



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



ESCUELA TÉCNICA
SUPERIOR INGENIEROS
INDUSTRIALES VALENCIA

TRABAJO FIN DE MASTER EN INGENIERÍA BIOMEDICA

**DISEÑO Y DESARROLLO DE UN SISTEMA
AUTOMÁTICO DE SEGMENTACIÓN DE
GLÁNDULAS HISTOLÓGICAS PARA
IDENTIFICAR EL CÁNCER DE PRÓSTATA EN
UNA ETAPA INICIAL.**

AUTORA: ELENA PAYÁ BOSCH

TUTORA: VALERY NARANJO ORNEDO

COTUTOR: JOSÉ GABRIEL GARCÍA PARDO

Curso Académico: 2018-19

Resumen

El presente trabajo fin de máster pretende abordar una de las fases intermedias de un proyecto nacional de mayor envergadura cuyo último objetivo es proporcionar una herramienta a los especialistas en anatomía patológica que actúe a modo de sistema de ayuda para el diagnóstico temprano del cáncer de próstata. En este TFM se lleva a cabo el diseño y desarrollo de un método de segmentación automática de glándulas de próstata mediante técnicas de *deep learning*.

Para conseguir el objetivo propuesto, se lleva a cabo una exhaustiva revisión del estado del arte y se desarrolla un método que permite realizar una segmentación glandular basada en arquitecturas encoder-decoder con neuronas convolucionales. La arquitectura diseñada permite a partir de la imagen completa obtener un mapa de probabilidad del mismo tamaño donde las glándulas se encuentran marcadas. Para el desarrollo del algoritmo de segmentación se ha contado con un total de 29 pacientes que aportan imágenes tanto de tejido sano como de tejido patológico de grado 3.

A fin de llevar a cabo el entrenamiento del modelo propuesto se ha implementado un enfoque supervisado y, para ello, se ha realizado una segmentación manual en la que se han generado las etiquetas correspondientes a cada píxel. Además, se ha desarrollado una partición de los datos y un balanceo con la finalidad de obtener resultados fiables y dar robustez al modelo propuesto.

Finalmente, los resultados obtenidos se han comparado con la segmentación manual realizada y se han calculado métricas que permiten evaluar la bondad del método propuesto. Asimismo, estos indicadores se han comparado en la medida de lo posible con los estudios del estado del arte. En conclusión, se ha obtenido un modelo de segmentación automática de estructuras glandulares muy prometedor.

Palabras Clave: cáncer de próstata, imágenes histológicas, segmentación automática, *deep learning*, red neuronal convolucional, *encoder-decoder*.

Resum

El present treball fi de màster pretén abordar una de les fases intermèdies d'un projecte nacional de major embergadura l'últim objectiu del qual és proporcionar una ferramenta als especialistes en anatomia patològica que actue a manera de sistema d'ajuda per al diagnòstic primerenc del càncer de pròstata. En aquest TFM es du a terme el disseny i desenvolupament d'un mètode de segmentació automàtica de glàndules de pròstata mitjançant tècniques de *deep learning*.

Per a aconseguir l'objectiu proposat, es du a terme una exhaustiva revisió de l'estat de l'art i es desenvolupa un mètode que permet realitzar una segmentació glandular basada en arquitectures *encoder-decoder* amb neurones convolucionals. La arquitectura dissenyada permet a partir de la imatge completa obtenir un mapa de probabilitat del mateix grandària on les glàndules es troben marcades. Per al desenvolupament de l'algoritme de segmentació s'ha comptat amb un total de 29 pacients que aporten imatges tant de teixit sa com de teixit patològic de grau 3.

A fi de dur a terme l'entrenament del model proposat s'ha implementat un enfocament supervisat i, per a això, s'ha realitzat una segmentació manual en la qual s'han generat les etiquetes corresponents a cada píxel. A més, s'ha desenvolupat una partició de les dades i un balanceig amb la finalitat d'obtenir resultats fiables i donar robustesa al model proposat.

Finalment, els resultats obtinguts s'han comparat amb la segmentació manual realitzada i s'han calculat mètriques que permeten avaluar la bondat del mètode proposat. Així mateix, aquests indicadors s'han comparat en la mesura que siga possible amb els estudis de l'estat de l'art. En conclusió, s'ha obtingut un model de segmentació automàtica d'estructures glandulars molt prometedora.

Paraules clau: càncer de pròstata, imatges histològiques, segmentació automàtica, *deep learning*, xarxa neuronal convolucional, *encoder-decoder*.

Abstract

The current master's thesis aims to approach one of the intermediate phases of a larger national project whose ultimate goal is to provide a tool for pathology specialists that acts as an aid system for the early diagnosis of prostate cancer. In this thesis, the design and development of an automatic segmentation method of prostate glands is carried out using deep learning techniques.

To achieve the proposed objective, a thorough review of the state of the art is accomplished and a method that allows glandular segmentation based on encoder-decoder architectures with convolutional neurons is developed. The designed architecture allows, from the complete image, to obtain a probability map of the same size where the glands are marked. For the development of the segmentation algorithm, a total of 29 patients who provide images of both healthy tissue and grade 3 pathological tissue were available.

In order to carry out the training of the proposed model, a supervised approach has been implemented and, to this end, it is necessary to do a manual segmentation in which the labels corresponding to each pixel have been generated. In addition, a data splitting and balancing has been developed in order to obtain reliable results and provide robustness to the proposed model.

Finally, the results obtained have been compared with the manual segmentation performed and metrics that allow to evaluate the goodness of the proposed method have been calculated. Likewise, these indicators have been compared as far as possible with studies of the state of the art. In conclusion, a very promising automatic segmentation model of glandular structures has been obtained.

Keywords: prostate cancer, histological images, automatic segmentation, deep learning, convolutional neural network, encoder-decoder.

Índice general

Resumen	III
Índice general	IX
I Memoria	1
1 Introducción	3
1.1 Motivación y descripción del problema	3
1.1.1 Cáncer y cáncer de próstata	3
1.1.1.1 Cáncer	3
1.1.1.2 Cáncer de próstata	5
1.1.2 Imagen histológica prostática	8
1.1.3 Escala de Gleason	9
1.2 Proyecto SICAP	12
1.3 Objetivos	12
1.4 Guía de la memoria	13
2 Marco teórico	15
2.1 Introducción al deep learning en segmentación	15
2.2 Fundamento teórico red neuronal	17
2.2.1 Estructura básica red neuronal	17
2.2.2 Algoritmo forward-backward propagation	19
2.3 Redes neuronales convolucionales	21
2.3.1 Capa convolucional	21
2.3.2 Capa activación	23
2.3.3 Capa <i>Pooling</i>	26
2.3.4 Capa <i>Dropout</i>	27
2.3.5 Arquitectura convolucional	27
2.4 Arquitectura Encoder-Decoder	28

3 Estado del arte	31
4 Material y método	35
4.1 Material	35
4.1.1 Base de datos	35
4.1.2 Software y hardware	38
4.1.2.1 Software	38
4.1.2.2 Hardware	38
4.2 Metodología	39
4.2.1 Creación Ground truth	39
4.2.2 Preparación de la base de datos de imágenes	40
4.2.3 Partición datos	42
4.2.4 Entrenamiento	43
4.2.4.1 Arquitectura del modelo	43
4.2.4.2 Coeficiente Dice como función de pérdida	46
4.2.4.3 Hiperparámetros	46
4.2.4.4 Optimizadores	48
4.2.5 Reconstrucción de las imágenes	49
4.2.6 Evaluación	49
5 Resultados de la segmentación	51
6 Conclusiones y líneas futuras	57
6.1 Conclusiones	57
6.2 Líneas futuras	58
II Presupuesto	61
7 Presupuesto	63
7.1 Presupuestos parciales	63
7.1.1 Coste mano de obra	63
7.1.2 Coste maquinaria	64
7.1.3 Coste materiales	65
7.2 Presupuesto total	66
Bibliografía	67

Parte I

Memoria

Introducción

1.1 Motivación y descripción del problema

1.1.1 *Cáncer y cáncer de próstata*

1.1.1.1 *Cáncer*

Cáncer es el término empleado para definir un conjunto de enfermedades que pueden afectar a cualquier parte del organismo. Esencialmente se trata de una enfermedad de división celular incontrolada y, puesto que el organismo está compuesto por billones de células, la transformación de células normales en células tumorales puede originarse en casi cualquier parte de él [1].

Este término genérico hace referencia a la multiplicación rápida y descontrolada de células anormales que se extienden más allá de sus límites habituales y que persiste tras finalizar el estímulo origen. Su desarrollo suele estar relacionado con una serie de alteraciones en la actividad de los reguladores del ciclo celular, normalmente debida a mutaciones en los genes que codifican estas proteínas. Estas alteraciones son el resultado de los factores genéticos y los agentes externos que se listan a continuación [2]:

- Carcinógenos físicos, como radiaciones ultravioletas o ionizantes.
- Carcinógenos químicos, como el amianto o el humo de tabaco.
- Carcinógenos biológicos, como virus o bacterias.

El cáncer tiene grandes repercusiones a nivel mundial, las estadísticas ofrecen una imagen a través del tiempo de la carga que supone esta enfermedad tal y como puede observarse en la gráfica de la Figura 1.1. Es la segunda causa de muerte en el mundo y se estima que en 2015 ocasionó 8,8 millones de muertes. De acuerdo con los últimos datos disponibles dentro del proyecto

GLOBOCAN¹, el número de tumores sigue aumentando y se estima que en 2040 alcance los 29,5 millones.

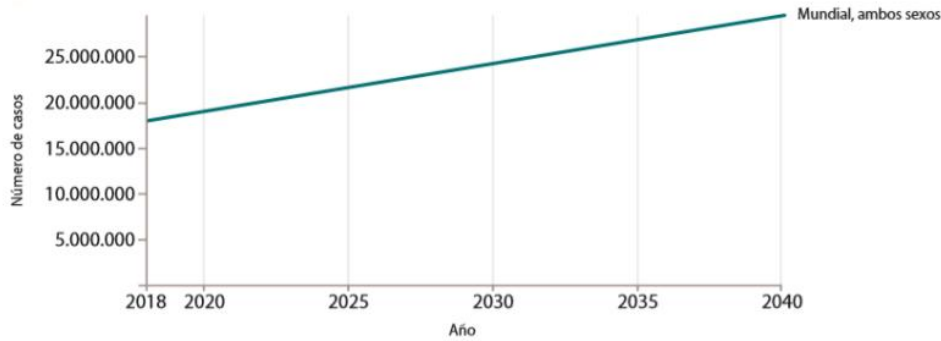


Figura 1.1: Incidencia estimada de tumores en la población mundial [4].

La Sociedad Española de Oncología Médica (SEOM) presenta en sus informes epidemiológicos datos alarmantes sobre la enfermedad del cáncer en España. Las posibilidades de desarrollar un tumor varían en función de la edad y del sexo, incrementando considerablemente el riesgo para hombres mayores de 65 años, como se puede ver en la Tabla 1.1. Se cree que la acumulación general de factores de riesgo y la pérdida de eficacia de los mecanismos de reparación celular convierten el envejecimiento poblacional en una de las causas más importantes.

	Hombres	Mujeres	Ambos sexos
<65 años	54.148	54.223	108.371
>65	106.916	61.947	168.863

Tabla 1.1: Estimación del número de nuevos casos en España para el año 2019 [4].

Los factores de riesgo varían en función del país, pero cabe destacar que los principales a nivel mundial son: el consumo de tabaco, el alcohol, la mala alimentación y la inactividad física. En general, los factores de riesgo se dividen en factores evitables como la exposición a ciertas sustancias o algunos comportamientos, y factores no evitables como la edad o los antecedentes familiares. Además, el peso de cada factor de riesgo varía en función del tipo de cáncer. En la gráfica de la Figura 1.2, se puede ver la incidencia de los tumores más frecuentes a nivel mundial en el año 2018.

¹Proyecto de la plataforma *Global Cancer Observatory* que presenta estadísticas globales para informar sobre el control del cáncer y su investigación [3].

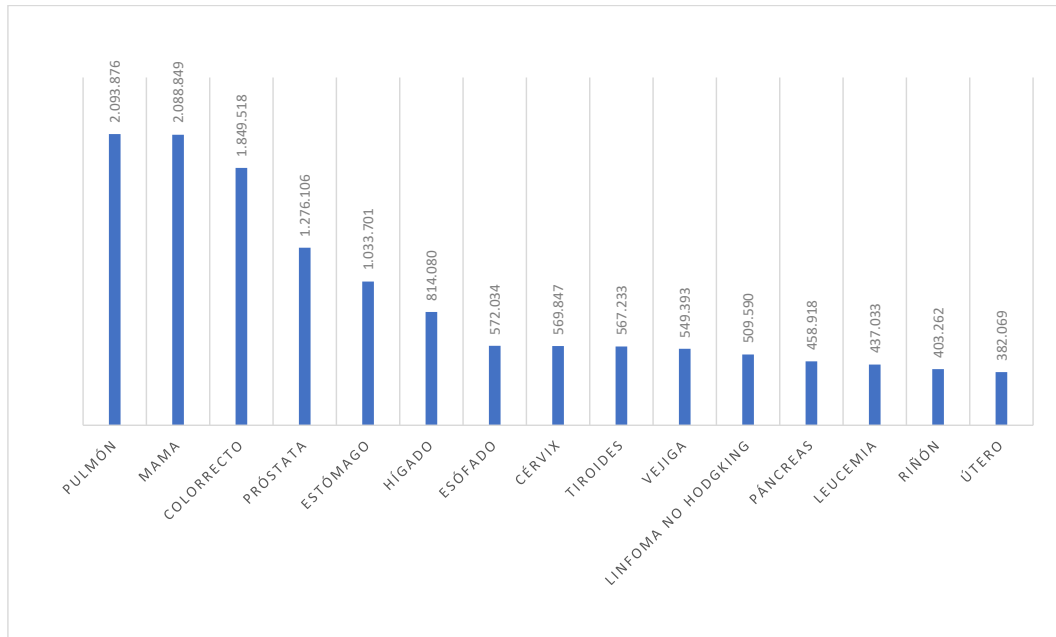


Figura 1.2: Incidencia estimada de los tumores más frecuentes a nivel mundial en el año 2018 (ambos sexos) [4].

1.1.1.2 Cáncer de próstata

En el presente proyecto se investiga sobre el cáncer de próstata, tal y como se observa en la Figura 1.2, se encuentra entre los tumores más frecuentes en España. El cáncer de próstata se diagnostica más frecuentemente en hombres de edad avanzada, siendo la edad media en la que se diagnostica a un paciente con dicha enfermedad de 66 años.

En base a los datos estadísticos recogidos por la SEOM para el año 2017 y gráficos, se hace evidente la necesidad de desarrollar tanto métodos preventivos como métodos de diagnóstico precoz que garanticen un buen pronóstico.

La anatomía de la próstata es relevante para comprender los diferentes métodos diagnósticos y sus dificultades. La próstata es un órgano interno del aparato reproductor masculino que se encuentra situado en la pelvis, detrás del pubis, delante del recto y por debajo de la vejiga (véase la Figura 1.3). Abarca la primera porción de la uretra y la atraviesa en toda su longitud, debido a ello, gran parte de los procesos patológicos ocasionados en esta glándula provocan alteraciones en la orina o en su evacuación [5].

El tamaño normal de este órgano es de 4 cm de largo por 3 cm de ancho pero se debe tener en cuenta que aumenta con la edad. Su función es producir el líquido prostático que junto con el líquido vesicular forman el líquido seminal que aporta el medio necesario para mantener, proteger y ayudar a los espermatozoides o gametos masculinos.

El organismo del ser humano está formado por tejidos, que a su vez están compuestos por células que siguen un conjunto ordenado de sucesos denominado el ciclo celular. Como se ha detallado en la subsección anterior, el cáncer se origina cuando las células sanas sufren un cambio y proliferan

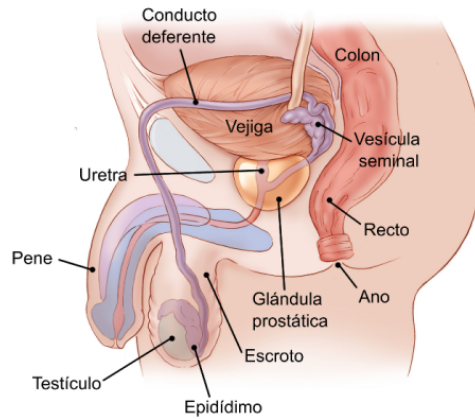


Figura 1.3: Órgano reproductor masculino [5].

sin control. De esta forma, cuando las células tumorales están ubicadas en el órgano prostático se habla de cáncer de próstata.

Cabe destacar que es de vital importancia un diagnóstico a tiempo de esta enfermedad ya que de lo contrario, el pronóstico de la enfermedad empeora considerablemente, así como la calidad de vida de las personas que la padecen. Algunos de los signos de cáncer de próstata son los siguientes [1]:

- Flujo débil de orina.
- Micción frecuente.
- Dificultad para comenzar el flujo de orina o para vaciar la vejiga.
- Sangre en el semen o en la orina.
- Cansancio, falta de aire, mareos o piel pálida.

El abordaje del tratamiento depende de una serie de factores tales como el estadio del cáncer o la edad del paciente. También es importante tener en cuenta otros problemas de salud y los efectos secundarios que pueda provocar el tratamiento previsto al paciente.

Los exámenes de detección que se emplean comúnmente son:

- Examen rectal digital. Exploración médica que consiste en la introducción de un dedo a través del esfínter anal para realizar la palpación digital en busca de anomalías en el tamaño de la glándula prostática.
- Análisis de PSA (del inglés - *Prostatic Specific Antigen*) en sangre. Análisis de sangre que busca la presencia del antígeno prostático específico (PSA). Cuanto mayor es el valor de la concentración de PSA, mayor es el riesgo de progresión. Aun así, es un marcador inespecífico, puede estar alterado por más razones y sus resultados deben ser interpretados cuidadosamente.

Si alguno de estos exámenes de detección dieran resultados positivos, se realizaran pruebas diagnósticas:

- Análisis de PCA3. Análisis de orina en el que se busca la existencia del gen *PCA3* (del inglés - *Prostate Cancer Antigen*) más específica que el análisis de PSA en sangre, ya que, en este caso, el gen únicamente se encuentra sobreexpresado en casos de cáncer de próstata. No se encuentra aún implementado en los hospitales.
- Ecografía transrectal. Procedimiento en el que se introduce una sonda en el recto produciendo ecos que rebotan en los tejidos. A partir de estos ecos se forman imágenes computarizadas de la próstata. Además, se utiliza para guiar las agujas durante la biopsia.
- Biopsia. Se trata del método diagnóstico que se emplea hoy en día para corroborar la existencia de un posible cáncer de próstata. Este procedimiento es una técnica invasiva y, por ello, se realizan previamente las pruebas inespecíficas de detección de cáncer de próstata nombradas anteriormente con el fin de evitar la biopsia. Esta técnica consiste en la extirpación de una pequeña zona de tejido para poder examinarla y analizar indicios de cáncer. Mediante la biopsia se puede realizar un diagnóstico definitivo cuando las pruebas de detección proporcionan resultados anormales. Esta prueba conlleva ciertos riesgos, debido a que se realiza vía transrectal, como dificultad para orinar, hemorragia e infección. Normalmente, se lleva a cabo un abordaje transrectal con una pistola para biopsias y una guía ecográfica.

En la Figura 1.4 se puede observar el procedimiento que se sigue para la detección y el diagnóstico de cáncer de próstata. El análisis de las muestras de tejido prostático extraído en la biopsia se establece como diagnóstico definitivo, ya que es lo que permite decidir con mayor seguridad si el paciente padece cáncer. Sin embargo, se utiliza previamente el análisis del PCA para evitar realizar dicha técnica invasiva.

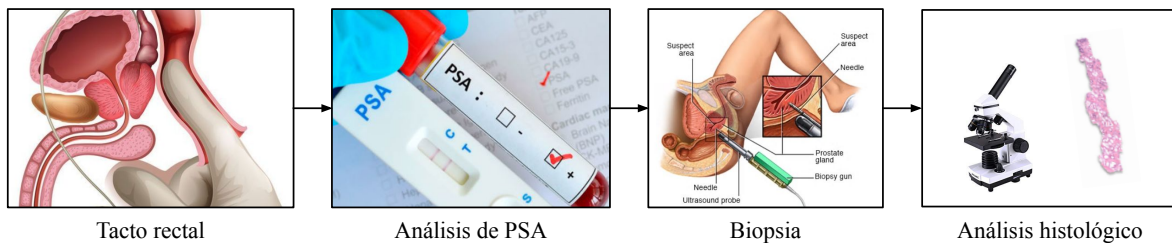


Figura 1.4: Procedimiento diagnóstico para detectar el cáncer de próstata [6], [7], [8].

Una vez realizada la biopsia, se obtienen los cortes histológicos de las muestras para su posterior análisis. Durante el análisis histológico, el especialista en anatomía patológica se encarga de buscar anomalías en el tejido mediante el uso de un microscopio. Este proceso conlleva un tiempo importante y existe cierta subjetividad entre diferentes especialistas. Por ello, el presente Trabajo Fin de Máster (TFM) pretende mejorar este proceso en términos de coste-efectividad mediante la digitalización y el análisis de estas imágenes con la finalidad de desarrollar un sistema de segmentación glandular automática de ayuda al diagnóstico por computación capaz de segmentar automáticamente las glándulas histológicas de próstata mediante algoritmos de inteligencia artificial.

Como se ha expuesto anteriormente, el cáncer de próstata se encuentra entre los tipos de cáncer que presentan una mayor tasa de incidencia en los últimos años debido al envejecimiento poblacional. Ello conlleva, un incremento del volumen de trabajo en el campo de la anatomía patológica y, por tanto, mayores cantidades de información a analizar. El avance de la tecnología

y, consecuentemente la patología digital, permiten aumentar las posibilidades de un diagnóstico más preciso y precoz, solucionando el problema anterior.

El uso de las TIC (Tecnologías de la información y de la comunicación) en anatomía patológica permite adquirir un fichero de imágenes digitales, obtenidas mediante la preparación convencional (técnica histológica), que pueden ser analizadas con herramientas software a fin de ayudar a los patólogos reduciendo su carga de trabajo mediante la automatización de determinados procesos durante el procedimiento diagnóstico.

A continuación, se detalla cómo se obtienen este tipo de imágenes, cómo son y de qué constan.

1.1.2 Imagen histológica prostática

La histología es la disciplina dentro de la biología que se encarga del estudio de la composición, la estructura y las características de los tejidos orgánicos de los seres vivos. Para poder realizar el análisis histológico del tejido extraído mediante la biopsia es necesario un procesado de la muestra conocido como técnica histológica. Dicho procesado consiste en un conjunto de operaciones a las que se somete el material biológico para que sea posible su estudio bajo el microscopio por los especialistas en anatomía patológica. El procedimiento incluye una serie de pasos, los más importantes son los que se muestran en la Figura 1.5.

