaciones y estructuras complejas entre los datos. En nuestro caso particular, se tiene la ventaja de poder reconstruir la imagen completa. Sin embargo, si tenemos en cuenta escenarios donde esta reconstrucción no es viable, las GNN resaltan por su flexibilidad y robustez para trabajar directamente con información estructural.
2. Se ha proporcionado un marco teórico sobre la teoría de grafos y las graph neural networks tratando de resaltar la relevancia de los grafos en el ámbito del deep learning. Se ha demostrado su importancia en la representación de relaciones.
3. En el presente estudio, se ha profundizado en el área de la patología digital y la histopatología, analizando la manera en que los métodos basados en grafos pueden ser utilizados en tareas de clasificación. La metodología que se ha seguido, englobando etapas como la detección de entidades, obtención de representaciones latentes y construcción de grafos, se ha mostrado capaz de afrontar los retos particulares que plantea este campo.
4. SICAPv2 ha sido esencial para nuestro estudio, al ser una colección de biopsias de próstata con anotaciones tanto a nivel de parche como de imagen completa. Ofrece ventajas significativas que ayudan a que se cumplan los objetivos del trabajo.

-
- 5. Se ha prestado especial atención a la representación de parches en nodos, explorando el uso de métodos basados en U-Net y redes preentrenadas. Se destaca la importancia de contar con un modelo entrenado con datos específicos del ámbito de aplicación, enfatizando cómo esto puede ser un factor determinante para el éxito de un modelo. Además, el enfoque RAG y la construcción de grafos con conexiones entre todos los nodos ofrecen ventajas e inconvenientes a examinar.
 - 6. La metodología SegGINI, a pesar de su aparente enfoque en segmentación, ha demostrado poseer una serie de componentes adaptables para tareas de clasificación de imágenes histológicas. Cada etapa del proceso, encaja de manera cohesiva con una metodología aparentemente más genérica. Más allá de su propósito, SegGINI ofrece un enfoque estructurado y bien definido que puede ser extrapolado y utilizado en otros contextos, obteniendo un modo útil de proceder.
 - 7. Se ha llevado a cabo una evaluación empírica de los modelos propuestos, extrayendo conclusiones fundamentadas sobre su desempeño y aplicabilidad. Esta evaluación ha permitido identificar áreas de fortaleza y oportunidades de mejora, proporcionando una visión clara de los pasos a seguir para optimizar futuras implementaciones.
 - 8. En cuanto a los resultados obtenidos, se ha proporcionado una discusión, destacando las configuraciones que han demostrado ser más efectivas. Un análisis más detallado de los resultados se encuentran en el capítulo 6.

Limitaciones y Perspectivas de Futuro

8

A medida que presentamos las conclusiones, es esencial reconocer que toda investigación lleva consigo ciertas limitaciones y, al mismo tiempo, abre la puerta a un horizonte de posibilidades futuras. En este capítulo, exploraremos las limitaciones que nos han acompañado, ofreciendo una visión transparente de las restricciones. Al mismo tiempo, consideramos las perspectivas de futuro como oportunidades para la expansión y enriquecimiento en este campo.

Cabe señalar que, si bien no hemos enfrentado limitaciones que impidieran la realización del proyecto, existen aspectos que merecen ser considerados. Estas limitaciones han influido en ciertos aspectos y pueden proporcionar información valiosa para el futuro:

1. Una limitación se relaciona con la capacidad de memoria de la máquina utilizada. Aunque esta ofrece recursos considerables, el procesamiento de imágenes y la generación de múltiples representaciones latentes ejercen una fuerte demanda. Ha sido necesario tener esto en cuenta para adaptar las operaciones y limitar la elección de modelos para evitar que la memoria se agote.
2. A pesar de que la base de datos consta de una cantidad enorme de parches, que resultan útiles, la cantidad de imágenes histológicas completas es limitada, con solo 155 muestras. Además, se distribuye de manera desigual entre las distintas categorías. Esta cantidad y desbalanceo pueden haber restringido la capacidad de los modelos.
3. Un aspecto digno de mención, aunque no es necesariamente una limitación, es la necesidad de que las características de los nodos estén en una dimensión. En el caso de las imágenes, esto implica que las representaciones latentes se tengan que aplanar o aplicar convoluciones y pooling hasta reducirlas. Esta transformación puede resultar en la pérdida de información valiosa y la estructura intrínseca de la imagen. La calidad de las características se ve afectada por este proceso, y es un punto crítico que influye en el éxito de un modelo.
4. Un condicionamiento importante radica en la escasez de referencias específicas relacionadas con el cáncer de próstata en nuestra base de datos. Aunque existe una amplia gama de métodos y referencias en el ámbito general de la histopatología, encontramos una falta de investigaciones centradas en nuestro conjunto de datos particular y nuestros objetivos específicos. Esta restricción nos lleva a basar este trabajo en una metodología

específica, ya que no hay suficiente evidencia para respaldar la exploración de otras opciones.