Para llevar a cabo la preparación del corte histopatológico, el primer paso es la fijación, que consiste en colocar una sustancia fijadora para conservar el tejido. Luego se limpia el tejido con agua destilada para retirar el exceso de sustancia, se realiza la deshidratación para evitar la deformación morfológica del tejido y, finalmente, se efectúa el proceso de aclaramiento.

El siguiente paso es la inclusión, cuya finalidad es proporcionar consistencia al tejido para permitir obtener cortes delgados que puedan ser observados en el microscopio. Normalmente se utiliza parafina. Posteriormente, se hace uso del micrótopo para realizar cortes en secciones lo suficientemente delgadas como para permitir el paso de luz.

Finalmente, se elimina la parafina y se aplica una tinción al corte que permita observar la morfología tisular. El tipo de tinte escogido varía en función de las estructuras que se quieran diferenciar. La tinción empleada en este proyecto es la hematoxilina y eosina (H&E).

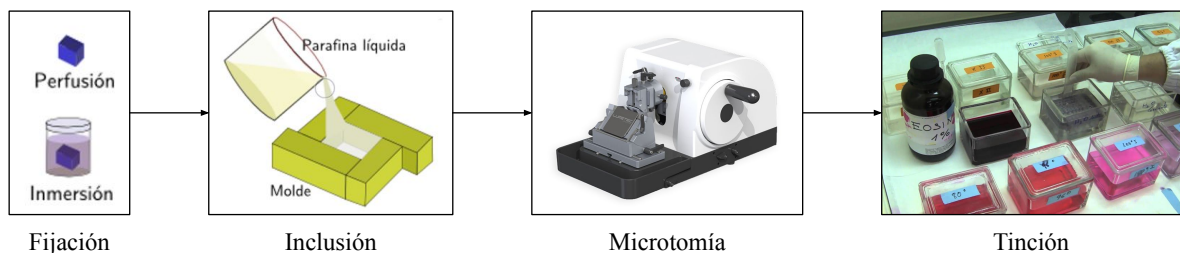


Figura 1.5: Procedimiento general para la preparación del corte histológico.

En el presente TFM, se ha trabajado con imágenes teñidas con H&E que fueron escaneadas y digitalizadas para su posterior estudio computacional. La tinción H&E es la usada comúnmente para medicina diagnóstica. La hematoxilina debido a su naturaleza básica tiñe en tonos azul y púrpura las estructuras ácidas (basófilas) como los núcleos. En cambio, la eosina, por ser ácida,

tiñe en tonos rosa los componentes básicos (acidófilos) como el estroma [9]. Gracias a esta tinción se pueden observar resaltadas las estructuras de interés.

En la Figura 1.6, se pueden apreciar los diferentes colores que adquiere cada estructura. Los cuatro componentes principales son el estroma (área rosa), los núcleos (estructuras tono azul-púrpura oscuro), el citoplasma (región púrpura clara) y el lumen (área blanca).

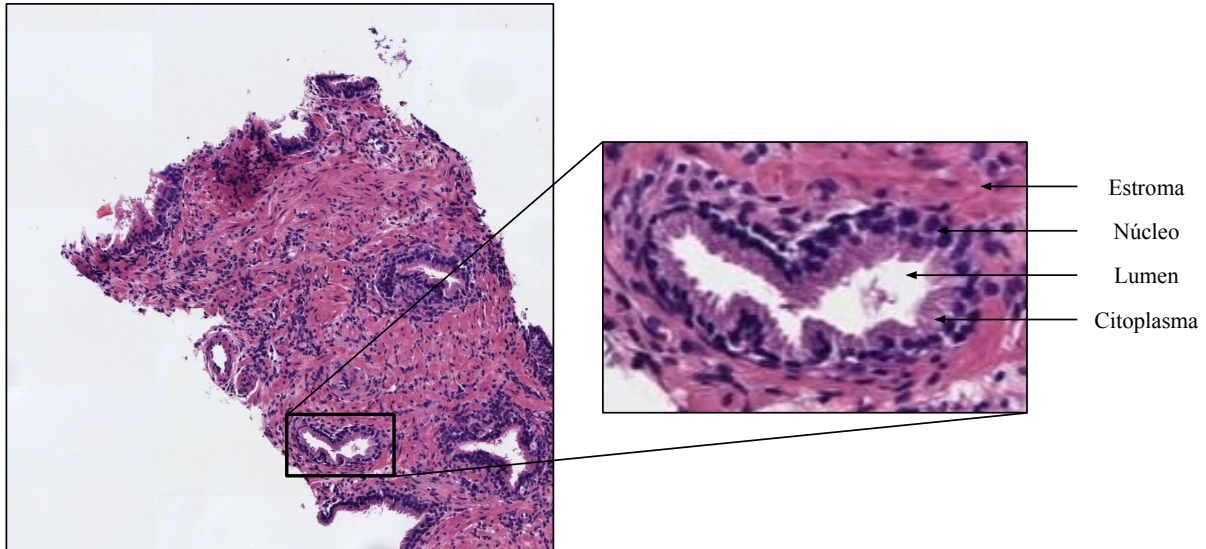


Figura 1.6: Imagen histológica prostática y glándula prostática.

En base al estudio del estado del arte y a anteriores proyectos, en este TFM se pretende segmentar de forma automática regiones glandulares mediante novedosas técnicas de inteligencia artificial basadas en algoritmos de *machine learning* y de *deep learning*. La finalidad es poder emplear estas regiones de interés para extraer características que permitan dilucidar las diferencias entre tejidos de próstata sanos y cancerosos en una etapa temprana.

1.1.3 Escala de Gleason

Es importante conocer el grado del cáncer del paciente ya que, mientras el estadio indica dónde está presente el cáncer en el organismo y su desarrollo (comportamiento a nivel macro), el grado describe el aspecto de las células al estudiarlas bajo el microscopio (comportamiento a nivel micro).

Tradicionalmente, el grado del cáncer prostático se asigna en función del *Gleason Score* o escala de Gleason, sistema de puntuación cuyo nombre hace referencia a su creador, el Dr. Donald Gleason. Esta escala separa los tejidos histológicos de próstata en 5 grados distintos atendiendo a 5 patrones diferentes. De forma que la escala Gleason va del 1 al 5, de menor a mayor malignidad o severidad de la enfermedad [10].

De acuerdo con el sistema de puntuación del *Gleason Score*, los especialistas en anatomía patológica asignan a cada muestra histológica los dos grados más predominantes. De forma que cuanto más alto sea el grado, más severo y mutado se encuentra el cáncer. Es por ello, que los patólogos emiten el diagnóstico final como la suma de los dos grados más predominantes. Cabe destacar que el orden durante la asignación de los grados es importante, ya que un *Gleason Score* = 7,

puede proceder de un diagnóstico 3+4 o 4+3. El primero indica que predomina el grado 3, a diferencia del segundo, cuyo grado predominante es el 4 (más severo que el 3).

En 2014, la Sociedad Internacional de Patólogos en Urología publicó un sistema revisado de clasificación del cáncer de próstata denominado *ISUS Grade Group*. Este sistema es más sencillo y, por eso, se utiliza más hoy en día. En la Tabla 1.2 se puede apreciar las conversiones entre un sistema y el otro.

<i>ISUS Grade Group</i>	<i>Gleason Score</i>
Grado 1	≤ 6
Grado 2	7 (3 + 4)
Grado 3	7 (4 + 3)
Grado 4	8
Grado 5	9 - 10

Tabla 1.2: Comparación entre los dos sistemas de puntuación del cáncer de próstata [11].

En el presente TFM, se hace uso del sistema *ISUS Grade Group*. En la Figura 1.7 se muestra una imagen de patrones en la que se pueden distinguir las principales características histológicas de cada grado.

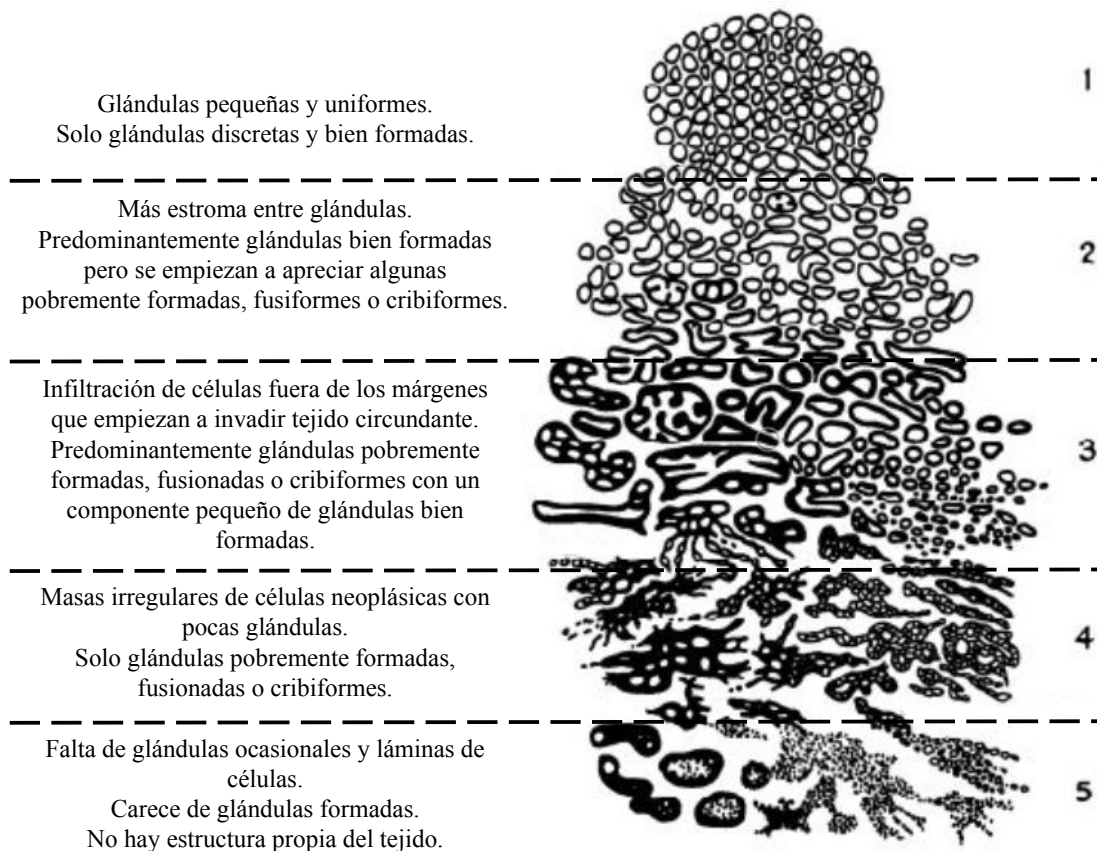


Figura 1.7: Escala de patrones de Gleason y sus principales características histológicas [12]

El grado 1 y 2, como se puede observar, son tejidos con patrones similares al tejido sano y cuya repercusión clínica es mínima. Por lo tanto, el estudio del cáncer de próstata es a partir del grado 3. En cuanto al grado 3 y 4, su diagnóstico es el más complicado y, por ello, en la literatura se encuentran una gran cantidad de estudios referidos a estos grupos. El grado 3 sigue un patrón similar al tejido normal, pero en el que ya se pueden apreciar indicios de rotura de glándulas y de infiltración celular. En cambio, el grado 4 se caracteriza por presentar una pérdida casi total de la estructura normal del un tejido. Finalmente, el grado 5 es el más avanzado y, por este motivo, su detección y su diagnóstico son más sencillos.

Por tanto, el mayor reto se encuentra en diagnosticar el cáncer de grado 3 de forma precoz. En base a diferentes estudios, se entiende que la forma más precisa y óptima de distinguir entre este grado y el tejido sano es mediante el estudio glandular. Por ello, en el presente TFM se pretende identificar y segmentar cada una de las glándulas de los diferentes tejidos a fin de evaluar el carácter benigno o maligno a nivel glandular.

A grandes rasgos, las glándulas benignas se encuentran distribuídas por el tejido de forma separada, son grandes y densas, sus lúmenes son grandes, con formas muy variables y con los límites anchos, es decir, poseen varias capas de núcleos. En contraposición, las glándulas patológicas de grado 3 son más pequeñas y con forma más circular, sus lúmenes son pequeños, circulares y con los límites estrechos, solo se observa una capa de células a su alrededor. En la Figura 1.8, se puede observar una imagen de 1024×1024 píxeles de tejido sano y una de tejido patológico de grado 3.

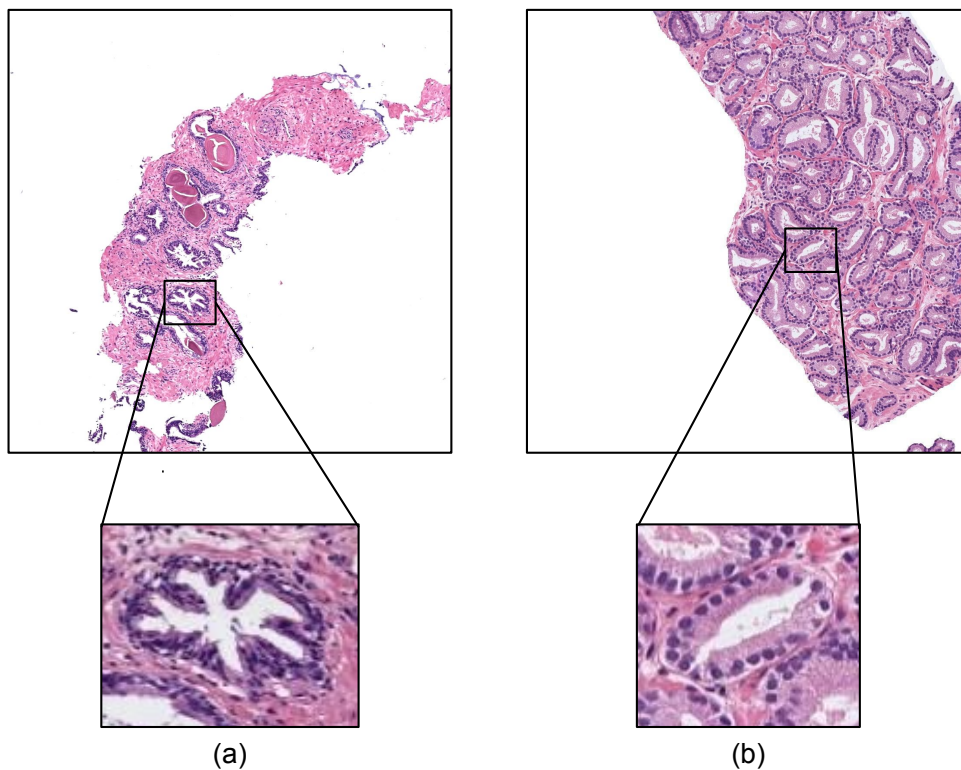


Figura 1.8: Ejemplo de muestras histológicas para dos tipos de tejido prostático. (a) Tejido sano y (b) tejido patológico de grado 3, ambas con sus respectivas zonas de interés (glándulas).

1.2 Proyecto SICAP

En el presente TFM, el objetivo reside en desarrollar un sistema de segmentación automática de glándulas prostáticas mediante una rama de la inteligencia artificial denominada aprendizaje profundo (*deep learning*) que ha cogido un gran auge en este campo de estudio durante los últimos años. El principal interés es el diseño y desarrollo de un sistema de ayuda al diagnóstico capaz de diferenciar entre un tejidos sano y uno patológico en su etapa más temprana. Este trabajo se enmarca dentro de un proyecto nacional de mayor envergadura denominado SICAP, cuyo acrónimo proviene de “Sistema de Interpretación de Imágenes Histopatológicas para la Detección del Cáncer de Próstata”.

SICAP es un proyecto nacional subvencionado por el Ministerio de Economía, Industria y Competitividad en el que colaboran conjuntamente el grupo CVBLab de la Universidad Politécnica de Valencia, la Universidad de Granada y el departamento de Anatomía Patológica del Hospital Clínico Universitario de Valencia. El objetivo global del proyecto es el desarrollo de un *software* para el diagnóstico automático de cáncer de próstata que se pueda integrar en la práctica clínica. La finalidad de ello es disminuir la carga de trabajo a la que se enfrentan los especialistas en anatomía patológica, así como aumentar la calidad y la exactitud diagnóstica, y mejorar en términos de coste-efectividad.

El proyecto SICAP abarca el desarrollo de diferentes técnicas que permitan clasificar las muestras con la mayor precisión posible acorde con el sistema de puntuación del Gleason Score. Para ello, se implementan técnicas de segmentación, métodos de extracción y selección de características y algoritmos de *machine learning* y de *deep learning*.

1.3 Objetivos

El objetivo último del presente trabajo es la segmentación automática de glándulas prostáticas a partir de imágenes histopatológicas teñidas con hematoxilina y eosina como parte de un proyecto mayor que tiene como finalidad ayudar a los profesionales en anatomía patológica a formular un diagnóstico preciso y precoz.

Para lograr dicho objetivo principal en este TFM se abordan una serie de objetivos secundarios:

1. Adquisición de la base de datos de imágenes que es objeto de estudio en este proyecto y adecuación de la misma para que puede ser empleada,
2. Realizar una revisión de la literatura científica con el fin de obtener información sobre el estado del arte y poder conocer los diferentes procedimientos y métodos utilizados para conseguir una segmentación automática de objetos.
3. Generación de un *ground truth*² de segmentación de estructuras glandulares para la base de datos de imágenes histopatológicas de próstata. Debido a que usaremos algoritmos de segmentación supervisados, se usará tanto para la fase de entrenamiento como para la fase de predicción.

²Término empleado para referirse al conjunto de información construida manualmente con la finalidad de emplearla para confirmar o validar los datos creados de forma automática.

4. Partición de la base de datos en diferentes subconjuntos en función de paciente y de grado de malignidad. De esta forma, las imágenes de un mismo paciente nunca se utilizarán para entrenar y para validar en la misma iteración y las imágenes empleadas en la fase de testeo serán completamente desconocidas para el algoritmo. Además, al dividir también en función de malignidad se trata de balancear la base de datos habiendo tantas imágenes de tejido sano como patológico en una misma iteración.
5. Realizar un parcheado de las imágenes y eliminar aquellas que poseen demasiados píxeles de una clase para conseguir balancear entre fondo y glándula.
6. Encontrar los hiperparámetros más óptimos para llevar a cabo la fase de entrenamiento y entrenar los modelos con los diferentes subconjuntos escogidos.
7. Predecir el comportamiento del modelo mediante muestras nuevas.
8. Realizar una reconstrucción de los mapas de segmentación obtenidos a la salida de la red convolucional para poder compararlos con el *ground truth* construido y obtener resultados cualitativos y cuantitativos.
9. Llevar a cabo una comparación del método de segmentación diseñado y desarrollado con los métodos propuestos por los autores del estado del arte.
10. Extraer conclusiones mediante el análisis de los resultados obtenidos y comprobar que se ha cumplido el objetivo principal. Identificar problemas y proponer posibles mejoras y líneas para investigaciones futuras.

1.4 Guía de la memoria

En el capítulo 2, se realiza una descripción de los fundamentos teóricos que rodean este proyecto y que han sido necesarios para llevar a cabo el diseño de la arquitectura de *deep learning* empleada.

En el capítulo 3, se hace referencia al estudio de la literatura científica tanto a nivel de segmentación glandular automática con diferentes métodos como a nivel de segmentación de objetos con redes neuronales.

El capítulo 4 consta de dos apartados bien diferenciados. En la primera sección, se detallan los materiales que han sido necesarios. En la segunda sección, se detalla el método diseñado e implementado para cumplir el objetivo final del trabajo.

En el capítulo 5 se muestran, se discuten y se comparan con la literatura los resultados obtenidos en este proyecto.

Finalmente, en el capítulo 6 se extraen conclusiones sobre el estudio realizado y se detallan posibles líneas de investigación dentro del proyecto.

Capítulo 2

Marco teórico

A la hora de abordar el marco teórico del presente proyecto, se empieza tratando unas nociones básicas sobre redes neuronales para luego enfocarse en las redes neuronales convolucionales y, finalmente, en los *autoencoders* y su uso en segmentación.

2.1 Introducción al deep learning en segmentación

La inteligencia artificial (IA) es un concepto complejo que abarca una gran cantidad de campos diferentes, donde se puede encontrar cualquier técnica que permita a las computadoras imitar el comportamiento humano. La IA tiene dos objetivos principales, uno es tecnológico: emplear los ordenadores para conseguir cosas útiles y, el otro es científico: usar conceptos y modelos que ayuden a resolver cuestiones sobre los seres humanos [13]. Se encuentran aplicaciones prácticas de la IA en el hogar, en los vehículos, en los hospitales, en los bancos o en internet. Cada vez que una máquina completa tareas basadas en un conjunto de reglas que resuelven un problema, tal comportamiento se considera “inteligente”.

Existen diversas categorías con múltiples variaciones. Una de ellas es la IA clásica que, especialmente combinada con la estadística, puede replicar el aprendizaje, la planificación y el razonamiento del cerebro humano. Aquí es donde se encuentra el aprendizaje automático (ML, del inglés - Machine Learning) cuya intención es permitir a las máquinas aprender por sí mismas utilizando los datos proporcionados con el fin de que hagan predicciones precisas. Las técnicas convencionales de ML consisten en extraer características “manualmente” para intentar encontrar patrones repetidos que ayuden a realizar de forma automática una tarea específica (por ejemplo, segmentación o clasificación).

El aprendizaje profundo (DL, del inglés - deep learning) es un subconjunto o una técnica dentro del ML considerada como la evolución del mismo. Los algoritmos de DL están inspirados por los patrones de procesamiento de información que se encuentran en el cerebro humano. De forma general, el cerebro trata de descifrar la información que recibe a través del etiquetado y la asignación de elementos en categorías, es decir, cada vez que se recibe información nueva, el cerebro

intenta compararla con los elementos conocidos antes de darle un sentido. Este concepto es el utilizado en los algoritmos de DL [14].

Este enfoque utiliza modelos de red neuronal profunda, que es una variación de la red neuronal, pero con una gran aproximación al cerebro humano utilizando un mecanismo avanzado. Las redes neuronales artificiales son un modelo computacional inspirado en la transmisión de los impulsos nerviosos a través del cuerpo. Se trata de un sistema que toma como analogía las redes neuronales biológicas de forma que adquiera una gran colección de unidades neuronales interconectadas para permitir la comunicación. Estas unidades son nodos o neuronas que funcionan como procesadores simples en paralelo. Hacia el año 1948, Rosenblatt propuso una neurona artificial básica llamada Perceptrón, que es, simplemente, un modelo lineal umbralizado [15]. Por tanto, su utilidad era construir clasificadores capaces de separar de forma lineal los datos pertenecientes a dos o más clases. A partir de este momento, se han investigado y desarrollado enfoques más profundos y complejos.

Los enfoques computacionales anteriores como el aprendizaje automático, requieren de características explícitas de la imagen definidas a priori y emplean técnicas convencionales de regresión o clasificación. En contraposición, los enfoques de aprendizaje profundo tienen el potencial de superar las limitaciones anteriores, obtener resultados reproducibles y lograr un rendimiento mediante la multiplexación de recursos computacionales.

Los sistemas de DL se basan en redes neuronales de varias capas que pueden extraer características cada vez más complejas relacionadas con tareas directamente de los datos. Los desarrollos recientes en el diseño y la capacitación de la arquitectura de redes neuronales han permitido a los investigadores resolver tareas de aprendizaje, como la visión por computador y el reconocimiento de voz, que antes no se podían resolver en el campo de la visión por computadora.

Las redes neuronales convolucionales (CNN) destacan en el aprendizaje de características representativas dentro de las imágenes. Esta capacidad hace que las CNN sean extremadamente valiosas para resolver problemas dentro del dominio de la visión artificial. La finalidad de las CNN es actuar a modo de extracción automática de características para que la red pueda aprender los patrones repetidos de las imágenes, señales o los *inputs* que se introduzcan a la red.

Históricamente, los modelos de CNN diseñados, a pesar de que evidenciaron una precisión más allá de las capacidades humanas para ciertas tareas, tendían a ser muy grandes a nivel de memoria e ineficientes computacionalmente. Sin embargo, los enfoques más recientes, junto con el avance de la tecnología, están demostrando prometedores modelos de alta precisión y alta eficiencia cuando se aplica para una amplia gama de tareas como detección de objetos, segmentación, etc.

Por otra parte, la segmentación es un campo de la visión artificial que consiste en dividir una imagen en elementos u objetos. La finalidad es simplificar o transformar la representación de una imagen en otra más significativa o más fácil de analizar. A grandes rasgos, las redes neuronales en este campo se basan en el procesamiento de pequeñas áreas para construir un mecanismo que marque las zonas de la imagen de acuerdo a la categoría reconocida por la red neuronal.

Centrándose en las redes convolucionales de segmentación, existen una serie de modelos publicados, como la red AlexNet [16], primera CNN utilizada en este campo, o la FCN (*Fully Convolutional Networks for Semantic Segmentation*) [17], red que popularizó el uso de las CNN para segmentación semántica.

En este apartado, se introducen los conceptos esenciales para entender cómo se ha tratado de resolver el problema planteado en este TFM.

2.2 Fundamento teórico red neuronal

2.2.1 Estructura básica red neuronal

Como se menciono anteriormente, los sistemas neuronales artificiales son modelos inspirados por la comprensión que se tiene del cerebro y, por tanto, es necesario un conocimiento previo del comportamiento de una célula nerviosa biológica.

Una célula nerviosa o neurona es una célula biológica que procesa información. Se trata de la unidad básica de una red neuronal biológica y consta de las siguientes partes:

- Dendritas. Tienen forma de ramas y son responsables de recibir la información de otras neuronas con las que se encuentran conectadas.
- Soma. Se trata del cuerpo celular de la neurona y se responsabiliza del procesamiento de la información recibida.
- Axón. Medio a través del cual las neuronas envían la información.
- Sinapsis. Es la conexión entre el axón y las dendritas de otras unidades neuronales.

En la Figura 2.1, se puede apreciar la equivalencia de la morfología de una neurona biológica con un modelo computacional. Las dendritas son las entradas de la neurona, en el núcleo se combinan las entradas mediante una suma ponderada por pesos, donde cada peso representa la fuerza de la sinapsis para esa entrada. La neurona se activa o se dispara si el resultado supera cierto umbral y transmite la información a la siguiente neurona. El axón es la salida de la neurona de forma que las señales de salida se producen después de combinar las señales de entradas y determinada regla de activación (conocida en las neuronas artificiales como función de activación).

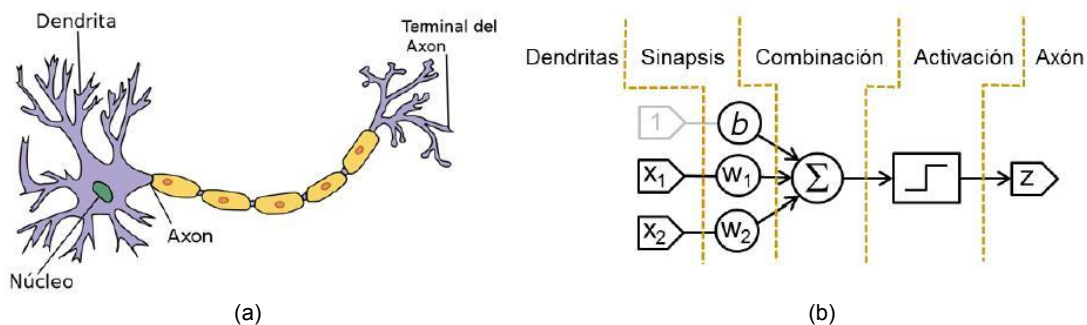


Figura 2.1: (a) Esquema básico neurona biológica y (b) esquema básico neurona artificial [18].

Matemáticamente, esto se puede expresar de la siguiente forma:

$$z = b + \sum_i^N x_i w_i \quad (2.1)$$

$$Z = a(z) \quad (2.2)$$

Donde x_i son las entradas, es decir, datos de origen que se introducen en la red neuronal. w_i son los pesos neuronales o parámetros del modelo que se actualizan de acuerdo con un algoritmo *backpropagation* (explicado más adelante en la memoria) para optimizar el valor de salida de la red. b es el *bias* o sesgo que permite mover la función en el gráfico numérico con la finalidad de acercarse al valor de salida deseado. z es la predicción que la red genera en forma de conjunto de valores reales. Finalmente, a es la función de activación, equivalente a la regla de activación en la neurona biológica (explicado más adelante en la memoria).

Es importante destacar que en el espíritu de las redes neuronales, no solo se intenta imitar la morfología de una neurona biológica en un modelo computacional, sino que también se pretende emular el proceso de aprendizaje: reforzar las conexiones sinápticas adecuadamente según se van enseñando ejemplos al sistema.

Una neurona sola o aislada carece de razón de ser, su función es valiosa en la medida que se asocia con otras neuronas dando lugar a una red. La manera más intuitiva para realizar la organización de las neuronas es mediante una serie de capas, en las que cada neurona se encuentra conectada a las neuronas de la capa anterior. La distribución varía según el problema que se desee resolver, en función del número de neuronas, del número de capas y de sus conexiones se obtiene una determinada red neuronal.

El Multilayer Perceptron (MLP) es una red neuronal que conecta varias capas en un gráfico dirigido, de forma que pueden responder a preguntas complejas con precisión [19]. Cada perceptrón recibe la información de la capa anterior, la procesa y la transmite a la capa posterior. Un MLP contiene una capa de entrada, una capa oculta mínimo y una capa de salida (véase Figura 2.2).

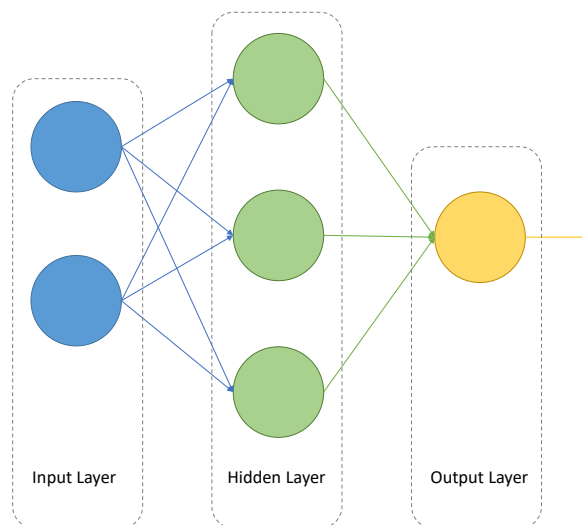


Figura 2.2: Estructura general *multilayer perceptron*.

En definitiva, una red neuronal básica tiene una capa visible de datos de entrada y una capa visible de datos de salida, el resto de neuronas se encuentran dentro de la caja negra y reciben el nombre de “capas ocultas”.

2.2.2 Algoritmo forward-backward propagation

En 1986, Rumelhart formalizó un método para que un perceptrón multicapa aprendiera la asociación entre un conjunto de entradas y sus salidas, el método conocido como *backpropagation error* [20].

Tras definir la red neuronal y los pesos iniciales se realiza una etapa *feed forward* para generar una predicción inicial y se calcula una función de pérdida o coste. El objetivo será minimizar dicha función. Esta etapa consiste en alimentar la red con ejemplos y hacerlos avanzar a través de las capas hasta producir una salida. De tal forma que cada una de las variables de entrada x_i se combina con unos pesos w_i (inicialmente aleatorios) en cada una de las neuronas de la capa posterior. Así, se obtiene una nueva variable Z (véase ecuación 2.2) que se combina con los pesos w_i de la siguiente capa para dar lugar a otra variable. Este proceso se repite para las distintas neuronas de las capas ocultas hasta la capa de salida.

Tras llevar a cabo el *forward propagation* se obtiene una salida, la cual se compara con la salida óptima mediante una función de pérdida. De este modo, se obtiene un error que se intenta minimizar en la siguiente iteración mediante el *backward propagation*. Para ello se utiliza un optimizador, como puede ser el *Stochastic Gradient Descent* (SGD) [21]:

$$w(t+1) = \alpha w(t) + \eta \alpha \frac{\partial}{\partial w(t)} L(y, y') \frac{\partial}{\partial b(t)} L(y, y') \quad (2.3)$$

donde $w(t)$ son los pesos iniciales, $b(t)$ es el sesgo, η es la tasa de aprendizaje y $L(y, y')$ la función de pérdidas que se pretende minimizar. La función de pérdidas o coste (*loss function*) es una medida del error cometido para un ejemplo. No se trata exactamente de la diferencia entre el valor real y el estimado, de ahí que se le denomine “medida del error” o “pérdida”.

En la Figura 2.3 se ilustra un ejemplo del proceso *forward-backward*.

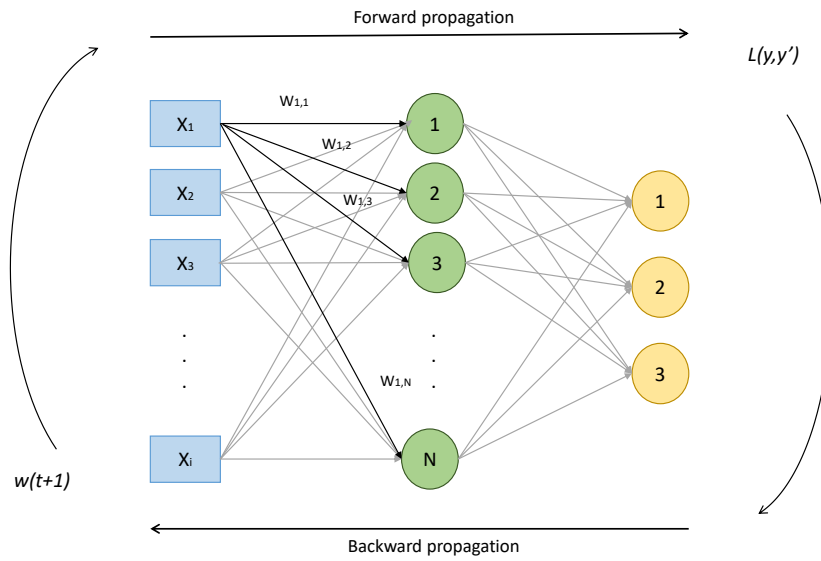


Figura 2.3: Ejemplo ilustrativo del algoritmo *forward-backward propagation* para una arquitectura de una capa oculta con N neuronas y una capa de salida con tres neuronas donde i son las variables de entrada que se tienen.

El procedimiento que sigue este algoritmo se puede expresar con una serie de pasos:

1. Inicialización de la red. Se asignan pesos aleatorios a las diferentes neuronas.
2. Etapa *feed forward*. Las neuronas de la capa de entrada reciben la información y la procesan de acuerdo a la siguiente ecuación:

$$Z_l = b_l + W_l X \quad (2.4)$$

donde l es el número de la capa. Este resultado se transforma mediante una función de activación:

$$a(Z_l) \quad (2.5)$$

3. Cálculo de la función de coste. Compara el valor real con la salida de la red. Es una medida del error cometido que se emplea para entrenar la red.

$$C(a(Z_l)) \quad (2.6)$$

4. *Back propagation*. En esta fase los errores se transmiten hacia atrás partiendo de la capa de salida. El error asociado a cada neurona es un error proporcional a su contribución en el error global de la red.

Se computa el error del modelo en el punto del gradiente en el que se encuentra y se calculan las derivadas parciales en dicho punto, con esto se obtiene un vector de dirección que indica la pendiente de la función, es decir, hacia donde el error se incrementa (gradiente). Véase Figura 2.4. Iterativamente, el modelo se va moviendo en la dirección contraria del gradiente para ir reduciendo la función de coste. Por tanto, el objetivo es calcular la derivada parcial del coste respecto de los parámetros de la red neuronal: los pesos y el término de sesgo.

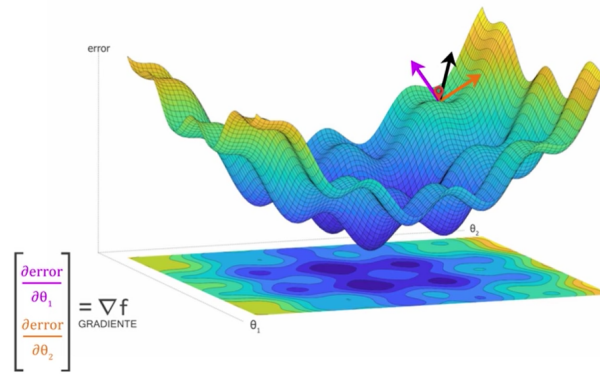


Figura 2.4: Ejemplo de una superficie de función de coste y su gradiente sobre el plano.

Como se puede ver, el proceso de entrenamiento es un problema de optimización de la función de pérdidas donde los parámetros -pesos y sesgos-, se deben ajustar para minimizar la función anterior. Estos parámetros se aproximan con algoritmos de optimización iterativos u optimizadores. Algunos de los optimizadores más conocidos son el descenso del gradiente estocástico (SGD, del inglés - Stochastic Gradient Descent) y la estimación del momento adaptativo (Adam).

2.3 Redes neuronales convolucionales

A continuación, se explican algunas de las capas más importantes en una arquitectura convolucional.

Una red neuronal convolucional (*Convolutional Neural Networks*, CNN) es un tipo de red con aprendizaje supervisado que identifica características en las imágenes de entrada. Una CNN consta mínimo de un bloque convolucional de extracción automática de características que obtiene a su salida un vector unidimensional, este se pasa a la etapa *top model* que se enfrenta al problema que se desea resolver. De forma general, esta arquitectura está formada por uno o más bloques convolucionales donde cada bloque está compuesto por una capa convolucional, una capa de activación y una capa de pooling. De esta manera, las primeras capas convolucionales detectan líneas, curvas, detalles pequeños y se van especializando para reconocer formas más complejas.

La ventaja de esta arquitectura es que sus neuronas se pueden entrenar mediante la técnica de *backpropagation* que sirve tanto para la etapa de convolución optimizando las características visuales como para el *top model*.

2.3.1 Capa convolucional

La convolución es una operación muy importante en procesamiento de señales y de imagen. De forma general, la operación a realizar para convolucionar dos señales es la siguiente:

$$f[n] \otimes g[n] = \sum_{k=-\infty}^{+\infty} f[k] \cdot g[n - k] \quad (2.7)$$

donde f y g son las señales que se van a convolucionar mediante muestreos de ambas funciones en ciertos instantes de tiempo $f[k]$ y $g[n - k]$ siendo n y k valores enteros.

Para explicar la convolución de imágenes se va a suponer el siguiente ejemplo (Figura 2.5): una imagen convolucionada con otra que contiene un array de 4×4 unidades cuyo resultado es una imagen compuesta con 4×4 clones de la primera imagen. En la misma ilustración se puede ver otro ejemplo de una convolución de imagen en el que la campana de Gauss se convoluciona en cada píxel según su valor de intensidad dando como resultado una imagen suavizada de la imagen de entrada.

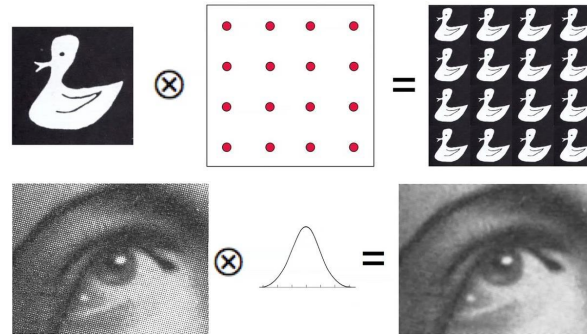


Figura 2.5: Arriba, ejemplo intuitivo de una convolución 2D. Abajo, ejemplo de convolución con una distribución normal [18].

Hay que tener en cuenta que no es posible convolucionar una imagen con otra de tamaño distinto, pero si que se puede convolucionar una región (parche o ventana) de la imagen original de tamaño $n \times n$ con una ventana de la otra del mismo tamaño. De forma general, a la otra imagen se le denomina filtro o *kernel* de convolución.

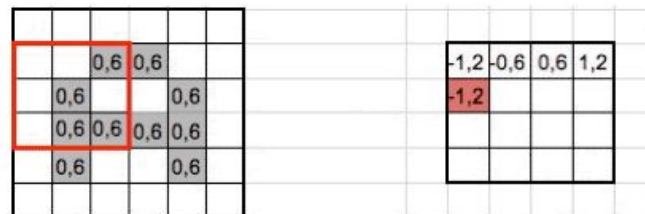


Figura 2.6: Ejemplo de un *kernel* desplazándose píxel a píxel por una imagen [22].

Existe una serie de parámetros que permiten controlar el tamaño resultante al realizar convolución de imágenes:

- *Size*. Dimensiones del filtro de convolución.
- *Stride*. Número de píxeles que se desplaza el filtro sobre la imagen.
- *Padding*. Relleno empleado para conseguir que todos los píxeles de la imagen puedan ser convolucionados junto a su vecindad. Cuando el borde del filtro alcanza el borde de la imagen, no puede avanzar más y no todos los píxeles actuarán como núcleo (por ejemplo en la Figura 2.6). Para evitar esto, se crea un borde de ceros alrededor de la imagen y así, se puede desplazar el filtro sobre ellos. De esta manera, se consigue que la imagen de salida tenga las mismas dimensiones que la imagen de entrada. En la Figura 2.7, es posible observar una representación gráfica de *padding valid* y *padding same*.

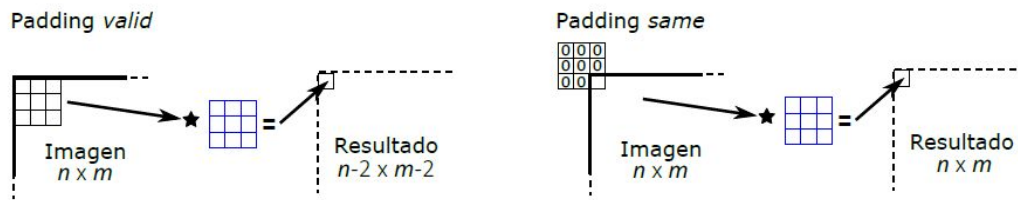


Figura 2.7: Representación de dos métodos de *padding* [18].

Combinando los diferentes parámetros explicados anteriormente, se puede controlar la anchura o altura de la imagen de salida, O , según la siguiente ecuación [18]:

$$O = 1 + \frac{2P + I - K}{S}, \quad (2.8)$$

donde I es la anchura o altura de la imagen de entrada, P es el valor del *padding*, S el valor del *stride* y K , el tamaño de la ventana.

En conclusión, las neuronas convolucionales dan como resultado elementos, filtros, que se utilizan para extraer características visuales de las imágenes y que son resultado de un entrenamiento. Así, al entrenar una neurona convolucional se está aprendiendo qué filtro utilizar.

De esta manera, mediante las neuronas convolucionales se es capaz de alimentar las redes neuronales con las imágenes originales directamente y no con un conjunto de características o variables extraídas de las imágenes. La etapa convolucional recibe la imagen y tiene la función de extraer características de la misma. Las neuronas involucradas son filtros convolucionales que se irán aprendiendo a lo largo del proceso de entrenamiento. Las características que el filtro es capaz de aprender dependen del tamaño del mismo, de forma que, por ejemplo, si el filtro es pequeño resaltarán frecuencias altas.

La capa convolucional es la parte central de un bloque convolucional de CNN debido a que se encarga de la extracción automática de características que se trata de la parte computacionalmente más costosa.

2.3.2 Capa activación

La función de activación decide si una neurona debe activarse o no realizando una transformación a la salida. Análogamente, en las redes neuronales biológicas, la función de activación representa la tasa de acción potencial de activación en la célula. En la forma más simple, se trata de una función binaria o una función escalón.

En las redes neuronales artificiales, la función de activación de un nodo define la salida de dicho nodo. El propósito de esta función es introducir la no linealidad a la salida de una neurona. Solo las funciones no lineales permiten que la red calcule problemas no triviales con un pequeño número de nodos.

A continuación, se comentan algunas funciones de activación comúnmente conocidas.

Función escalón

La función escalón es la más sencilla que se puede encontrar. En esta, la neurona se activa solo si el valor de la salida está por encima de cierto umbral. En la Figura 2.8, se muestra una representación de la misma.

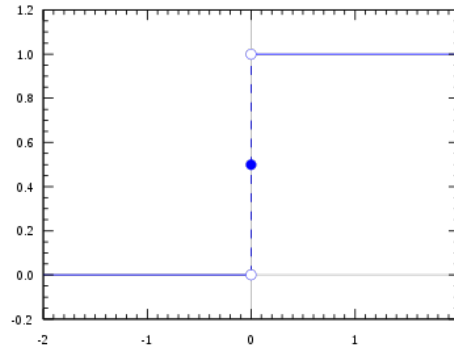


Figura 2.8: Representación de una función escalón.

Función sigmoide

La función sigmoide es la más antigua y extendida. Se define de la siguiente manera:

$$\sigma(z) = \frac{1}{1 + e^{-z}} \quad (2.9)$$

Teniendo en cuenta que la función z es igual a:

$$z = b + \sum_i w_i x_i \quad (2.10)$$

Se obtendría una transformación sigmoide de la función de entrada z . En la Figura 2.9 se puede observar como la función sigmoide comprime la salida a un rango de 0 a 1. Esta función es continuamente diferenciable. Además, hay que tener en cuenta que converge de forma lenta y que puede saturar.