Una perspectiva clave para el futuro es la ampliación de la base de datos. Además de buscar la adquisición de un mayor número de imágenes histológicas de cáncer de próstata, se puede explorar la generación de muestras de alta calidad mediante técnicas de aumento de grafos (graph augmentation). Esta estrategia no solo se limitaría a crear nuevos grafos modificando la estructura o características del grafo original, lo que contribuiría a fortalecer la robustez y a mejorar su capacidad de detección y clasificación. Esta área podría desempeñar un papel fundamental en el enriquecimiento de la base de datos y, en última instancia, en el éxito de futuras investigaciones.

Otra dirección prometedora es la exploración de enfoques alternativos para la extracción de características en lugar de centrarse exclusivamente en representaciones latentes. Es esencial considerar otras técnicas que puedan enriquecer aún más la información contenida en los grafos. Además, se podría analizar cómo otros enfoques exitosos en el campo de la histopatología ([David Ahmedt-Aristizabal \(2021\)](#)) pueden aplicarse o adaptarse a nuestra tarea. De esta manera, se tendría en cuenta la incorporación de otros filtros como graph convolutional networks (GCN) o GraphSAGE para procesar la estructura e información del grafo. Por otro lado, sería valioso indagar en enfoques de construcción de grafos ya que tienen un impacto significativo en el rendimiento.

Por último, comentar que otro trabajo futuro consiste en considerar la incorporación de graph explainers, herramientas que proporcionan una mayor comprensión y explicabilidad de los modelos. Estos permiten identificar las zonas de los grafos en las que los modelos se han centrado. Además, sirven como herramienta de comprobación para garantizar que la metodología se está llevando a cabo correctamente.