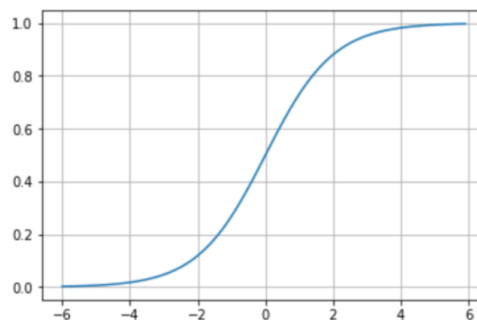


Figura 2.9: Representación de una función sigmoide.

Función tangente hiperbólica

Esta función tiene la peculiaridad de que transforma los valores a un rango de -1 a 1, de manera que los valores altos tienden de manera asintótica a 1 mientras que los valores bajos tienden a -1. Es una función que puede saturar y que converge de forma lenta. Normalmente, se utiliza como clasificador para decidir entre dos clases [23]. La función se expresa de la siguiente forma:

$$\tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}} \quad (2.11)$$

Su representación gráfica se puede observar en la Figura 2.10.

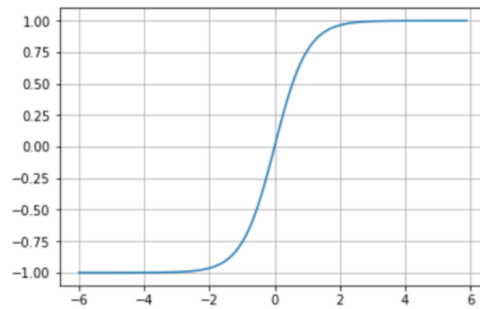


Figura 2.10: Representación de una función tangente hiperbólica.

Función ReLU

La función *Rectified Lineal Unit* (ReLU) transforma los valores negativos a 0 y permite el paso de los valores positivos. La mayoría de las redes neuronales utilizan esta función o alguna de sus variantes. Tiene un buen comportamiento con imágenes y se desempeña correctamente en redes convolucionales. Se expresa como:

$$R(z) = \max(0, z) \quad (2.12)$$

Su representación gráfica se puede apreciar en la Figura 2.11.

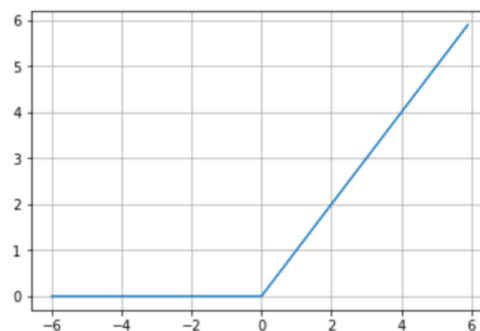


Figura 2.11: Representación de una función ReLU.

Las dos ventajas más destacadas de esta función son que permite que no todas las neuronas se activen siempre (sólo las que tienen respuesta positiva) y que posee una mejor etapa *backpropagation*. Véase un ejemplo de convolución con esta función de activación en la Figura 2.12.

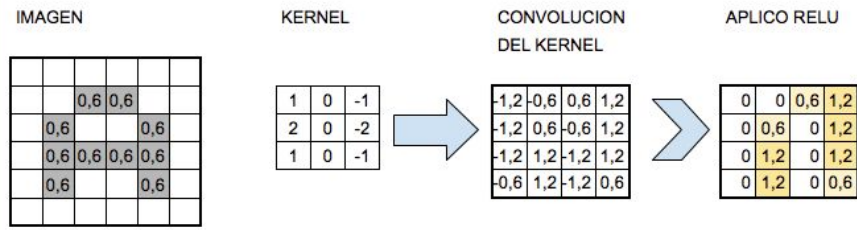


Figura 2.12: Ejemplo de convolución con un kernel 3×3 y de la aplicación de la función de activación ReLU [22].

Función Softmax

Esta función transforma las salidas a una representación en forma de probabilidades de manera que el sumatorio de todas las salidas para una entrada es de 1. Se expresa como:

$$f(z) = \frac{e^{z_j}}{\sum_{k=1}^K e^{z_k}} \quad (2.13)$$

Esta función posee algunas propiedades que la hacen muy interesante ya que permite una representación en forma de probabilidad y es diferenciable.

2.3.3 Capa Pooling

Consiste en la agrupación espacial de características de la imagen. Esta operación se define mediante una vecindad V de tamaño $N \times N$. Por ejemplo, la capa *MaxPooling* se expresa como:

$$pooling(V) = max(V). \quad (2.14)$$

La ventana deslizante se va desplazando sobre la imagen como el filtro de convolución pero aplicando otra operación, por ello, se define con los mismos parámetros (*size* y *stride*). Sin embargo, al terminar esta operación, se obtiene como resultado una imagen de tamaño menor (véase Figura 2.13).

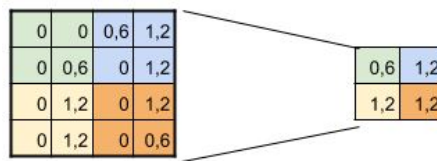


Figura 2.13: Ejemplo de aplicación de capa *Max-Pooling* con *size* 2×2 y *stride* 2[22].

2.3.4 Capa Dropout

Realiza una desconexión al azar de un cierto porcentaje de neuronas en cada iteración (véase la Figura 2.14). Así, se puede asegurar que parte de la información aprendida se elimina al azar y se reduce el riesgo de *overfitting*.

De forma general, las neuronas que se encuentran cercanas tienden a aprender los mismos patrones y esto puede llegar a causar un sobreajuste durante el entrenamiento, con una capa *dropout* esta correlación entre neuronas se disminuye.

El parámetro con el que se define va desde 0 hasta 1 y permite establecer la probabilidad de que cada neurona se mantenga activada. Es posible establecer valores diferentes por cada capa dependiendo de los requerimientos de cada una.

Es importante tener en cuenta que esta capa solo se utiliza durante la fase de entrenamiento.

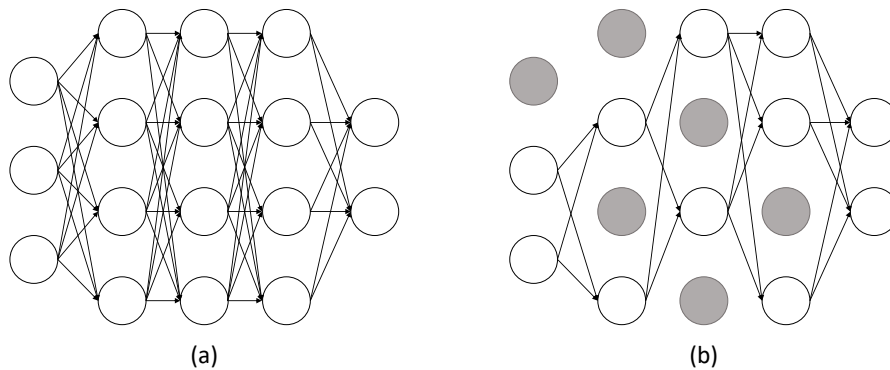


Figura 2.14: Ejemplo de *dropout* en una red de cinco capas de neuronas. (a) Red neuronal estándar y (b) red neurona después de aplicar *dropout*.

2.3.5 Arquitectura convolucional

Además, es importante conocer que la última capa de una arquitectura CNN convierte los datos en un vector unidimensional mediante la aplicación de una capa de *flatten* que se encarga de reorganizar el conjunto de datos. Después de esto, se emplea una capa conocida como *fully connected* que aplica una convolución 1D que transforma el número de elementos al número de clases existentes. Normalmente, se utiliza la función de activación *softmax* en esta última capa ya que permite obtener un mapa de probabilidades, como se ha expuesto anteriormente.

A continuación, se muestra y se representa una primera etapa de dos bloques convolucionales y un *top model* con una capa oculta. Como se puede apreciar en la Figura 2.15, tras el banco de filtros K_1 y la capa *MaxPool*, se obtiene como resultado K_1 imágenes con las dimensiones a la mitad de la original. Este grupo de imágenes son la entrada al segunda bloque convolucional donde se tiene otro banco de filtros K_2 de profundidad K_1 . Tras aplicar la segunda capa *MaxPool*, el tamaño se habrá reducido $1/4$.

La entrada al *top model* es un tensor de dimensiones $N/4 \times M/4 \times K_2$ que se serializa (*flatten*), es decir, se construye un array unidimensional con el mismo número de píxeles, K_3 (véase Figura 2.16), de manera que cada píxel es la entrada de una neurona. Posteriormente, se aplica una capa *fully connected* que permite obtener C_Q salidas correspondientes a las diferentes clases existentes.

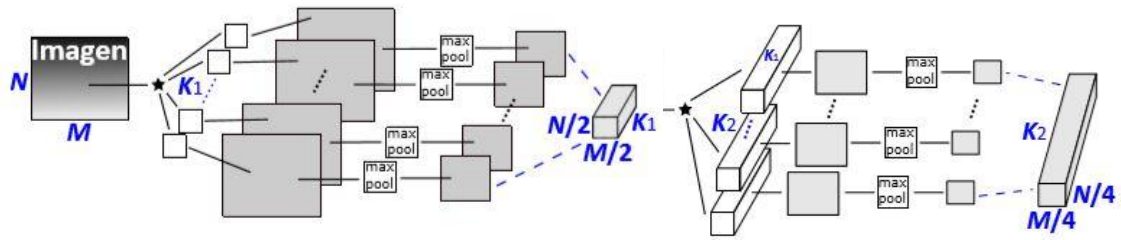


Figura 2.15: Procesado de la imagen de entrada en la 1ª fase convolucional cuya salida es la entrada de la 2ª fase convolucional [18].

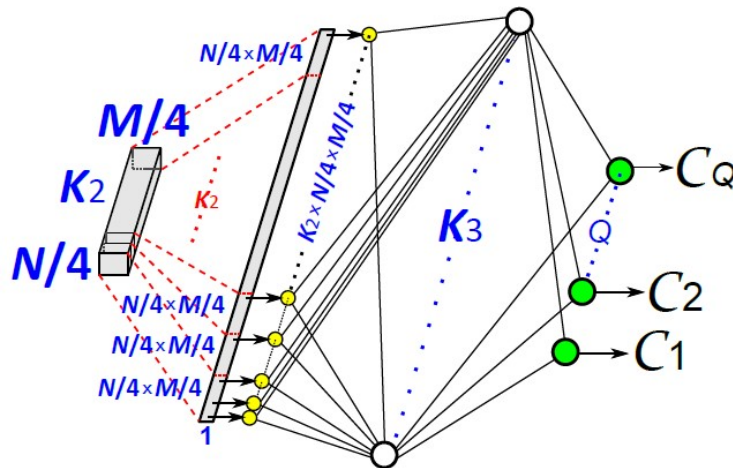


Figura 2.16: Etapa neuronal *top model* [18].

2.4 Arquitectura Encoder-Decoder

Un *autoencoder* es una arquitectura neuronal cuyo objetivo es producir una salida lo más similar posible a nivel de tamaño al ejemplo de entrada. Por tanto, el vector de salida posee las mismas dimensiones que el vector de entrada y el rango de cada elemento de salida debe corresponder con los rangos de los elementos de entrada. Es una arquitectura idónea si se quiere preservar la dimensionalidad.

Los ejemplos o muestras son imágenes y se considera cada píxel como un atributo, por ello, se necesitan tantas neuronas como píxeles haya en las imágenes que se van a tratar. A continuación, se presenta en la Figura 2.17 un esquema gráfico de un *autoencoder*. La capa oculta central es conocida como “cuello de botella” y separa la fase *encoder* de la fase *decoder*. Además, la arquitectura es simétrica respecto a esta capa.

De esta forma, las dos propiedades básicas de las arquitecturas clásicas de un *encoder-decoder* son:

- Toma el *input* y reduce su dimensionalidad al pasar a través de las capas del *encoder*. Se trata de una etapa similar a adquirir las componentes principales. Al final de la etapa de codificación, la entrada se habrá reducido a una representación de entidad lineal.

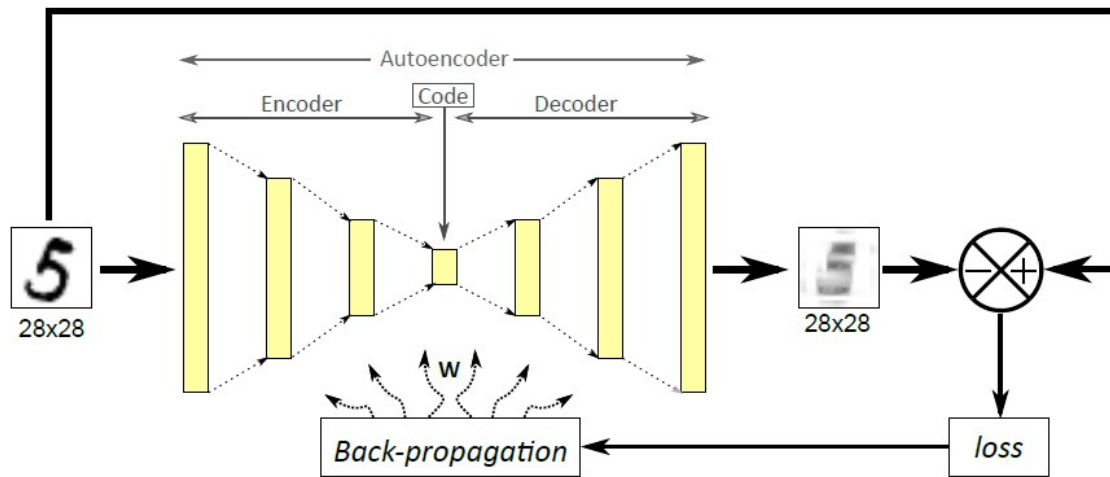


Figura 2.17: Ejemplo de arquitectura básica y su entrenamiento [18].

- A continuación, la entidad lineal aumenta su dimensionalidad debido a la acción del decodificador. De modo que en el extremo opuesto del *decoder*, el resultado es del mismo tamaño que la entrada.

Anteriormente, se ha expuesto cómo se construye una arquitectura convolucional en la cual se reduce el tamaño de la imagen según el ejemplo va atravesando las capas de los diferentes bloques convolucionales. Al terminar, se obtiene un gran volumen de datos que se transforma a un vector para alimentar a una red donde se realiza la tarea específica. Al emplear *autoencoders convolucionales*, dicha red es la etapa de decodificación (*Decoder*) que permite obtener una imagen del mismo tamaño a la salida. Existen diferentes maneras de construir decodificadores:

- **Capa *UpSampling*.** Capa de interpolación de nuevos píxeles. Se debe especificar el modo en el que se rellenan los píxeles nuevos y el tamaño de la vecindad resultante.
- **Capa convolucional transpuesta.** Trata de reconstruir la resolución de la imagen a partir de otra obtenida mediante convolución. Esta operación es parametrizable, al igual que la convolución, y por ello, es posible encontrar la mejor forma de reconstruir mediante entrenamiento. Como se ha desarrollado, la convolución de una imagen con un filtro se puede representar como una multiplicación matricial. Si se quiere invertir el efecto y aumentar la resolución se debería emplear como matriz de convolución un filtro cuyas dimensiones sean las transpuestas.

Por tanto, la fase *encoder* está formada por bloques convolucionales como los detallados en la arquitectura convolucional y la fase *decoder* está formada por bloques de deconvolución compuestos por una capa de convolución transpuesta, una capa de activación y una capa de *UpSampling*.

Debido a que el *autoencoder* trata de reconstruir el ejemplo de entrada y lo compara con la salida del mismo, se debe tratar como un problema que tendrá tantas funciones como atributos tenga cada muestra, es decir, tantas funciones como píxeles haya en la imagen.

Por otra parte, es importante la elección de la función de activación de las neuronas. Como la profundidad de un *autoencoder* es siempre el doble que la de una red convolucional simple, se

recomienda ampliamente en la literatura (como en [24] y [25]) el uso de activaciones que eviten el desvanecimiento del gradiente como ReLU.

Desde el punto de vista de *deep learning*, la segmentación consiste en asignar una determinada clase a cada uno de los píxeles de la imagen. Por tanto, la salida es una imagen del mismo tamaño que la original pero con un rango de valores distinto. En la Figura 2.18, se puede apreciar un ejemplo de una arquitectura *encoder-decoder* para segmentación.

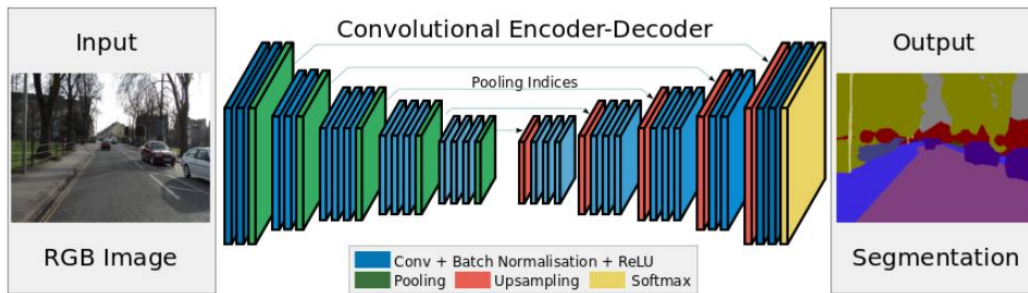


Figura 2.18: Ejemplo de arquitectura *Encoder-Decoder* para segmentación de imagen [24].

Para esta aplicación, es necesario un conjunto de datos etiquetado, *ground truth*, ya que la salida generada por la red se compara con la salida deseada, que consiste en las imágenes de entrenamiento segmentadas manualmente.

Capítulo 3

Estado del arte

El proceso diagnóstico del cáncer de próstata es una tarea tediosa que aún se sigue realizando de manera manual. Por ello, se está investigando ampliamente en el desarrollo de un sistema completo de clasificación en los diferentes grados de severidad para poder ayudar a los especialistas a reducir la subjetividad a la que está sujeta esta tarea.

En la literatura, se pueden encontrar diferentes enfoques para abordar el problema en cuestión. En concreto, se observan principalmente tres técnicas diferentes que los autores realizan de forma previa a la clasificación [26]: construcción de parches, detección de *regions of interest* (ROIs) y segmentación glandular.

La construcción de parches consiste en dividir la imagen completa y a partir de las resultantes, obtener la información necesaria. Es un enfoque sencillo ya que no requiere ningún paso previo de detección de zonas de interés para evaluar la imagen y extraer características. Este enfoque se puede observar en el artículo [27], donde emplean un protocolo de ventana deslizante de 512×512 con un 10 % de solape, y en estudios más recientes como [28] donde los autores emplearon regiones de la imagen de diferentes dimensiones.

La técnica basada en identificar ROIs permite extraer características más específicas ya que se centran en las zonas que poseen información más relevante lo que conlleva entrenar el modelo con características más discriminantes. El principal inconveniente es que requiere una identificación manual o automática previa. Sin embargo, es una de las estrategias más empleadas en la literatura, por ejemplo en [29] donde los autores emplean *ROI patches* de tejido etiquetado, y en [30] que desarrolla un sistema de clasificación utilizando *patches* con múltiples anotaciones con la finalidad de entrenar una CNN.

Actualmente, existen pocos estudios en comparación en los que los modelos predictivos utilicen glándulas en vez de regiones de interés. Ambos métodos son similares, pero ahora solo un tipo de estructura es segmentada a lo largo de la imagen completa, las glándulas. Es importante destacar que en la literatura, normalmente, se encuentran artículos con un enfoque *end-to-end* y, por ello, no suelen aportar resultados cuantitativos de la fase de segmentación. Un ejemplo de este enfoque se encuentra en [31], donde emplean técnicas texturales para segmentar las glándulas prostáticas.

Otro estudio que demuestra el prometedor futuro de este abordaje es [32], donde se utilizan 189 características basadas en descriptores de color y de textura para segmentar las glándulas.

Una estrategia empleada en el estado del arte para segmentar unidades glandulares es utilizar el lumen como punto de partida como en [26] donde luego aplican la técnica de *watershed*. Contrariamente, también se pueden encontrar algoritmos como el de [33] donde para segmentar las glándulas emplean la unión de los núcleos que hay alrededor del lumen obteniendo un índice Jaccard de 0.66.

Siguiendo la línea de la segmentación glandular prostática, es posible encontrar artículos que aplican métodos muy diversos, por ejemplo en [34] y [35], para llevar a cabo la segmentación siguen una serie de pasos: clasificación bayesiana de las estructuras glandulares, detección del lumen, extracción de los lúmenes falsos basándose en las estructuras circundantes, encontrar los lúmenes correctos y segmentación de los límites. Una vez han segmentado sus estructuras glandulares obtienen un índice Jaccard de 0.43.

Igualmente, en el artículo [36] llevan a cabo la segmentación glandular de forma previa a la clasificación. En primer lugar, se realiza una identificación de los candidatos a glándula, a los que luego se les aplica la técnica *Locally Constrained Watershed Transform* (LCWT) que permite delimitar las glándulas prostáticas obteniendo un índice Jaccard de 0.73.

Asimismo, el artículo [37] del mismo autor que el [33], emplea otro método muy interesante para la segmentación glandular: en primer lugar, realiza una clasificación de los píxeles realzando sus colores y empleando el método KNN, a continuación, utiliza los conocimientos sobre límites y distribución estructural para segmentar los bordes y, finalmente, realiza una expansión a partir del lumen para unificar el citoplasma y los núcleos.

Finalmente, el autor del artículo [38] utiliza el espacio de color *CIE Lab* a la hora de llevar a cabo la segmentación glandular, ya que en las imágenes de luminancia las glándulas aparecen como regiones contiguas de píxeles de alta intensidad circunscritas por límites definidos y pronunciados. El método aplica filtros gaussianos a este canal de color y obtienen un índice Jaccard de 0.31.

Por otra parte, se realiza un estudio de la literatura de autores que realicen segmentación en imagen histopatológica de otros tejidos o con otros objetivos. Por ejemplo, en [39] se desarrolla un método de segmentación y clasificación para distinguir entre tejido epitelial y estroma en imagen de colon y pecho. Para ello, emplea una arquitectura de CNN profunda para extraer características complejas y se demuestra la precisión de la misma al compararla con características extraídas manualmente.

También, se puede encontrar métodos en los cuales utilizan, junto con la imagen histopatológica, información inmunohistoquímica (IHC, del inglés - Immunohistochemistry) del tejido. Por ejemplo, en [40] quieren segmentar tejido epitelial prostático entrenando una primera red CNN profunda con el subconjunto de datos de IHC, las segmentaciones producidas se transfieren a la imagen H&E y se utilizan para entrenar la red final obteniendo un índice Jaccard de 0.72.

En general, las técnicas de segmentación se benefician al incorporar conocimiento previo sobre las estructuras, como en [41]. Sin embargo, esto tiene la limitación de que dependen de dicha información para detectar cada componente glandular y esto es susceptible a variaciones tanto biológicas como del proceso de adquisición. En [42], se trata de resolver este problema mediante el uso de una red profunda con objetivo de producir una segmentación glandular topológicamente

pausable y de alta resolución a partir de imagen histopatológica. Para ello, diseñan una función de pérdida con penalización específica que codifique los límites de suavidad y las relaciones entre etiquetas obteniendo un coeficiente Dice de 0.8.

Este proyecto se centra en la segmentación glandular mediante técnicas de *deep learning* para su posterior clasificación mediante métodos de *machine learning*. Se realiza un enfoque glandular y, por ello, requiere de una segmentación manual previa. Esta manera de llevar a cabo la segmentación glandular es bastante novedosa, además de que requiere un gran coste manual y, debido a esto, a penas es posible encontrar artículos que empleen esta método para tejido prostático. Por este motivo, se amplía la búsqueda bibliográfica a segmentación con algoritmos de *deep learning* para diferentes finalidades.

Las investigaciones recientes en visión computacional y reconocimiento de patrones han subrayado las capacidades de las redes neuronales convolucionales, logrando rendimientos de vanguardia en tareas difíciles como la clasificación, segmentación y detección de objetos. En el artículo [43], se presenta una novedosa segmentación 3D para procesamiento de volúmenes de imágenes prostáticas con resonancia magnética (MRI). Se propone maximizar una función diseñada específicamente para segmentación de imagen médica basada en el *Dice coefficient*. Presenta una CNN que realiza las sucesivas convoluciones con el objetivo de extraer características de los datos y formula cada etapa, de manera que aprende una función residual. Finalmente, los mejores resultados los consigue con la red V-Net y la función de pérdidas basada en Dice obteniendo un Dice medio de 0.86.

Por último, los mejores resultados en segmentación los obtiene [24] de la Universidad de Cambridge. Se trata, también, de un algoritmo *encoder-decoder*, donde la primera parte es idéntica a la red VGG16 [44] y la segunda parte consiste en una jerarquía de decodificadores, de forma que, los índices de agrupamiento son transferidos del codificador al decodificador mejorando la resolución de la segmentación. Cada uno de estos, utilizan los índices de agrupación máxima (*max-pooling*) que reciben de su correspondiente codificador para realizar el muestreo de sus mapas de características.

En conclusión, se pueden encontrar una gran diversidad de abordajes y técnicas diferentes para llevar a cabo segmentación. En este TFM, se ha decidido utilizar una red convolucional *encoder-decoder* utilizando la función Dice como función de pérdida. Se puede considerar este abordaje muy novedoso dentro de la segmentación glandular en tejido prostático, ya que no es posible encontrar otros autores que se enfrenten al mismo problema con esta técnica.

Material y método

4.1 Material

4.1.1 Base de datos

La base de datos de imágenes utilizada para el desarrollo de este proyecto se obtiene a partir de imágenes histopatológicas de biopsias de próstata procedentes del departamento de Anatomía Patológica del Hospital Clínico Universitario de Valencia. Las muestras se encuentran digitalizadas y almacenadas en una base de datos privada donde únicamente tiene acceso el personal autorizado de las diferentes instituciones que participan en el proyecto. Es necesaria la privacidad de la base de datos para garantizar la confidencialidad del paciente, los cuales firman un consentimiento informado con la finalidad de proteger sus derechos y su seguridad.

En este TFM, se utilizan imágenes de muestras biopsiadas de diferentes pacientes, 8 de los cuales aportan imágenes de tejido sano, mientras que el resto, de tejido patológico. Cabe destacar que de las imágenes patológicas, solo se han empleado aquellas etiquetadas por los especialistas de anatomía patológica como grado 3 ya que la finalidad de este proyecto es discriminar y clasificar glándulas sanas y glándulas patológicas en su etapa más inicial, el grado 3 (según la escala de *Gleason*). De esta forma, el número de pacientes y el número de imágenes originales (*slides*) empleados en este trabajo se divide como se observa en la Tabla 4.1.

	Tejido benigno	Tejido patológico de grado 3
Nº pacientes	8	21
Nº imágenes originales	17	30

Tabla 4.1: Número de pacientes y de imágenes originales de cada tejido empleadas en este proyecto.

Las muestras obtenidas en la biopsia se escanearon para su posterior análisis digital mediante un microscopio especializado disponible en el Hospital Clínico que permite capturar imágenes que alcanzan dimensiones de $[50000, 90000] \times [90000, 190000]$ píxeles y ocupan espacio de almacenamiento desde 300 hasta 1.500 MB (1,5 GB). Debido a su alta resolución espacial y sus elevadas dimensiones, se hizo uso de una librería llamada *OpenSlide* que proporciona una interfaz sencilla para leer estas imágenes.

Además, dicha librería resuelve problemas existentes asociados a las librerías estándares: el hecho de que éstas se encuentran diseñadas para leer imágenes que puedan ser descomprimidas en la RAM, y que no existe un formato universal para trabajar con estos datos. En general, es muy importante conocer el tamaño y la resolución espacial de las imágenes que se van a tratar, se encuentran relacionados con los formatos de imagen utilizados y con el aumento óptico aplicado.

Debido a sus enormes dimensiones, al digitalizar las imágenes para crear la base de datos, se realiza el muestreo con el máximo nivel de *zoom*, 40x, con el fin de conseguir una resolución espacial lo más alta posible. Luego, se remuestrearon a 10x para su análisis digital, ya que con este aumento óptico es posible ver las estructuras glandulares en su totalidad e incluso encontrar un cierto número por cada imagen. Aunque la resolución no sea óptima, es necesario para poder realizar la segmentación glandular y la posterior clasificación del tejido en una clase u otra. En conclusión, se consiguen fragmentos pequeños ($1024 \times 1024 \times 3$ píxeles) de la imagen que permiten el estudio de las regiones de interés.

La finalidad de esta etapa del trabajo es obtener imágenes histopatológicas con un aumento de 10x a partir de las imágenes digitalizadas con un aumento de 40x. El procedimiento consiste en:

1. Partiendo de una base de datos original de *slides* completas como la que se muestra en la Figura 4.1.a, se realiza un paso previo de preprocesado para identificar la *bounding box*¹ que contiene la región de tejido y, de esta forma, eliminar las zonas de píxeles sin información útil.
2. Una vez identificada la región de interés (ROI), se implementa un protocolo de ventana deslizante para trabajar con imágenes de tamaño menor, Figura 4.1.b, que permiten mejorar en términos de resolución e información local. En concreto, se divide en imágenes de 1024×1024 píxeles (Figura 4.1.c), como se ha comentado, corresponde a un aumento óptico de 10x.
3. A continuación, se lleva a cabo otro paso por el cual se descartan aquellos *patches* con menos de un 5% de píxeles de tejido con la finalidad de garantizar un mínimo de información útil (Figura 4.1.d). Con esto se consigue que en los *patches* de 1024×1024 píxeles se pueda encontrar por lo menos una unidad glandular.

Además, hay que tener en cuenta que la cantidad de información disponible de cada clase de tejido prostático debe estar balanceada, por ello, se implementa un algoritmo de ventana deslizante con un solapamiento del 50% a la hora de tomar las subimágenes. Con esto, se consigue añadir información de aquellas glándulas que quedan entre una imagen y la siguiente.

¹Se trata del rectángulo más pequeño que puede contener la zona seleccionada.

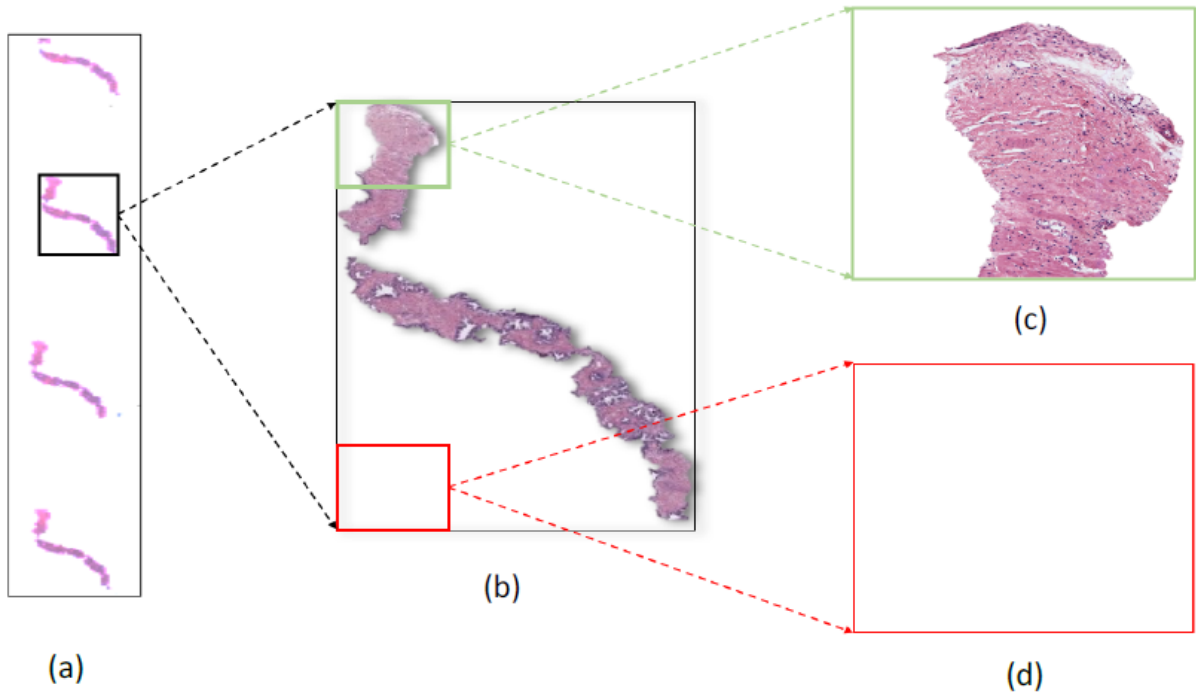


Figura 4.1: (a) Corte histológico de tejido prostático benigno; (b) *Bounding box* que alberda una de las estructuras tisulares; (c) *patch* de 1024×1024 píxeles con información relevante; (d) *patch* de 1024×1024 píxeles sin información de interés [45].

En conclusión, esta etapa previa de preparación de la base de datos permite conseguir un número de unidades glandulares balanceado entre ambos tejidos. Finalmente, para este proyecto se cuenta con:

	Tejido benigno	Tejido patológico de grado 3
Nº de <i>patches</i> 1024×1024	591	391
Tejido seleccionado (%)	5	0
Solapamiento (%)	0	50

Tabla 4.2: Número de *patches* de 1024×1024 , porcentaje de tejido mínimo seleccionado por cada imagen y porcentaje de solapamiento implementado.

Cuando se han obtenido las imágenes a dimensiones que permitan su estudio, se preparan las imágenes patológicas de grado 3 mediante la participación de especialistas en anatomía patológica. Los médicos especializados emplearon la aplicación que se les facilitó, *MicroDraw*, para anotar las áreas relevantes en la investigación. Esta aplicación, realizada en JavaScript, se desarrolla sobre una librería de código abierto que posibilita la visualización sin pérdida de calidad y facilita el etiquetado de imágenes [46]. Gracias a las imágenes de etiquetas proporcionadas por esta aplicación es posible eliminar las zonas que no aportan información relevante para el trabajo.

Realizadas las anotaciones por parte de los especialistas, se genera la base de datos privada donde se almacenan las imágenes sanas y patológicas de grado 3.

Finalmente, se realiza una segmentación manual y se crean máscaras de las correspondientes imágenes donde se encuentran al máximo nivel de intensidad las glándulas que se desean segmentar. Posteriormente, se explica con mayor profundidad la creación del *ground truth*.

4.1.2 Software y hardware

4.1.2.1 Software

Para el desarrollo y la implementación de los algoritmos de preprocesado y postprocesado de las imágenes, así como para el cálculo de los resultados, se ha empleado la plataforma de MATLAB[®] v.R2019a, de The MathWorks, Inc. (Natick, Massachusetts, Estados Unidos). Este software combina un entorno de escritorio perfeccionado para el análisis iterativo y los procesos de diseño con un lenguaje de programación que expresa las matemáticas de matrices y vectores directamente. MATLAB[®] resulta de gran utilidad para numerosas aplicaciones como analítica de datos, comunicaciones inalámbricas, aprendizaje profundo, procesamiento de señales, visión artificial, finanzas, robótica y sistemas de control [47].

En este TFM, se emplea una *app* que incluye la plataforma MATLAB[®], se trata de una las herramientas de procesamiento de imagen llamada *Image Labeler*. Esta aplicación proporciona una manera sencilla de marcar las ROI con etiquetas rectangulares, polilíneas, etiquetas de píxeles o de secuencias de imágenes.

Para la etapa de entrenamiento y predicción de redes convolucionales (CNN) basadas en *deep learning*, se emplea el lenguaje de programación Python3.5 mediante el uso de PyCharm 2019.1.1 como entorno de desarrollo integrado (IDE). Además, se hace uso de la librería *NVIDIA Cuda*[®] *Deep Neural Network* y de la librería *OPENCV*. Finalmente, a lo largo de este proyecto se ha empleado la biblioteca de *Keras* como *framework* usando *TensorFlow* como *backend* para implementar, entrenar y probar diferentes tipos neuronales con diferentes parámetros.

4.1.2.2 Hardware

Es importante conocer las características internas del hardware en el que se ha llevado a cabo el proyecto, ya que de esto depende tanto el funcionamiento como la velocidad y el rendimiento que se puede alcanzar al compilar y ejecutar los distintos algoritmos.

Por una lado, este proyecto ha sido desarrollado en el ordenador HP Envy 13-ah0006ns, cuyas especificaciones son las siguientes:

Sistema operativo:	Windows 10 Home 64 bits
Procesador:	Intel [®] Core(TM) i7-8550U CPU
Velocidad:	1.80 GHz
Memoria del sistema:	8 GB
Disco duro:	SSD 256 GB
Tarjeta gráfica:	NVIDIA GEFORCE MX150

Tabla 4.3: Especificaciones del hardware empleado.

Por otro lado, la implementación de las técnicas desarrolladas requiere de una gran complejidad computacional y por ello, para llevar a cabo la etapa de entrenamiento de las redes convolucionales se hizo uso de un conjunto de procesadores con unas prestaciones altas, 32GB de RAM y tarjetas gráficas NVIDIA Tital V 3300. Además, se dispone de un servidor de 16 TB de capacidad, Synology DS46, donde se alojan los datos necesarios para el desarrollo del proyecto y la documentación relativa a ellos.

4.2 Metodología

El esquema que se ha seguido en el presente proyecto se puede observar en la Figura 4.2.

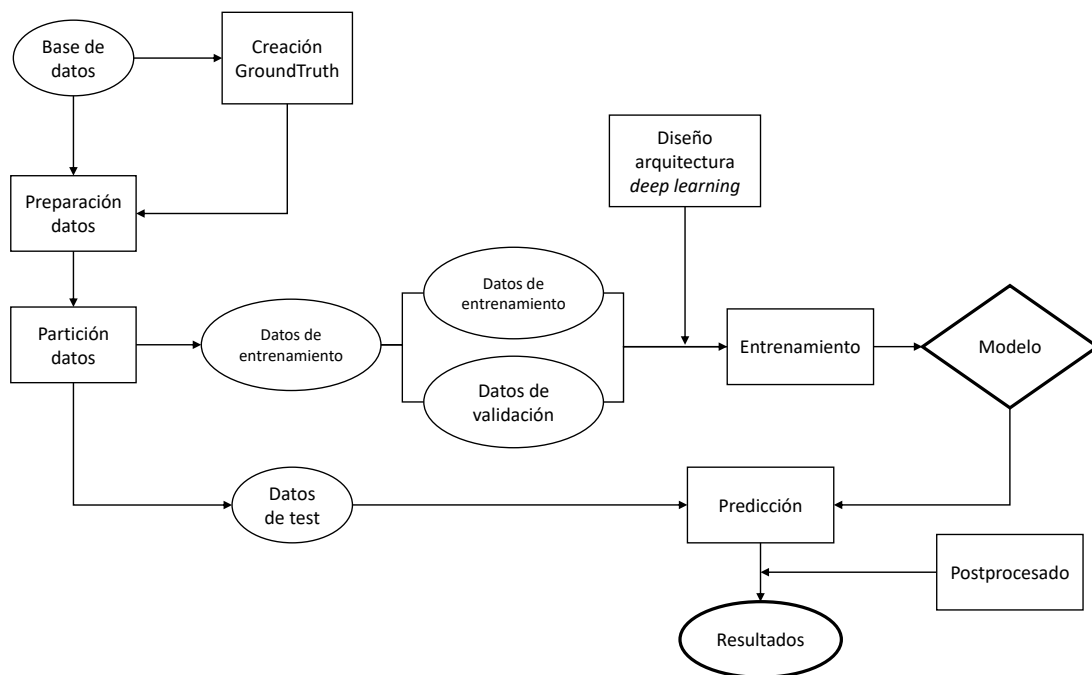


Figura 4.2: Diagrama representativo del procedimiento seguido en este TFM.

4.2.1 Creación Ground truth

La construcción del *groundtruth* para resolver el problema de segmentación automática se basa en etiquetar manualmente las glándulas presentes en cada imagen histopatológica con el objetivo de entrenar el modelo de *deep learning* y, posteriormente, evaluarlo. En este proyecto, la segmentación manual se ha llevado a cabo personalmente en base al conocimiento aportado por los patólogos del Hospital Clínico de Valencia (HCV) y al adquirido durante el grado y el máster de Ingeniería Biomédica.

Para realizar la segmentación manual, se ha hecho uso de una aplicación del software MATLAB[®], *Image Labeler app*, que proporciona una manera fácil de marcar etiquetas de región rectangular de interés, etiquetas mediante polilínea, etiquetas mediante píxeles y etiquetas de escena en una secuencia de vídeo.

En primer lugar, se cargan los datos a partir de una carpeta de imágenes y seguidamente, se define la etiqueta que se quiere emplear, en este caso, se usa la etiqueta de píxel. El procedimiento que se sigue consiste en rodear con un lápiz azul las glándulas que aparecen en las imágenes histopatológicas de $1024 \times 1024 \times 3$. En la Figura 4.3 se puede apreciar el ejemplo de la segmentación glandular de una imagen histopatológica mediante *Image Labeler* app.

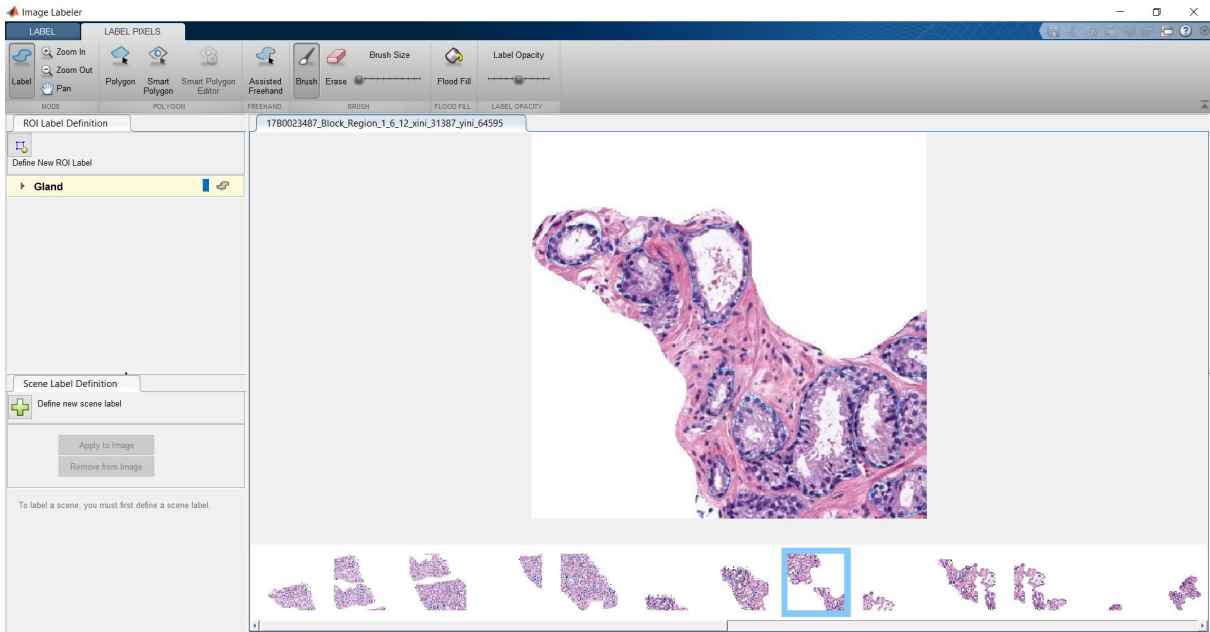


Figura 4.3: Ejemplo de segmentación glandular con la aplicación *Image Labeler* de MATLAB[®].

Una vez realizada la segmentación manual mediante dicha *app*, se obtienen imágenes con los bordes de las glándulas segmentados y, a partir de ellas, se generan máscaras de las glándulas mediante un algoritmo implementado en MATLAB[®].

Nótese que este procedimiento es un trabajo muy tedioso que requiere una gran cantidad de tiempo de trabajo manual.

4.2.2 Preparación de la base de datos de imágenes

La entrada de la red neuronal de este proyecto es la base de datos de imágenes que a la hora de entrenar el modelo resulta en un gran coste computacional si las imágenes tienen tamaños relativamente grandes. Por ello, es necesario el parcheado de las imágenes de 1024×1024 píxeles en subimágenes de 256×256 píxeles.

La forma de llevar a cabo este proceso es partiendo las imágenes en 16 parches como se puede observar en la Figura 4.4.

A continuación, se lleva a cabo una fase de balanceo entre clases, como se ha comentado anteriormente, es necesario mostrar a la red lo que se quiere conseguir para que ella aprenda de los ejemplos. En este TFM, se quiere conseguir segmentar las glándulas de las imágenes y por ello, se trata de distinguir entre fondo y clase.

Cuando la imagen tiene muchos más píxeles de una clase que de otra, un problema habitual que surge a la hora de entrenar una red neuronal profunda es que da mayor peso a la clase mayoritaria.