Lista de Acrónimos

BN *Batch Normalization.*

CNN *Convolutional Neural Network.*

CPC *Contrastive Predictive Coding.*

CV *Computer Vision.*

DL *Deep Learning.*

GAT *Graph Attention Network.*

GCN *Graph Convolutional Network.*

GIN *Graph Isomorphism Network.*

GNN *Graph Neural Network.*

H&E *Hematoxylin and Eosin stain.*

ISUP *International Society of Urological Pathology.*

MLP *Multilayer Perceptron.*

PSNR *Peak Signal-to-Noise Ratio.*

RAG *Regional Adjacency Graph.*

SSIM *Structural Similarity Index Measure.*

TG *Tissue Graph.*

TMA *Tissue Microarrays.*

VAE *Variational Autoencoder.*

WSI *Whole Slide Image.*

WSS *Weakly-supervised Semantic Segmentation.*

Bibliografía

- Cancer prevention organisation. <https://www.wcrf.org/>. Último acceso: 2023-10.
- Global cancer observatory. <https://gco.iarc.fr/>. Último acceso: 2023-10.
- Anklin, V., Pati, P., Jaume, G., Bozorgtabar, B., Foncubierta-Rodriguez, A., Thiran, J.-P., Sibony, M., Gabrani, M., y Goksel, O. (2021). Learning whole-slide segmentation from inexact and incomplete labels using tissue graphs. In *Medical Image Computing and Computer Assisted Intervention – MICCAI 2021*, pages 636–646. Springer International Publishing, Cham.
- Arvaniti, E., Fricker, K. S., Moret, M., Rupp, N., Hermanns, T., Fankhauser, C., Wey, N., Wild, P. J., Rüschhoff, J. H., y Claassen, M. (2018a). Automated gleason grading of prostate cancer tissue microarrays via deep learning. *Sci. Rep.*, 8(1):12054.
- Arvaniti, E., Fricker, K. S., Moret, M., Rupp, N., Hermanns, T., Fankhauser, C., Wey, N., Wild, P. J., Rüschhoff, J. H., y Claassen, M. (2018b). Automated gleason grading of prostate cancer tissue microarrays via deep learning. *Sci. Rep.*, 8(1):12054.
- Chan, L., Hosseini, M. S., y Plataniotis, K. N. (2021). A comprehensive analysis of weakly-supervised semantic segmentation in different image domains. *Int. J. Comput. Vis.*, 129(2):361–384.
- Chen, Y., Rohrbach, M., Yan, Z., Shuicheng, Y., Feng, J., y Kalantidis, Y. (2019). Graph-based global reasoning networks. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE.
- Cohen, J. (1968). Weighted kappa: Nominal scale agreement provision for scaled disagreement or partial credit. *Psychol. Bull.*, 70(4):213–220.
- David Ahmedt-Aristizabal, Mohammad Ali Armin, S. D. C. F. L. P. (2021). A survey on graph-based deep learning for computational histopathology.
- Defferrard, M., Bresson, X., y Vandergheynst, P. (2016). Convolutional neural networks on graphs with fast localized spectral filtering. In *Proc*, pages 3844–3852.
- Deng, S., Zhang, X., Yan, W., Chang, E. I.-C., Fan, Y., Lai, M., y Xu, Y. (2020). Deep learning in digital pathology image analysis: a survey. *Front. Med.*, 14(4):470–487.
- Doyle, S., Hwang, M., Shah, K., Madabhushi, A., Feldman, M., y Tomaszewski, J. (2007). Automated grading of prostate cancer using architectural and textural image features. In *2007 4th IEEE International Symposium on Biomedical Imaging: From Nano to Macro*. IEEE.
- Epstein, J. I., Egevad, L., Amin, M. B., Delahunt, B., Sri-gley, J. R., Humphrey, P. A., y Grading Committee (2016). The 2014 international society of urological pathology (ISUP) consensus conference on gleason grading of prostatic carcinoma: Definition of grading patterns and proposal for a new grading system. *Am. J. Surg. Pathol.*, 40(2):244–252.
- Esteban, A. E., López-Pérez, M., Colomer, A., Sales, M. A., Molina, R., y Naranjo, V. (2019). A new optical density granulometry-based descriptor for the classification of prostate histological images using shallow and deep gaussian processes. *Computer Methods and Programs in Biomedicine*, 178:303–317.
- Georgousis, S., Kenning, M. P., y Xie, X. (2021). Graph deep learning: State of the art and challenges. *IEEE Access*, 9:22106–22140.
- Gertych, A., Ing, N., Ma, Z., Fuchs, T. J., Salman, S., Mohanty, S., Bhele, S., Velásquez-Vacca, A., Amin, M. B., y Knudsen, B. S. (2015). Machine learning approaches to analyze histological images of tissues from radical prostatectomies. *Comput. Med. Imaging Graph.*, 46 Pt 2:197–208.
- Gilmer, J., Schoenholz, S. S., Riley, P. F., Vinyals, O., y Dahl, G. E. (2017). Neural message passing for quantum chemistry.
- Gleason, D. F. (1992). Histologic grading of prostate cancer: A perspective. *Hum. Pathol.*, 23(3):273–279.
- Gordetsky, J. y Epstein, J. (2016). Grading of prostatic adenocarcinoma: current state and prognostic implications. *Diagn. Pathol.*, 11(1):25.