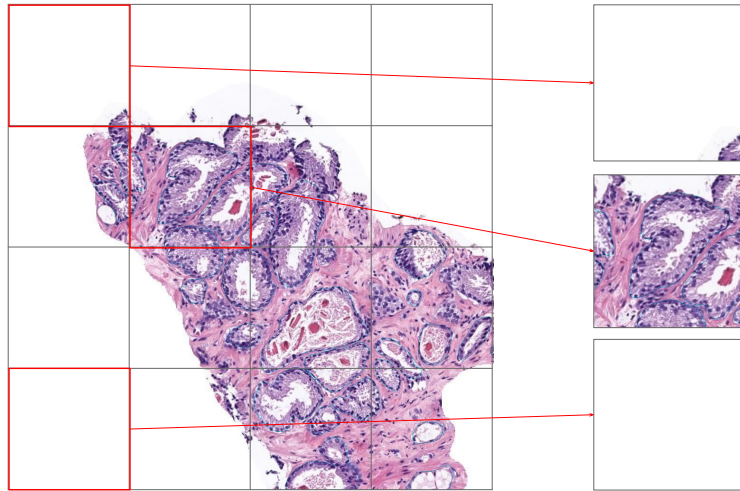


Figura 4.4: Ejemplo de parcheado de una imagen.

En el ejemplo concreto para distinguir glándula y fondo, la red tiende a ponderar más altos los pesos de la clase fondo. Para solucionar este problema existen diferentes métodos como:

- *Class weights*. Consiste en asignar un valor más alto a las instancias de la clase minoritaria en la función de pérdidas. De forma que cada muestra de la clase minorita se trata como x muestras de la clase mayoritaria.
- Eliminar parches. Es la opción escogida en el presente trabajo. Consiste en eliminar aquellos parches que contienen menos de cierto porcentaje de glándula respecto de fondo. En la Figura 4.5, se puede apreciar un ejemplo, donde el último parche se eliminaría y el primer parche se eliminaría en función del porcentaje de glándula especificado. En este TFM, se eliminan aquellas imágenes que poseen menos de 10% de glándula.

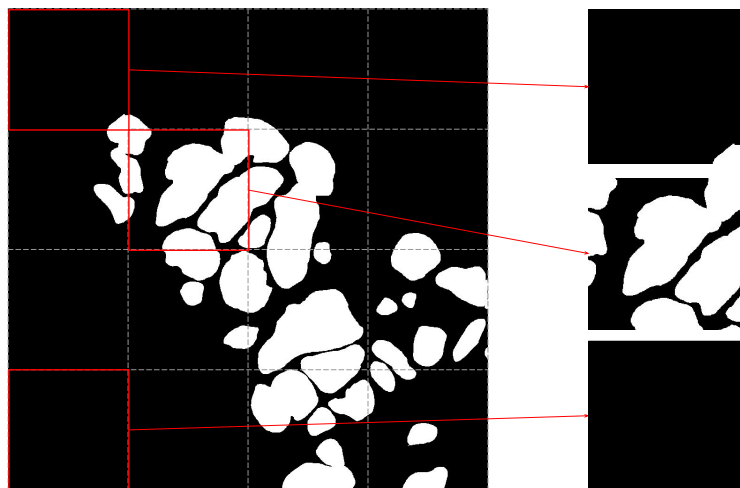


Figura 4.5: Ejemplo de parcheado del *ground truth* de la imagen anterior.

Cabe destacar que esta técnica solo se aplica a los datos que se utilizan para entrenar cada modelo, mientras que para testear el modelo se emplean todos los parches de cada imagen.

4.2.3 Partición datos

Una vez preparado el *ground truth* es necesario llevar a cabo una partición de nuestro conjunto de datos para poder realizar el entrenamiento de la red neuronal y el posterior testeo de la misma. Con el fin de garantizar la robustez del modelo implementado, se realiza un *k-fold cross-validation* y además, se dividen los datos en función de los pacientes, de modo que todas las *slides* de un paciente se encuentran en un mismo *set*.

En primer lugar, se estudia el número de subconjuntos de datos más óptimo en base a las posibilidades que ofrecen los diferentes pacientes y, así, se estima que para poder dividir en función del número de pacientes, y que el número de imágenes en cada *set* sea lo más parecido posible, el mejor número de subconjuntos es $k = 4$.

Una vez seleccionado el *4-fold cross-validation*, se decide en qué subconjuntos se distribuyen los diferentes pacientes y se crean los 4 *sets*. En cada iteración se utiliza un conjunto de entrenamiento de dos *sets*, un subconjunto de validación para el procedimiento de aprendizaje y un subconjunto de testeo para la posterior evaluación del modelo con datos nuevos (Véase la Figura 4.6). Cabe destacar que la entrada de la red son subconjuntos formados por parches de 256×256 con un mínimo de 10% de glándula en cada uno de ellos.

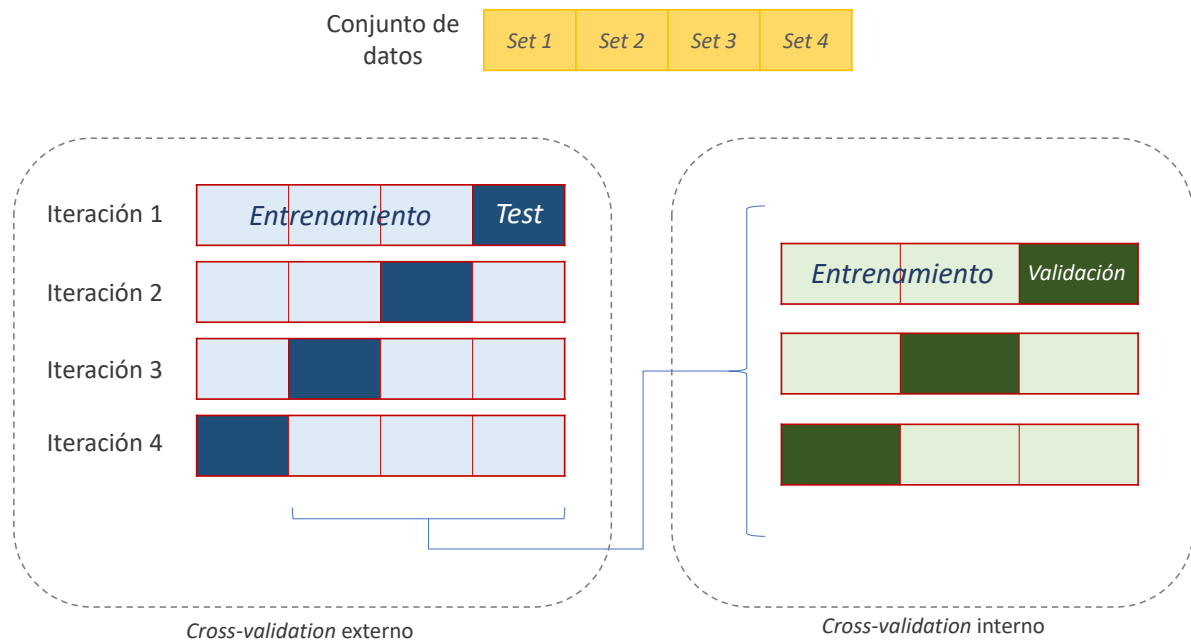


Figura 4.6: Método *4-fold cross-validation* empleado.

4.2.4 Entrenamiento

El problema de aprendizaje de una red neuronal se formula en términos de minimización de la función de pérdidas asociada para obtener los pesos óptimos para los filtros de las capas convolucionales. El valor de esta función depende totalmente de los parámetros de la red neuronal: los pesos sinápticos y los bias asociados. Estos a su vez dependen de la arquitectura del modelo, de los hiperparámetros escogidos y de la base de datos empleada.

Inicialmente, al diseñar una arquitectura CNN la red no tiene ningún conocimiento y, en este caso, los pesos establecidos son totalmente aleatorios. Dicho conocimiento se irá adquiriendo a partir de los ejemplos proporcionados con sus respectivas etiquetas. Esto, construye la base de la etapa de entrenamiento de una red neuronal.

En general, se puede dividir el procedimiento en los siguientes pasos: propagación hacia delante, función de pérdidas, retropropagación y actualización de los pesos. Cuando un conjunto de datos finaliza las etapas *forward* y *backward propagation* se dice que ha finalizado una época. En la siguiente época, el mismo subconjunto de datos lleva a cabo el mismo procedimiento intentando disminuir la función de error con los pesos actualizados de la época anterior.

La idea es que en un principio los pesos de los filtros de las diferentes capas convolucionales son aleatorios y, por tanto, no son capaces de llevar a cabo una predicción correcta en las primeras iteraciones. Las salidas de cada iteración se comparan con las muestras etiquetadas y, a partir de esta información, se realiza el aprendizaje y actualización de los pesos de los filtros de acuerdo a una función de pérdidas.

Por otra parte, hay que tener en cuenta que al entrenar una red neuronal es posible que se produzca *overfitting*, esto ocurre cuando la red neuronal aprende su conjunto de entrenamiento satisfactoriamente pero no puede generalizar sus predicciones a ejemplos que no ha visto antes. Esto se caracteriza por un bajo sesgo y una alta varianza [19]. Los métodos empleados en este TFM para evitar el *overfitting* son [19]:

- Parada temprana (en inglés, *early stopping*). Se trata de monitorear el error del conjunto de validación tras cada iteración con la finalidad de detenerla cuando la red comienza a sobreajustar los datos.
- *Dropout* o abandono al azar. Realiza una desconexión al azar de un cierto porcentaje de neuronas en cada iteración. Así, se puede asegurar que parte de la información aprendida se elimina al azar y se reduce el riesgo de sobreajuste. Se ha detallado anteriormente.

4.2.4.1 Arquitectura del modelo

El diseño del modelo del presente TFM se ha inspirado en la arquitectura *encoder-decoder* validada por numerosos trabajos como [24] y [48]. La finalidad de esta arquitectura es facilitar la reconstrucción de la segmentación de forma gradual por parte de la red *decoder* empleando las características extraídas por la red *encoder*. Por tanto, en este tipo de arquitecturas el decodificador es entrenado de manera que, haciendo uso de la información comprimida, sea capaz de reconstruir una nueva señal [49]. Véase un esquema gráfico en la Figura 4.7.

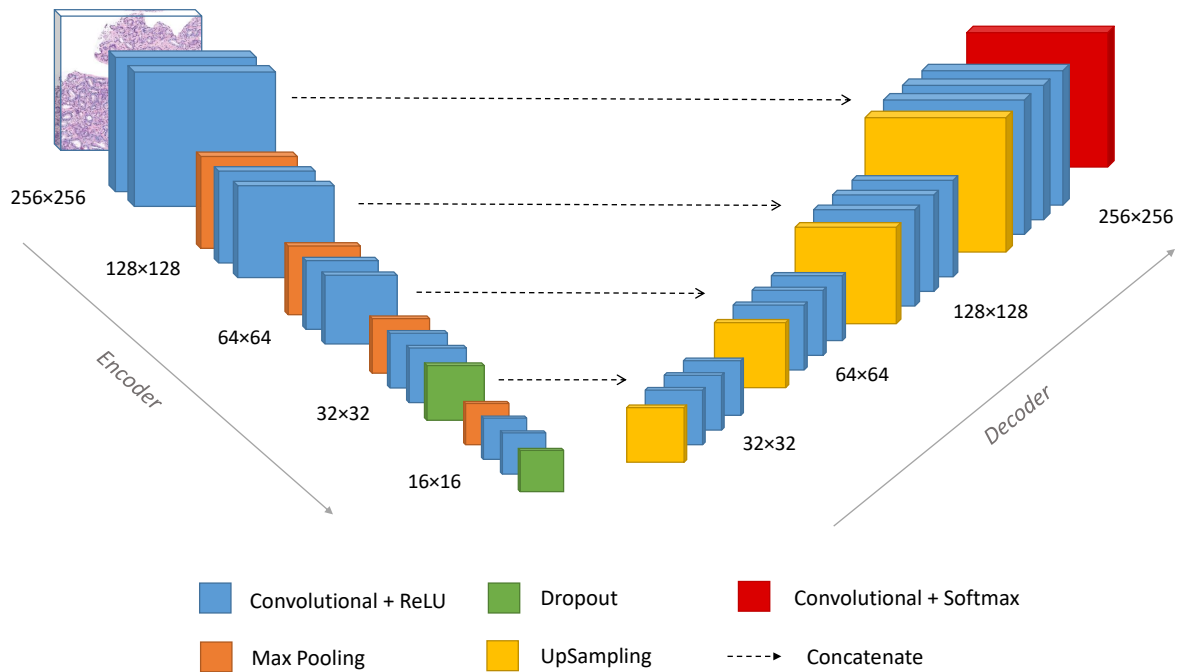


Figura 4.7: Esquema ilustrativo de la arquitectura empleada en este TFM. Las dimensiones son las correspondientes a la salida de cada capa.

Todas las capas convolucionales de la etapa *encoder* aplican un *kernel* 3×3 mientras que los de la etapa *decoder* aplican un *kernel* 2×2 , varían en el número de filtros de salida de la convolución, que indica a su vez, la dimensionalidad del espacio de salida de la capa. Se utiliza un *padding same* y la función de activación *ReLU* en todas las capas convolucionales de la arquitectura salvo en la última que se emplea una función *Softmax* con el fin de obtener un mapa de probabilidad para cada clase.

Asimismo, para la capa *MaxPooling* se ha escogido $(2, 2)$ como factores de reducción de escala, de manera que, la salida producida será dimensionalmente la mitad que la entrada. Además, la capa *Dropout* aplica un porcentaje de desconexión del 50%.

Por otra parte, en la etapa *decoder* se emplean capas *UpSampling* en las que los píxeles se rellenan repitiendo las filas y columnas según los factores $(2, 2)$. Además, la última capa convolucional utiliza como función de activación, una función *Softmax* que permite obtener mapas de probabilidad para cada clase que se quiera distinguir. En este caso, se ha decidido que el modelo trate de separar entre fondo, borde y glándula.

En total, el modelo propuesto tiene 31.031.875 parámetros entrenables que se tratarán de optimizar durante la fase de entrenamiento para conseguir los mejores valores para estos.

A continuación, se muestra en la Tabla 4.4, las diferentes capas definidas con sus respectivos tamaños de salida, el número de parámetros de cada una y sus correspondientes conexiones.

<i>Layer</i>	<i>Type</i>	<i>Output shape</i>	<i>Param</i>	<i>Connected to</i>
input_1	InputLayer	(256, 256, 3)	0	
conv2d_1	Conv2D	(256, 256, 3)	1792	input_1
conv2d_2	Conv2D	(256, 256, 64)	36928	conv2d_1
max_pooling2d_1	MaxPooling2D	(128, 128, 64)	0	conv2d_2
conv2d_3	Conv2D	(128, 128, 128)	73856	max_pooling2d_1
conv2d_4	Conv2D	(128, 128, 128)	147584	conv2d_3
max_pooling2d_2	MaxPooling2D	(64, 64, 128)	0	conv2d_4
conv2d_5	Conv2D	(64, 64, 256)	295168	max_pooling2d_2
conv2d_6	Conv2D	(64, 64, 256)	590080	conv2d_5
max_pooling2d_3	MaxPooling2D	(32, 32, 256)	0	conv2d_6
conv2d_7	Conv2D	(32, 32, 512)	1180160	max_pooling2d_3
conv2d_8	Conv2D	(32, 32, 512)	2359808	conv2d_7
dropout_1	Dropout	(32, 32, 512)	0	conv2d_8
max_pooling2d_4	MaxPooling2D	(16, 16, 512)	0	dropout_1
conv2d_9	Conv2D	(16, 16, 1024)	4719616	max_pooling2d_4
conv2d_10	Conv2D	(16, 16, 1024)	9438208	conv2d_9
dropout_2	Dropout	(16, 16, 1024)	0	conv2d_10
up_sampling2d_1	UpSampling2D	(32, 32, 1024)	0	dropout_2
conv2d_11	Conv2D	(32, 32, 512)	2097664	up_sampling2d_1
concatenate_1	Concatenate	(32, 32, 1024)	0	dropout_1 conv2d_11
conv2d_12	Conv2D	(32, 32, 512)	4719104	concatenate_1
conv2d_13	Conv2D	(32, 32, 512)	2359808	conv2d_12
up_sampling2d_2	UpSampling2D	(64, 64, 512)	0	conv2d_13
conv2d_14	Conv2D	(64, 64, 256)	524544	up_sampling2d_2
concatenate_2	Concatenate	(64, 64, 512)	0	conv2d_6 conv2d_14
conv2d_15	Conv2D	(64, 64, 256)	1179904	concatenate_2
conv2d_16	Conv2D	(64, 64, 256)	590080	conv2d_15
up_sampling2d_3	UpSampling2D	(128, 128, 256)	0	conv2d_16
conv2d_17	Conv2D	(128, 128, 128)	131200	up_sampling2d_3
concatenate_3	Concatenate	(128, 128, 256)	0	conv2d_4 conv2d_17
conv2d_18	Conv2D	(128, 128, 128)	295040	concatenate_3
conv2d_19	Conv2D	(128, 128, 128)	147584	conv2d_18
up_sampling2d_4	UpSampling2D	(256, 256, 128)	0	conv2d_19
conv2d_20	Conv2D	(256, 256, 64)	32832	up_sampling2d_4
concatenate_4	Concatenate	(256, 256, 128)	0	conv2d_2 conv2d_20
conv2d_21	Conv2D	(256, 256, 64)	73792	concatenate_4
conv2d_22	Conv2D	(256, 256, 64)	36928	conv2d_21
conv2d_23	Conv2D	(256, 256, 3)	195	conv2d_22

Tabla 4.4: Arquitectura Encoder-Decoder.

4.2.4.2 Coeficiente Dice como función de pérdida

Se trata de uno de los hiperparámetros requeridos para cuantificar cómo de parecidos son los pesos de la red neuronal respecto a los ideales durante el procedimiento de aprendizaje. Existen distintos tipos de función de pérdida disponibles en la biblioteca de *keras*, la elección del mejor está íntimamente relacionado con el problema a resolver. Algunas funciones típicas son el Error Cuadrático Medio o la validación cruzada (*crossentropy*), no obstante, el coeficiente Dice es el más empleado en segmentación de imágenes y, por ello, será el empleado en este TFM.

El coeficiente Dice es un estadístico utilizado para comparar la similitud existente entre dos imágenes. Cuando se resuelve un problema de segmentación, es muy probable que exista cierto desbalanceo de clases, por ello, se utilizan métricas que evalúan el error frente al total de los datos de una sola clase.

El coeficiente Dice entre dos imágenes binarias se puede expresar de la siguiente manera [49]:

$$Dice = \frac{2 \sum_i^N p_i g_i}{\sum_i^N p_i^2 + \sum_i^N g_i^2} \quad (4.1)$$

donde N es el número total de píxeles, p_i son los valores de los píxeles generados por el modelo y g_i es el *ground truth*.

Como se ha comentado anteriormente, es necesario que la función de pérdida sea diferenciable respecto a la salida de la red para poder realizar el *backpropagation* y poder optimizar el valor de los pesos de la red. La derivada del coeficiente Dice se desarrolla de la siguiente forma:

$$\frac{\partial Dice}{\partial p_j} = 2 \left(\frac{g_j (\sum_i^N p_i^2 + \sum_i^N g_i^2) - 2p_j (\sum_i^N p_i g_i)}{(\sum_i^N p_i^2 + \sum_i^N g_i^2)^2} \right) \quad (4.2)$$

Es importante tener en cuenta que si el Dice está definido para segmentaciones binarias y la salida de la red son probabilidades, por tanto, el p_j tomará valores entre 0 y 1 mientras que g_j tomará únicamente los valores 0 y 1.

Además, para que la función Dice definida pueda actuar como función de pérdidas, se ha de minimizar y se obtiene la siguiente expresión:

$$Dice_{loss} = - \frac{2 \sum_i^N p_i g_i}{\sum_i^N p_i + \sum_i^N g_i} \quad (4.3)$$

4.2.4.3 Hiperparámetros

Los hiperparámetros son variables optimizables a la hora de entrenar una arquitectura de red neuronal. El valor de los hiperparámetros debe ser especificado por el programador para ajustar los algoritmos de entrenamiento. La optimización de estos es un reto, existen diferentes formas para escoger los mejores: desde prueba y error de manera manual a métodos algorítmicos más sofisticados. En conclusión, se trata de un conjunto de parámetros externos cuya variación puede tener un gran impacto en el rendimiento de la red [50].

Por una parte, hay algunos hiperparámetros que permiten definir la estructura y la topología de la red neuronal como el número de capas ocultas, el número de neuronas por capa, la función de activación, etc. En la subsección 4.2.4.1 se ha especificado qué hiperparámetros se han utilizado, con qué valores y de qué manera se ha hecho.

Por otra parte, se detallan a continuación los diferentes hiperparámetros a nivel de algoritmo de aprendizaje que se van a ir modificando para obtener los pesos óptimos [50]:

- Número de *epochs* o iteraciones. Este hiperparámetro indica el número de veces que el conjunto de datos de entrenamiento ha pasado por la red neuronal durante el proceso de aprendizaje. Una época ha concluido cuando un subconjunto de datos finaliza la etapa *forward-backward* y, en la siguiente época, se realizará el mismo procesamiento pero intentando disminuir la función de error con los pesos actualizados de la época anterior.
- *Batch size*. Existe una técnica que trata de particionar el conjunto de datos en lotes pequeños para pasarlos por la red. El *batch size* es el argumento que indica el número de muestras que se agrupan dentro del mismo lote. Cuando todas las muestras un *batch size* han pasado por la etapa *forward* y *backward propagation* se dice que han cumplido una época. El tamaño óptimo depende de muchos factores cómo la capacidad de memoria disponible.
- *Learning rate* o tasa de aprendizaje. Se trata de un hiperparámetro que controla cuánto se están ajustando los pesos de la red en cada iteración con respecto al gradiente de pérdida. El valor adecuado de este es dependiente del problema que se esté resolviendo, pero de forma general cuanto más bajo sea el valor, se desciende más lento y se puede caer en un mínimo local, mientras que cuanto más alto sea, se avanza más rápido pero puede que se salte el mínimo objetivo y dificultar el proceso de aprendizaje.
- *Momentum* (o impulso). Consiste en un hiperparámetro que acumula una suma decreciente de los gradientes anteriores en un vector de momento y lo usa para minimizar la función de error. Esto tiene la ventaja de acelerar el aprendizaje del descenso de gradiente a lo largo de dimensiones donde permanece relativamente consistente y, ralentizarlo a lo largo de dimensiones turbulentas donde oscila significativamente.

En conclusión, se trata de una constante entre 0 y 1 que se usa como ponderación de los gradientes anteriores. Por ejemplo, cuando se alcanza un mínimo local en vez de global, el gradiente actual se convierte en cero y el algoritmo se encuentra atascado, mediante este hiperparámetro se soluciona dicho problema. Se ha demostrado que los algoritmos que emplean *momentum* funcionan mejor en la práctica.

- Optimizador. Se trata de un parámetro necesario a la hora de compilar y entrenar el modelo. De forma general, el proceso de entrenamiento es un un problema de optimización y, por ello, necesita de un optimizador para actualizar los pesos de la red.

4.2.4.4 Optimizadores

Como se ha comentado anteriormente, para llevar a cabo el entrenamiento de la red neuronal es necesario un optimizador. Los optimizadores empleados en segmentación y que se prueban en este TFM para encontrar el que permite obtener mejores resultados son:

- Adam. Estimación del momento adaptativo. Utiliza los gradientes cuadrados para escalar la tasa de aprendizaje y aprovecha el *momentum* mediante el uso de la media móvil del gradiente, en lugar del gradiente en sí mismo.

Como se ha mencionado, es un método de *learning rate* adaptativo, calcula las tasas individuales para diferentes pesos y usa estimaciones del primer y segundo momento del gradiente para adaptar las tasas. A continuación, se pueden observar las ecuaciones que emplea este optimizador:

$$m(t+1) = \beta_1 * m(t) + (1 - \beta_1) * \nabla J \quad (4.4)$$

$$\hat{m}(t+1) = \frac{m(t+1)}{1 - \beta_1^t} \quad (4.5)$$

$$v(t+1) = \beta_2 * v(t) + (1 - \beta_2) * \nabla^2 J \quad (4.6)$$

$$\hat{v}(t+1) = \frac{v(t+1)}{1 - \beta_2^t} \quad (4.7)$$

$$W(t+1) = W(t) - \frac{\eta}{\sqrt{c(t+1) + eps}} \frac{\hat{m}(t+1)}{\sqrt{\hat{v}(t+1)}} \quad (4.8)$$

$$\eta_{efectiva} = \frac{\eta}{\sqrt{c(t+1) + eps}} \quad (4.9)$$

donde m es el primer momento, v es el segundo momento, ∇J es el gradiente en la iteración t , β son los hiperparámetros que definen al implementar el optimizador, W son los parámetros de la red y η es la tasa de aprendizaje. La tasa de aprendizaje efectiva se reduce para los coeficientes que reciben un alto gradiente y crece para los que reciben pequeños gradientes o no se actualizan frecuentemente.

El método es fácil de implementar y computacionalmente eficiente. Es invariable al cambio de escala de los gradientes y es adecuado para problemas con una gran cantidad de datos y/o parámetros.

- Adamax. Se trata de una variante de Adam basada en la norma del infinito. Es decir, la regla de actualización para los pesos individuales es escalar sus gradientes inversamente proporcionales a la norma L_∞ de sus gradientes actuales y pasados.
- Nadam. Nadam es una versión de Adam pero con un *momentum* de Nesterov. La aceleración de Nesterov se puede conceptualizar como una actualización correctiva del *momentum* que permite obtener una aproximación de dónde se encuentran los parámetros después de la actualización.

4.2.5 Reconstrucción de las imágenes

Tal y como se ha detallado en el apartado ??, ha sido necesario realizar un parcheado previo al entrenamiento para aumentar el rendimiento de la red a fin de que sea computacionalmente más eficiente. Sin embargo, los resultados han de ser analizados en base a la imagen completa y, por ello, es necesario llevar a cabo una reconstrucción de la imagen.

Los parches de cada imagen se han guardado en un tensor en el que se ha colocado un parche detrás de otro añadiendo una cuarta dimensión. La tarea en esta fase del proyecto es volver a colocar cada parche en su posición obteniendo de nuevo una imagen de 1024×1024 .

Siendo P_K de tamaño 256×256 donde K es el número de parches (16 por imagen) y I_M de tamaño 1024×1024 donde M es el número total de imágenes testeadas:

$$I_m = P_k(i, j) \quad (4.10)$$

donde $m = 1 : M$, $k = 1 : K$ y (i, j) es la posición de cada parche k en la imagen m .

4.2.6 Evaluación

Una vez testeados los modelos construidos con un *data set* que no había visto antes y hecha la reconstrucción de las imágenes, se procede a evaluar la bondad del método empleado.

En este caso, se ha optado por emplear las siguientes métricas ya que son las más adecuadas para comparar máscaras de segmentación:

1. Coeficiente *Jaccard*. Medida del grado de similitud entre dos conjuntos. Se computa como la intersección de imágenes binarias divididas por la unión de las mismas.

$$Jaccard = \frac{|A \cap B|}{|A \cup B|} \quad (4.11)$$

siendo A la imagen de etiquetas (*ground truth*) y B el *output* de la red binarizado.

2. Coeficiente *Dice*. Como se ha detallado anteriormente, el coeficiente *Dice* es la métrica elegida como función de pérdida a la hora de entrenar el modelo. También, se usará para evaluar la bondad del modelo una vez realizada la predicción. Se trata de una medida de similitud entre imágenes ampliamente usada para segmentación.
3. *BFscore* (BF). Medida de coincidencia de contorno empleada en la segmentación de imágenes. Este valor será 1 cuando los contornos de los objetos tengan una coincidencia perfecta. Por otra parte, se calcula también la precisión BF que es la proporción de píxeles de límite de la imagen predicha que están suficientemente cerca del límite en la imagen, se pueden considerar como verdaderos positivos. Finalmente, se obtiene el *recall* BF que es la proporción de puntos en el contorno de la segmentación de la imagen real que están suficientemente cerca en la segmentación predicha [51].

Resultados y discusión

Como se ha expuesto anteriormente, la finalidad de este proyecto es conseguir un sistema de segmentación automática de glándulas a partir de las imágenes histopatológicas completas sin realizar ningún procesado previo. Para ello, se han ido variando los diferentes hiperparámetros presentados en la subsección 4.2.4.3 y se han seleccionado aquellos que permitían mejores resultados al contrastarlos con los subconjuntos de validación. Se ha escogido el optimizador NADAM al que se le ha aplicado un learning rate de $1e-5$, se ha utilizado un *batch size* de 16 puesto que es el valor óptimo a nivel computacional y el número de épocas que se ha entrenado el modelo es de 500. Además, como método para evitar el *overfitting*, se ha empleado un *early stopping* de 10 épocas.

Con el objetivo de conseguir un sistema robusto se ha realizado un abordaje de *4-fold cross-validation*. En cada *fold*, se entrena y se valida el modelo para posteriormente realizar un testeo con datos que el sistema no había visto, con ello se consigue que los resultados sean fiables y extrapolables a pacientes nuevos. Durante el entrenamiento, se obtiene una gráfica donde se muestra el valor del coeficiente Dice a lo largo de las épocas (véase Figura 5.1).

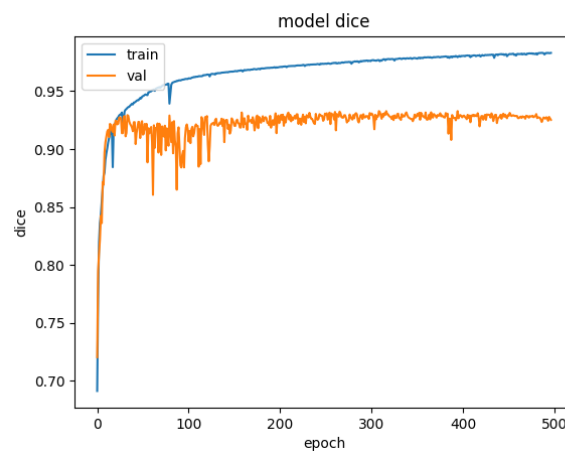


Figura 5.1: Gráfica representativa del valor del Dice a lo largo de las épocas de entrenamiento.

En esta sección, se exponen resultados tanto cualitativos como cuantitativos obtenidos comparando la máscara generada por el modelo durante la fase de predicción con el *ground truth*. A continuación, en la Figura 5.2 se muestran algunas imágenes de la segmentación manual y la automática (*output* del sistema) superpuestas sobre su correspondiente imagen RGB (*input* del sistema), con la finalidad de presentar ejemplos visuales de la precisión del modelo de *deep learning* construido.

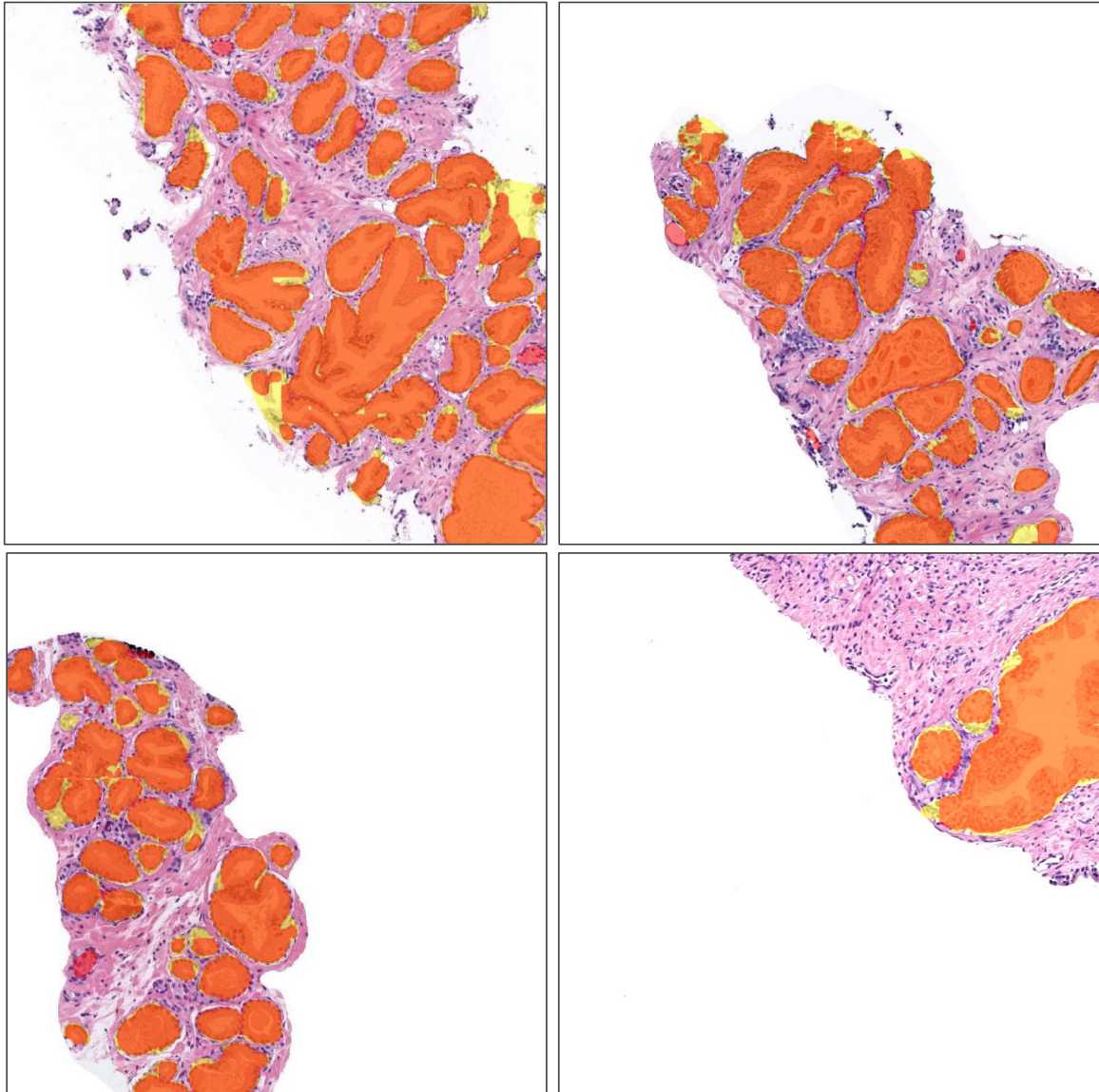


Figura 5.2: Ejemplos de imágenes con glándulas segmentadas manualmente en amarillo, segmentadas automáticamente en rojo y las áreas solapadas se ven en naranja.

En general, se pueden apreciar resultados cualitativos bastante buenos. No obstante, es posible observar en la Figura 5.3, como el modelo presenta ciertos problemas a la hora de distinguir entre artefactos y glándulas.

Por una parte, es considerablemente más preciso a la hora de segmentar glándulas redondas, cerradas y con límites bien definidos. Asimismo, ofrece mejores resultados cuando la glándula

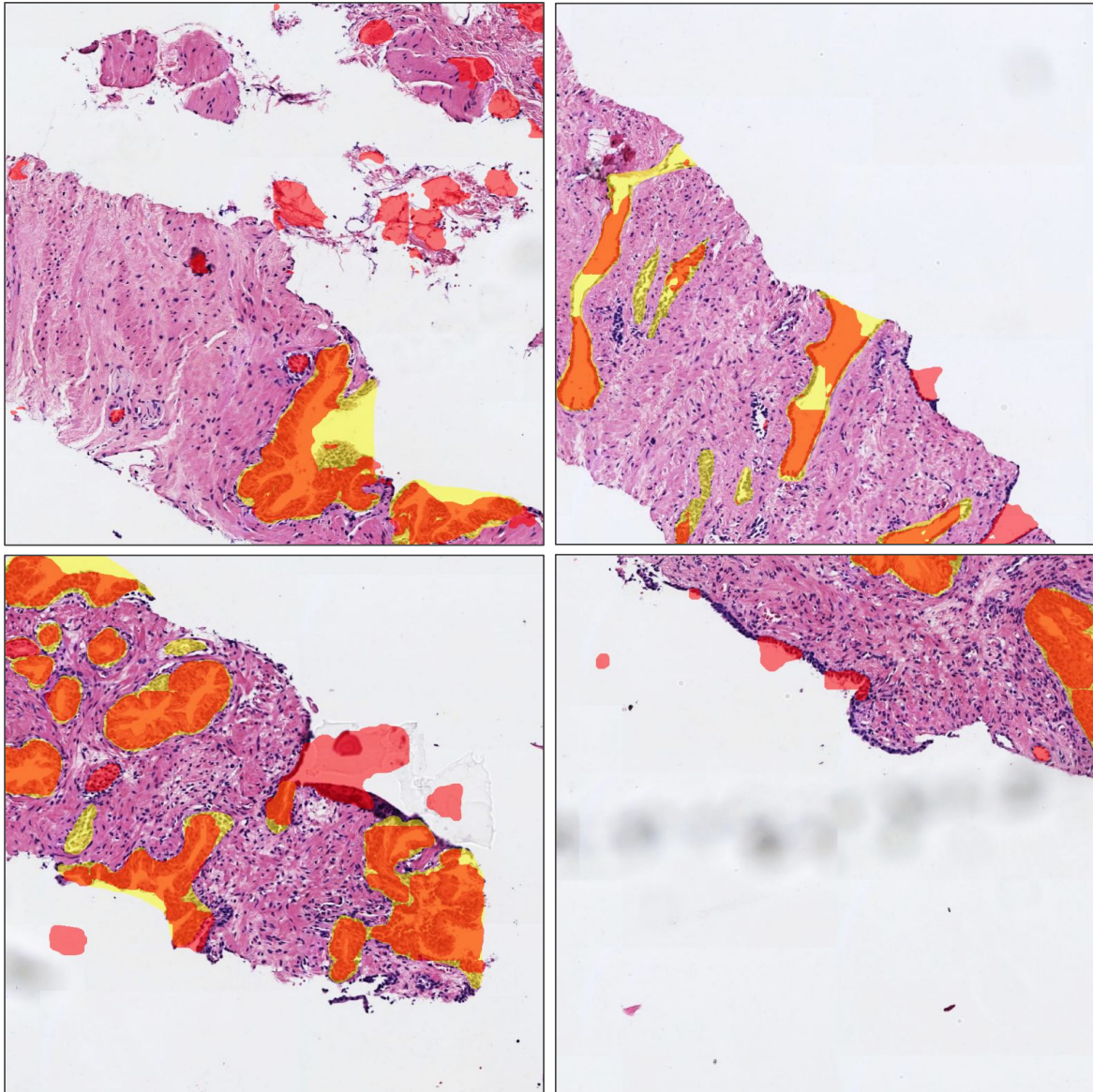


Figura 5.3: Ejemplos de imágenes con glándulas mal segmentadas.

posee una sola capa de núcleos. Por otra parte, no consigue segmentar correctamente aquellas glándulas que son más grandes e irregulares. Principalmente, tiene dificultad cuando las glándulas se encuentran abiertas o rotas, al igual que cuando se encuentran en el borde del parche.

A la hora de evaluar cuantitativamente el método propuesto en este proyecto, se han calculado los indicadores propuestos en el apartado de evaluación. Para ello, se ha obtenido una imagen binarizada a partir de los mapas de probabilidad de cada clase y se han aplicado dichas métricas tanto para la clase fondo como para la clase glándula de cada imagen y, a continuación, se ha calculado la media y la desviación típica del conjunto de resultados para cada indicador. En la memoria se da más importancia a los resultados de la clase glándula ya que es el objetivo final del TFM.

Indicador	Glándula ($\mu \pm \sigma$)	Fondo ($\mu \pm \sigma$)
JACCARD	0.6135 \pm 0.1773	0.9622 \pm 0.0684
DICE	0.7342 \pm 0.1632	0.9804 \pm 0.0175
BF score	0.7353 \pm 0.1904	0.8830 \pm 0.0803
Precisión BF	0.6552 \pm 0.2278	0.8358 \pm 0.1161
<i>Recall</i> BF	0.8601 \pm 0.0768	0.9459 \pm 0.0429

Tabla 5.1: Resultados medios de los modelos entrenados.

Como se puede apreciar en la Tabla 5.2, los valores de la desviación estándar son bastante elevados, puede deberse a que el modelo predice muy bien imágenes patológicas con glándulas más redondas y pequeñas mientras que imágenes con glándulas grandes que se encuentran cortadas confunden al modelo a la hora de realizar su predicción.

Una vez obtenidos los resultados del presente trabajo, se procede a la comparación con los diferentes autores del estado del arte. Normalmente, en la literatura no se ofrecen resultados cuantitativos de la etapa de segmentación automática ya que el objetivo final de los proyectos es la posterior clasificación en los diferentes grados. Debido a esto, para poder realizar una comparación de los resultados se ha tratado de buscar artículos cuya finalidad sea expresamente la segmentación.

Como se ha comentado en el estado del arte, en [33] se realiza la segmentación a partir de la unión de los núcleos, tomando el lumen de las glándulas como punto de partida, y obtienen un índice Jaccard de 0,66. Cabe destacar que seleccionan glándulas cuyos bordes se encuentran bien definidos. En [35] desarrollan un método complejo integrando información de bajo nivel, de alto nivel y de dominio, y obtienen en segmentación glandular un índice jaccard de 0,43. En [38], hacen uso del canal de luminancia del espacio de color CIE Lab obteniendo finalmente un índice jaccard de 0,31.

A continuación, se muestra una tabla comparativa del estado del arte.

	Nguyen, Sarkar y Jain [33]	Naik, Doyle, Feldman y col. [35]	Monaco, Tomaszewski, Feldman y col. [38]	García, Colomer, Naranjo y col. [36]	Método propuesto TFM
Tipo de muestras	A nivel de glándula	A nivel de glándula	A nivel de glándula	A nivel de glándula	A nivel de imagen
Número de muestras	309	309	309	500	981
Índice Jaccard	0.66	0.43	0.31	0.73	0.62

Tabla 5.2: Comparación resultados del estado del arte.

Cabe destacar que la comparación de los resultados con los propuestos en la literatura es considerablemente complicada debido a diversos factores. En primer lugar, cada estudio utiliza métodos

y abordajes muy diferentes. Otra variable es la de la base de datos, estas tienden a ser privadas lo que dificultan en gran medida la comparación objetiva de los resultados de cada método. Además, no solo se trabaja con muestras de diferentes pacientes, sino que las dimensiones empleadas para abordar el problema de segmentación también son diferentes, no existe una estandarización y, por tanto, para testear un método con muestras de otro proyecto sería necesario una adaptación. Finalmente, un problema tanto a la hora de comparar resultados como al evaluar el propio modelo es la subjetividad del *ground truth* al haber sido realizado por un único investigador.

Teniendo en cuenta lo expuesto, se puede considerar que los resultados de este TFM son muy prometedores ya que, además, el resto de artículos realizan una selección previa de las glándulas que van a segmentar en lugar de hacerlo a partir de la imagen de biopsia directamente. De forma que, llevan a cabo una evaluación de la segmentación solo a nivel de glándula, mientras que en este proyecto, la evaluación se realiza de la identificación y de la segmentación glandular a nivel de imagen. Por tanto, el método propuesto en este proyecto, a parte de ofrecer uno de los mejores resultados, no necesita de un preprocesado previo ni de una selección de candidatos a glándula previa para llevar a cabo la segmentación, lo cual añade valor al sistema diseñado.

Igualmente, destacar el hecho de que el valor empleado para comparar los resultados con el estado del arte es el índice Jaccard solo de la clase glándula pero si se utilizará el índice teniendo cuenta ambas clases, el resultado sería mayor y este proyecto destacaría considerablemente.

Por último, a fin de corroborar los resultados y teniendo en cuenta lo descrito anteriormente, se ha tratado de realizar la segmentación automática de la base de datos empleada en este TFM mediante el método del autor del artículo [52]. Esto se ha podido llevar a cabo gracias a que el código era público y, de esta forma, se ha podido aplicar a las mismas imágenes segmentadas en este estudio para obtener resultados más objetivos del método desarrollado. Los resultados obtenidos para las métricas Dice y Jaccard han sido $0,5132 \pm 0,2294$ y $0,3769 \pm 0,2085$ respectivamente. Al comparar directamente estos coeficientes con los obtenidos en el presente proyecto, se podría decir que el método propuesto es más preciso a la hora de segmentar glándulas.

Conclusiones y líneas futuras

6.1 Conclusiones

La principal motivación del proyecto es el desarrollo de un sistema completo de diagnóstico de cáncer de próstata mediante nuevas técnicas de análisis de imagen que permitan suplir los problemas que presentan las técnicas convencionales. En el presente TFM, se ha llevado a cabo una de las fases para conseguir este propósito, la segmentación glandular automática de tejido sano y de tejido patológico de grado 3.

En primer lugar, se ha realizado un estudio del estado del arte donde se ha podido observar que existe una amplia literatura en el diagnóstico de los diferentes grados de cáncer de próstata llevando a cabo técnicas muy diferentes, pero es muy escasa en el desarrollo de métodos de segmentación automática y, normalmente, no ofrecen resultados cuantitativos. Esto genera una mayor dificultad a la hora de determinar la bondad del sistema construido en este trabajo. Además, no se han podido encontrar artículos en los que diseñen arquitecturas *encoder-decoder* con el mismo objetivo que en este proyecto.

Previamente al diseño de la arquitectura de aprendizaje profundo se ha necesitado entender el fundamento teórico detrás de las redes neuronales con el fin de conocer qué tipo de red se emplea para este propósito y qué partes contiene la misma. Una vez adquiridos los conocimientos necesarios, se ha propuesto una arquitectura *encoder-decoder* ya que resulta la más adecuada para realizar segmentación automática debido a que permite obtener la salida del mismo tamaño que la entrada. A pesar de lo comentado anteriormente, si que se emplea para muchos otros problemas de segmentación en los que se quieren mantener las dimensiones y, por eso, se considera la manera óptima de encarar el problema presentado en este TFM.