- Hamilton, W. L. (2020). Graph representation learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 14(3):1–159.
- Hamilton, W. L., Ying, R., y Leskovec, J. (2017a). Inductive representation learning on large graphs.
- Hamilton, W. L., Ying, R., y Leskovec, J. (2017b). Representation learning on graphs: Methods and applications.
- Kipf, T. N. y Welling, M. (2016). Semi-supervised classification with graph convolutional networks.
- Li, J., Xie, X., Zhao, Z., Cao, Y., Pan, Q., y Shi, G. (2020). Temporal graph modeling for skeleton-based action recognition.
- Long, J., Shelhamer, E., y Darrell, T. (2014). Fully convolutional networks for semantic segmentation.
- Lucas, M., Jansen, I., Savci-Heijink, C. D., Meijer, S. L., de Boer, O. J., van Leeuwen, T. G., de Bruin, D. M., y Marquering, H. A. (2019). Deep learning for automatic gleason pattern classification for grade group determination of prostate biopsies. *Virchows Arch.*, 475(1):77–83.
- Ma, Y. y Tang, J. (2021). *Deep Learning on Graphs*. Cambridge University Press.
- Medina, S. (2022). PANDA_ConvNeXT_K - hugging face. https://huggingface.co/smc/PANDA_ConvNeXT_K. Último acceso: 2023-10.
- Nir, G., Hor, S., Karimi, D., Fazli, L., Skinnider, B. F., Tavassoli, P., Turbin, D., Villamil, C. F., Wang, G., Wilson, R. S., Iczkowski, K. A., Lucia, M. S., Black, P. C., Abolmaesumi, P., Goldenberg, S. L., y Salcudean, S. E. (2018). Automatic grading of prostate cancer in digitized histopathology images: Learning from multiple experts. *Med. Image Anal.*, 50:167–180.
- Nir, G., Karimi, D., Goldenberg, S. L., Fazli, L., Skinnider, B. F., Tavassoli, P., Turbin, D., Villamil, C. F., Wang, G., Thompson, D. J. S., Black, P. C., y Salcudean, S. E. (2019). Comparison of artificial intelligence techniques to evaluate performance of a classifier for automatic grading of prostate cancer from digitized histopathologic images. *JAMA Netw. Open*, 2(3):e190442.
- Potjer, F. K. (1996). Region adjacency graphs and connected morphological operators. In *Mathematical Morphology and its Applications to Image and Signal Processing*, pages 111–118. Springer US, Boston, MA.
- Qi, S., Wang, W., Jia, B., Shen, J., y Zhu, S.-C. (2018). Learning human-object interactions by graph parsing neural networks. In *Computer Vision – ECCV 2018*, pages 407–423. Springer International Publishing, Cham.
- Remotti, H. (2013). Tissue microarrays: Construction and use, methods in molecular biology. *Methods in Molecular Biology*.
- Ren, J., Sadimin, E., Foran, D. J., y Qi, X. (2017). Computer aided analysis of prostate histopathology images to support a refined gleason grading system, medical imaging 2017. *Image Processing*, 10133.
- Ronneberger, O., Fischer, P., y Brox, T. (2015). U-net: Convolutional net-works for biomedical image segmentation. In *Proc. Med*, pages 234–241.
- Silva-Rodríguez, J., Colomer, A., Sales, M. A., Molina, R., y Naranjo, V. (2020a). Going deeper through the gleason scoring scale: An automatic end-to-end system for histology prostate grading and cribriform pattern detection. *Comput. Methods Programs Biomed.*
- Silva-Rodríguez, J., Colomer, A., Sales, M. A., Molina, R., y Naranjo, V. (2020b). SICAPv2 - prostate whole slide images with gleason grades annotations. <https://data.mendeley.com/datasets/9xxm58dvs3/1>.
- Srinidhi, C. L., Ciga, O., y Martel, A. L. (2021). Deep neural network models for computational histopathology: A survey. *Med. Image Anal.*, 67(101813):101813.
- Toro, O., Atzori, M., Otalora, S., Andersson, M., Euren, K., Hedlund, M., Ronquist, P., y Muller, H. (2017). Convolutional neural networks for an automatic classification of prostate tissue slides with high-grade gleason score, medical imaging 2017. *Digital Pathology*, 10140.
- Vahadane, A., Peng, T., Sethi, A., Albarqouni, S., Wang, L., Baust, M., Steiger, K., Schlitter, A. M., Esposito, I., y Navab, N. (2016). Structure-preserving color normalization and sparse stain separation for histological images. *IEEE Trans. Med. Imaging*, 35(8):1962–1971.
- van der Laak, J., Litjens, G., y Ciompi, F. (2021). Deep learning in histopathology: the path to the clinic. *Nat. Med.*, 27(5):775–784.
- Velickovic, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., y Bengio, Y. (2018). Graph attention networks. In *Proc. Int. Conf. Learn. Repr. (ICLR)*.
- Weinstein, J. N., Collisson, E. A., Mills, G. B., Shaw, M., Ozenberger, B. A., Ellrott, K., Shmulevich, I., Sander, C., y Stuart, J. M. (2013). The cancer genome atlas Pan-Cancer analysis project. *NIH Public Access*, 4518:219–223.
- Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., y Yu, P. S. (2021). A comprehensive survey on graph neural networks. *IEEE Trans. Neural Netw. Learn. Syst.*, 32(1):4–24.
- Xu, K., Hu, W., Leskovec, J., y Jegelka, S. (2018). How powerful are graph neural networks?

Lista de Acrónimos

You, J. Cs224w. <https://web.stanford.edu/class/cs224w>. Último acceso: 2023-09.

Zhou, J., Cui, G., Hu, S., Zhang, Z., Yang, C., Liu, Z., Wang, L., Li, C., y Sun, M. (2020). Graph neural networks: A review of methods and applications. *AI Open*, 1:57–81.