El paso más importante para conseguir un sistema completo de segmentación automática es el entrenamiento de la red neuronal que, en este caso, es supervisado y, por ello, se necesita de una base de datos completa de imágenes etiquetadas que la arquitectura utiliza para aprender cómo segmentar glándulas independientemente del grado que estas tengan. Actualmente y por lo que respecta al conocimiento de la autora, no existen base de datos públicas etiquetadas, por tanto,

para poder llevar a cabo este proyecto ha sido necesaria una fase previa de creación del *ground truth* mediante una aplicación llamada *Image Labeler*.

Una vez preparada la base de datos de imágenes que se va a emplear, se ha implementado la técnica de *k-fold cross-validation*. Al realizar la partición de datos en los diferentes subconjuntos, se ha realizado una estrategia en la que todos los datos correspondientes a un mismo paciente se han establecido dentro de un mismo subconjunto. De esta manera, se ha implementado un *4-fold cross-validation* en el cual los datos se han particionado atendiendo tanto al historial clínico como al grado de cáncer resultando en un sistema totalmente balanceado por clases y número de muestras. Mediante esta metodología se consigue que un sistema robusto y fiable con resultados reproducibles.

Una vez definido el protocolo a seguir, se procede a realizar el entrenamiento de la arquitectura, para ello se ha escogido como función de pérdidas el coeficiente Dice y se ha minimizado dicha función mediante el optimizador NADAM. Además, se han definido una serie de hiperparámetros necesarios y se ha tratado de escoger sus valores óptimos para el problema actual.

Una vez entrenado cada modelo se ha testado con un conjunto de datos diferente y se ha obtenido el correspondiente mapa probabilístico de segmentación. Con el fin de obtener resultados tanto cualitativos como cuantitativos se han reconstruido las imágenes obtenidas como salida del modelo y se han comparado con su respectivo *ground truth*. Las métricas calculadas para conocer la precisión del método desarrollado han sido el coeficiente Dice, el índice Jaccard y el BFscore.

Una vez obtenidos los resultados cuantitativos, se ha escogido el mejor modelo de los implementados y se ha procedido a comparar con el estado del arte. Este paso resulta de gran dificultad ya que los protocolos implementados y las bases de datos empleadas son totalmente diferentes, pero teniendo esto en cuenta, se puede concluir que el enfoque de *deep learning* desarrollado en el presente proyecto es una buena línea de investigación.

En conclusión, se ha conseguido diseñar y desarrollar un sistema automático de segmentación glandular en imágenes histopatológicas de próstata con el objetivo de, posteriormente, servir como sistema de apoyo al diagnóstico para los especialistas en Anatomía Patológica facilitándoles el trabajo reduciendo los tiempos y el nivel de subjetividad.

6.2 Líneas futuras

Como se ha comentado anteriormente, este TFM aborda una de las etapas intermedias de un proyecto nacional, por tanto, la principal línea de futuro a seguir es el objetivo final del proyecto SICAP: diseño y desarrollo de un sistema completo de ayuda al diagnóstico capaz de diferenciar entre un tejido sano y uno patológico.

Resultaría interesante enlazar este proyecto con el TFG de la misma autora llamado “Desarrollo de un sistema de extracción avanzada de características en imagen histológica para la identificación automática del cáncer de próstata” que clasifica glándulas en tejido sano o patológicas de grado 3. De esta manera se habría conseguido un sistema completo en el que se segmentan automáticamente glándulas de tejido histopatológico mediante técnicas de *deep learning* para luego clasificarlas según su malignidad empleando técnicas de *machine learning*.

Límitándose exclusivamente a lo que respecta el presente trabajo de fin de máster, el principal objetivo sería mejorar los resultados logrados para ello, las líneas de investigación pueden incluir un postprocesado de eliminación de artefactos o un preprocesado en el que se obtengan diferentes canales de color que acentúen ciertas estructuras permitiendo que la arquitectura reconozca mejor las diferentes glándulas.

Por otra parte, al realizar un enfoque supervisado, la base de datos y el *ground truth* construido resultan de las partes más importantes del método diseñado. Por ello, una posible línea de investigación sería validar los datos mediante la realización de la segmentación manual por parte de otro especialista con el fin de disminuir la subjetividad a la que esta sujeta esta fase del proyecto. Además, sería interesante conseguir un mayor número de imágenes histopatológicas de próstata ya que en el entrenamiento de redes neuronales es muy importante contar con una base de datos muy amplia.

Finalmente, sería interesante el hecho de utilizar otro tipo de arquitectura de *deep learning* para conseguir una mejor segmentación de las glándulas, realizar un enfoque diferente en el balanceo de clases o utilizar otros métodos para optimizar los hiperparámetros empleados.

Parte II

Presupuesto

Capítulo 7

Presupuesto

El objetivo del presente capítulo es presentar una valoración económica del proyecto realizado. Se detallan las diferentes secciones que conforman el presupuesto del TFM basado en el desarrollo de un sistema de segmentación automática de glándulas prostáticas. Además, se comentan algunas consideraciones que se han tenido en cuenta.

7.1 Presupuestos parciales

El informe se desglosa en tres cuadros de precios: coste de mano de obra, coste de maquinaria y coste de materiales.

7.1.1 Coste mano de obra

A continuación, se describen los recursos humanos necesarios para el desarrollo del presente proyecto. Se realiza una estimación de los costes en función del tiempo dedicado al trabajo. Concretamente, se considera la contribución de: D^a. Valery Naranjo Ornedo (como directora del proyecto), D. José Gabriel García Pardo (como cotutor del proyecto) y D^a. Elena Payá Bosch (como estudiante del Máster de Ingeniería Biomédica y autora del proyecto).

Denominación	Uds.	Cantidad	Precio Unitario (€)	Total (€)
Tutora (Catedrática)	h	16	42,00	672,00
Cotutor (Doctorando)	h	36	17,20	619,20
Autora (Estudiante MIB)	h	600	13,80	8.280,00
			Total	9.571,20

Tabla 7.1: Cuadro de precios de la mano de obra.

7.1.2 Coste maquinaria

En esta subsección se detalla el cuadro de precios de los recursos correspondientes al *hardware* y al *software* que han sido necesarios para el desarrollo del presente TFM.

Cabe destacar que para la realización del proyecto, las herramientas *software* empleadas han sido MATLAB[®], concretamente, *Image Processing Toolbox*, y PyCharm 2019.1.1 como IDE de programación. Además, se ha empleado la librería *NVIDIA Cuda[®] Deep Neural Network* y la librería *OPENCV*.

Por otra parte, para el desarrollo del trabajo escrito, se ha empleado la aplicación conocida como *Overleaf* (una herramienta de escritura, edición y publicación en línea de LaTeX).

Denominación	Uds.	Cantidad	Precio Unitario (€)	Periodo de amortización (años)	Intervalo amortizado (meses)	Total (€)
Overleaf	u	1	0,00	1	8	0,00
Licencia MATLAB [®]	u	1	800,00	1	8	533,33
Image Processing Toolbox	u	1	400,00	1	8	266,66
Python 3.5	u	1	0	1	8	0
Librerías cuDNN y OPENCV	u	1	0	1	8	0
Total						799,99

Tabla 7.2: Cuadro de precios de las herramientas *software*.

Asimismo, se tienen en cuenta las herramientas *hardware* que se han empleado específicamente para la elaboración del TFM. El proyecto se ha desarrollado haciendo uso de un ordenador personal, un servidor para el almacenamiento de los datos y una GPU del grupo de investigación CVBLab que permitiera el entrenamiento de las redes neuronales construidas.

Diseño y desarrollo de un sistema automático de segmentación de glándulas histológicas para identificar el cáncer de próstata en una etapa inicial.

Denominación	Uds.	Cantidad	Precio Unitario (€)	Periodo de amortización (años)	Intervalo amortizado (meses)	Total (€)
HP Envy 13-ah0006ns	u	1	1.199,00	4	8	199,83
Procesador Intel i7 @4.20GHz	u	1	344,00	4	4	28,67
Disco SSD 250GB	u	1	77,00	4	4	6,41
NVIDIA Titan V	u	1	3300,00	4	2	137,50
Servidor Synology CVBLab	u	1	00,00	4	6	00,00
Total						372,41

Tabla 7.3: Cuadro de precios de las herramientas *hardware*.

7.1.3 Coste materiales

Finalmente, para llevar a cabo el proyecto se ha requerido de la realización de pruebas de biopsia, a partir de las cuales, se han obtenido las muestras correspondientes a las imágenes utilizadas durante el proyecto. El número de biopsias corresponde con el número de pacientes. Además, hay que tener en cuenta el coste de la adquisición de la muestra del corte histológico, la *slide*.

Denominación	Uds.	Cantidad	Precio Unitario (€)	Total (€)
Biopsia	u	25	600,00	15.000,00
Muestras	u	35	10,00	350,00
Total				15.350,00

Tabla 7.4: Cuadro de precios de los materiales.

7.2 Presupuesto total

Para el cálculo del presupuesto total del proyecto, es necesario tener en cuenta los cuadros de precios de los presupuestos parciales definidos anteriormente. Además, se añadirán el porcentaje de gastos generales (13 %) y el asociado al beneficio industrial (6 %). A continuación, se añadirá al precio total bruto el impuesto del IVA (21 %), obteniendo como resultado el presupuesto total que supondría la realización del presente TFM.

CAPÍTULOS	IMPORTE (€)
1. Coste de la mano de obra	9.571,20
2. Coste de las herramientas	1.172,40
3. Coste de los materiales	15.350,00
PRESUPUESTO DE EJECUCIÓN DE MATERIAL	26.093,60
13 % de gastos generales	3.392,17
6 % de beneficio industrial	1.565,62
PRESUPUESTO DE EJECUCIÓN POR CONTRATA	31.051,39
21 % de IVA	6.520,79
PRESUPUESTO TOTAL	37.572,18

Tabla 7.5: Presupuesto total.

Bibliografía

- [1] National Cancer Institute, *Cáncer*, <https://www.nih.gov/>, 2018 (vid. págs. 3, 6).
- [2] Organización Mundial de la Salud, *Cáncer*, <http://www.who.int/es/news-room/fact-sheets/detail/cancer>, 2018 (vid. pág. 3).
- [3] World Health Organization, *Global Cancer Observatory (GCO)*, <https://gco.iarc.fr/> (vid. pág. 4).
- [4] SEOM, “Las Cifras del Cáncer en España”, Sociedad Española de Oncología Médica (SEOM), inf. téc., 2019 (vid. págs. 4, 5).
- [5] Asociación Española Contra el Cáncer, *¿QUÉ ES EL CÁNCER DE PRÓSTATA?*, <https://www.aecc.es/es/todo-sobre-cancer/tipos-cancer/cancer-prostata/que-es-cancer-prostata>, 2018 (vid. págs. 5, 6).
- [6] Galenored International, *Tacto rectal*, <https://galenored.com/tacto-rectal/>, 2017 (vid. pág. 7).
- [7] LABORATORIOS CHONG, *Análisis de sangre del Antígeno Prostático Específico (PSA)*, <http://laboratorioschong.com.mx/analisis-de-sangre-del-antigeno-prostatico-especifico-psa/>, 2018 (vid. pág. 7).
- [8] MAYO CLINIC, *Biopsia de próstata*, <https://www.mayoclinic.org/es-es/tests-procedures/prostate-biopsy/about/pac-20384734> (vid. pág. 7).
- [9] Wikipedia, *Tinción hematoxilina-eosina* — *Wikipedia, La enciclopedia libre*, [Internet; descargado 28-junio-2018], 2018 (vid. pág. 9).
- [10] P. C. Foundation, *Gleason Score and ISUP Grade Group*, <https://www.pcf.org/about-prostate-cancer/diagnosis-staging-prostate-cancer/gleason-score-isup-grade/> (vid. pág. 9).

- [11] J. I. Epstein, “A new contemporary prostate cancer grading system.”, en *Annales de pathologie*, vol. 35, 2015, pág. 474 (vid. pág. 10).
- [12] PCEC, “Gleason Score, Prostate Cancer Grading & Prognostic Scoring”, Prostate Conditions Education Council, inf. téc. (vid. pág. 10).
- [13] M. A. Boden, *Inteligencia artificial*. Turner, 2017 (vid. pág. 15).
- [14] D. M. J. Garbade, “Clearing the Confusion: AI vs Machine Learning vs Deep Learning Differences”, *Towards Data Science, September*, vol. 14, 2018 (vid. pág. 16).
- [15] I. BERNARD WIDROW FELLOW y M. A. LEHR, “30 Years of Adaptive Neural Networks: Perceptron, Madaline, and Backpropagation”, *PROCEEDINGS OF THE IEEE*, vol. 78, sep. de 1990. DOI: 0018-9219/90/0900-1415 (vid. pág. 16).
- [16] M. Z. Alom, T. M. Taha, C. Yakopcic, S. Westberg, P. Sidike, M. S. Nasrin, B. C. Van Esesn, A. A. S. Awwal y V. K. Asari, “The history began from alexnet: A comprehensive survey on deep learning approaches”, *arXiv preprint arXiv:1803.01164*, 2018 (vid. pág. 16).
- [17] J. Long, E. Shelhamer y T. Darrell, “Fully convolutional networks for semantic segmentation”, en *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, págs. 3431-3440 (vid. pág. 16).
- [18] A. Cuesta, *Apuntes de Reconocimiento de Patrones*. Universidad Rey Juan Carlos. 2018/19 (vid. págs. 17, 22, 23, 28, 29).
- [19] . MissingLink.ai, *Complete Guide to Artificial Neural Network Concepts and Models*. [online] (vid. págs. 18, 43).
- [20] D. E. Rumelhart, G. E. Hinton, R. J. Williams y col., “Learning representations by back-propagating errors”, *Cognitive modeling*, vol. 5, n.º 3, pág. 1, 1988 (vid. pág. 19).
- [21] G. García, A. Colomer y V. Naranjo, “First-Stage Prostate Cancer Identification on Histopathological Images: Hand-Driven versus Automatic Learning”, *Entropy*, vol. 21, n.º 4, pág. 356, 2019 (vid. pág. 19).
- [22] J. I. Bagnatot, “¿Cómo funcionan las Convolutional Neural Networks? Visión por Ordenador”, (vid. págs. 22, 26).
- [23] Diego Calvo, *Función de activación - Redes neuronales*, www.diegocalvo.es/funcion-de-activacion-redes-neuronales/, 7 diciembre 2018 (vid. pág. 25).
- [24] V. Badrinarayanan, A. Kendall y R. Cipolla, “SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, págs. 2481-2495, 2015 (vid. págs. 30, 33, 43).

- [25] J. Yang, B. Price, S. Cohen, H. Lee y M.-H. Yang, “Object contour detection with a fully convolutional encoder-decoder network”, en *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, págs. 193-202 (vid. pág. 30).
- [26] G. García, A. Colomer y V. Naranjo, “First-Stage Prostate Cancer Identification on Histopathological Images: Hand-Driven versus Automatic Learning”, *Entropy*, vol. 21, pág. 356, abr. de 2019. DOI: 10.3390/e21040356 (vid. págs. 31, 32).
- [27] J. Diamond, N. H. Anderson, P. H. Bartels, R. Montironi y P. W. Hamilton, “The use of morphological characteristics and texture analysis in the identification of tissue composition in prostatic neoplasia.”, *Human pathology*, vol. 35 9, págs. 1121-31, 2004 (vid. pág. 31).
- [28] A. Esteban, A. Colomer, V. Naranjo y M A. Sales, “Granulometry-Based Descriptor for Pathological Tissue Discrimination in Histopathological Images”, oct. de 2018, págs. 1413-1417. DOI: 10.1109/ICIP.2018.8451805 (vid. pág. 31).
- [29] S. Doyle, M. Hwang, K. Shah, A. Madabhushi, M. Feldman y J. Tomaszewski, “AUTOMATED GRADING OF PROSTATE CANCER USING ARCHITECTURAL AND TEXTURAL IMAGE FEATURES”, en *2007 4th IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, 2007, págs. 1284-1287. DOI: 10.1109/ISBI.2007.357094 (vid. pág. 31).
- [30] M. M.N.R.T.H.C.F.N.W.P.J.W.J.H.R.M. C. Eirini Arvaniti Kim S. Fricker, “Automated Gleason grading of prostate cancer tissue microarrays via deep learning”, en *Scientific Reports volume 8, Article number: 12054 (2018)*, 2018, pág. 12 054. DOI: 10.1038/s41598-018-30535-1 (vid. pág. 31).
- [31] R. Farjam, H. Soltanian-Zadeh, K. Jafari-Khouzani y R. Zoroofi, “An Image Analysis Approach for Automatic Malignancy Determination of Prostate Pathological Images”, *Cytometry. Part B, Clinical cytometry*, vol. 72, págs. 227-40, jul. de 2007. DOI: 10.1002/cyto.b.20162 (vid. pág. 31).
- [32] J. T. Kwak y S. Hewitt, “Multiview boosting digital pathology analysis of prostate cancer”, *Computer Methods and Programs in Biomedicine*, vol. 142, págs. 91-99, abr. de 2017. DOI: 10.1016/j.cmpb.2017.02.023 (vid. pág. 32).
- [33] K. Nguyen, A. Sarkar y A. K. Jain, “Structure and Context in Prostatic Gland Segmentation and Classification”, en *Proceedings of the 15th International Conference on Medical Image Computing and Computer-Assisted Intervention - Volume Part I*, ép. MICCAI'12, Nice, France: Springer-Verlag, 2012, págs. 115-123, ISBN: 978-3-642-33414-6. DOI: 10.1007/978-3-642-33415-3_15 (vid. págs. 32, 54).
- [34] S. Naik, S. Doyle, S. Agner, A. Madabhushi, M. Feldman y J. Tomaszewski, “Automated gland and nuclei segmentation for grading of prostate and breast cancer histopathology”,

- en *2008 5th IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, 2008, págs. 284-287. DOI: 10.1109/ISBI.2008.4540988 (vid. pág. 32).
- [35] S. Naik, S. Doyle, M. Feldman, J. Tomaszewski y A. Madabhushi, “Gland segmentation and computerized gleason grading of prostate histology by integrating low-, high-level and domain specific information”, en *MIAAB workshop*, Citeseer, 2007, págs. 1-8 (vid. págs. 32, 54).
- [36] J. G. García, A. Colomer, V. Naranjo, F. Peñaranda y M. Á. Sales, “Identification of Individual Glandular Regions Using LCWT and Machine Learning Techniques”, en *International Conference on Intelligent Data Engineering and Automated Learning*, Springer, 2018, págs. 642-650 (vid. págs. 32, 54).
- [37] K. Nguyen, A. K. Jain y R. L. Allen, “Automated Gland Segmentation and Classification for Gleason Grading of Prostate Tissue Images”, en *2010 20th International Conference on Pattern Recognition*, 2010, págs. 1497-1500. DOI: 10.1109/ICPR.2010.370 (vid. pág. 32).
- [38] J. Monaco, J. Tomaszewski, M. Feldman, I. Hagemann, M. Moradi, P. Mousavi, A. Boag, C. Davidson, P. Abolmaesumi y A. Madabhushi, “High-Throughput Detection of Prostate Cancer in Histological Sections Using Probabilistic Pairwise Markov Models”, *Medical image analysis*, vol. 14, págs. 617-29, ago. de 2010. DOI: 10.1016/j.media.2010.04.007 (vid. págs. 32, 54).
- [39] J. Xu, X. Luo, G. Wang, H. Gilmore y A. Madabhushi, “A deep convolutional neural network for segmenting and classifying epithelial and stromal regions in histopathological images”, *Neurocomputing*, vol. 191, págs. 214-223, 2016 (vid. pág. 32).
- [40] W. Bulten, P. Bándi, J. Hoven, R. van de Loo, J. Lotz, N. Weiss, J. van der Laak, B. van Ginneken, C. Hulsbergen-van de Kaa y G. Litjens, “Epithelium segmentation using deep learning in H&E-stained prostate specimens with immunohistochemistry as reference standard”, *Scientific reports*, vol. 9, n.º 1, pág. 864, 2019 (vid. pág. 32).
- [41] C. Gunduz-Demir, M. Kandemir, A. B. Tosun y C. Sokmensuer, “Automatic segmentation of colon glands using object-graphs”, *Medical image analysis*, vol. 14, n.º 1, págs. 1-12, 2010 (vid. pág. 32).
- [42] A. BenTaieb y G. Hamarneh, “Topology aware fully convolutional networks for histology gland segmentation”, en *International Conference on Medical Image Computing and Computer-Assisted Intervention*, Springer, 2016, págs. 460-468 (vid. pág. 32).
- [43] F. Milletari, N. Navab y S.-A. Ahmadi, “V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation”, oct. de 2016, págs. 565-571. DOI: 10.1109/3DV.2016.79 (vid. pág. 33).

- [44] K. Simonyan y A. Zisserman, “Very deep convolutional networks for large-scale image recognition”, *arXiv preprint arXiv:1409.1556*, 2014 (vid. pág. 33).
- [45] J. G. G. Pardo, *Diseño y desarrollo de un sistema automático de clasificación de estructuras glandulares en imágenes histológicas de próstata*, 2017-18 (vid. pág. 37).
- [46] *MicroDraw*, <http://microdraw.pasteur.fr/> (vid. pág. 37).
- [47] The MathWorks, Inc., *MATLAB*, <https://es.mathworks.com/products/matlab.html> (vid. pág. 38).
- [48] M. Drozdal, G. Chartrand, E. Vorontsov, L. Di-Jorio, A. Tang, A. Romero, Y. Bengio, C. Pal y S. Kadoury, “Learning Normalized Inputs for Iterative Estimation in Medical Image Segmentation”, *CoRR*, vol. abs/1702.05174, 2017. arXiv: 1702.05174 (vid. pág. 43).
- [49] G. A. García Ferrando, *Deep Learning en segmentación de imagen médica*, 2017 (vid. págs. 43, 46).
- [50] J. Torres, “Deep Learning. Introducción práctica con Keras (PRIMERA PARTE)”, Universitat Politècnica de Catalunya - UPC Barcelona Tech, 2018, ISBN: 978-0-244-07895-9 (vid. págs. 46, 47).
- [51] The MathWorks, Inc., *BFscore*, <https://es.mathworks.com/help/images/ref/bfscore.html> (vid. pág. 49).
- [52] K. Nguyen, B. Sabata y A. K. Jain, “Prostate cancer grading: Gland segmentation and structural features”, *Pattern Recognition Letters*, vol. 33, n.º 7, págs. 951 -961, 2012, Special Issue on Awards from ICPR 2010, ISSN: 0167-8655. DOI: <https://doi.org/10.1016/j.patrec.2011.10.001> (vid. pág. 55).

