



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



ESCUELA TÉCNICA
SUPERIOR INGENIERÍA
INDUSTRIAL VALENCIA

TRABAJO FIN DE MASTER EN INGENIERÍA BIOMÉDICA

DISEÑO Y DESARROLLO DE UN SISTEMA AUTOMÁTICO PARA LA GRADACIÓN DEL CÁNCER DE PRÓSTATA APLICANDO TÉCNICAS MULTI-RESOLUCIÓN CON ALGORITMOS DE DEEP LEARNING EN WHOLE-SLIDE IMAGES

AUTOR: ELENA NIETO GONZÁLEZ

TUTOR: VALERY NARANJO ORNEDO

COTUTORES: JULIO JOSÉ SILVA RODRÍGUEZ
JOSÉ GABRIEL GARCÍA PARDO

Curso Académico: 2019-20

Agradecimientos

*“Gracias a Julio y Gabri por ayudarme y enseñarme tanto en la realización de este proyecto,
A Valery por darme la oportunidad de trabajar en él
Y a mi familia y amigos por apoyarme”*

Resumen

El cáncer de próstata es uno de los tipos de cánceres más frecuentemente diagnosticados en todo el mundo. Actualmente, el diagnóstico definitivo de esta enfermedad viene precedido del examen visual de las muestras de tejido biopsiadas por parte del patólogo, quien asigna una puntuación de las estructuras histológicas acorde con la escala Gleason. Esta inspección manual consume grandes cantidades de tiempo al médico y presenta cierto grado de subjetividad en cada patólogo.

En los últimos años, ha surgido la necesidad de desarrollar sistemas de ayuda al diagnóstico para reducir la gran carga de trabajo y el nivel de variabilidad entre los diagnósticos de diferentes especialistas, basados en sistemas automáticos de visión por computador. Cabe destacar que la mayoría de estos sistemas utilizan una resolución fija de la imagen. No obstante, los patólogos analizan la muestra de tejido bajo el microscopio haciendo uso de diferentes aumentos, ya que, para determinar un diagnóstico, necesitan no solo la información de la región de interés a nivel local, sino también la información del contexto en el que se encuentra dicha región. Por este motivo, el objetivo de este TFM versa sobre el desarrollo de enfoques multi-resolución aplicando algoritmos de *deep learning* para la detección de patrones cancerosos en WSIs, según la escala Gleason, teniendo en cuenta la información de las imágenes a distintas resoluciones.

En el presente TFM, se propone el desarrollo de una red neuronal convolucional que procese conjuntamente las muestras de las biopsias a distintas resoluciones, y combine sus características para realizar una clasificación más precisa. Esta red, ha sido comparada con modelos a resolución fija. En la etapa experimental se demuestra que el método multi-resolución mejora los resultados, alcanzando un kappa cuadrático de Cohen de 0.5430.

Palabras clave: cáncer de próstata, *Deep learning*, red neuronal convolucional, imágenes histológicas, multi-resolución

Resum

El càncer de pròstata és un dels tipus de càncers més sovint diagnosticats en tot el món. Actualment, el diagnòstic definitiu d'aquesta malaltia ve precedit de l'examen visual de les mostres de teixit biopsiades per part del patòleg, qui assigna una puntuació de les estructures histològiques d'acord amb l'escala Gleason. Esta inspecció manual consumix grans quantitats de temps al metge i presenta un cert grau de subjectivitat en cada patòleg.

En els últims anys, ha sorgit la necessitat de desenvolupar sistemes d'ajuda al diagnòstic per a reduir la gran càrrega de treball i el nivell de variabilitat entre els diagnòstics dels diferents especialistes, basats en sistemes automàtics de visió per computador. Cal destacar que la majoria d'aquests sistemes utilitzen una resolució fixa de la imatge. No obstant això, els patòlegs analitzen la mostra de teixit davall el microscopi fent ús de diferents augments, ja que, per a determinar un diagnòstic, necessiten no sols la informació de la regió d'interès a nivell local, sinó també la informació del context en què es troba la dita regió. Per este motiu, l'objectiu d'aquest TFM versa sobre el desenvolupament d'enfocaments multi-resolució aplicant algoritmes de *deep learning* per a la detecció de patrons cancerosos en WSIs, segons l'escala Gleason, tenint en compte la informació de les imatges a distintes resolucions.

En el present TFM, es proposa el desenrotllament d'una xarxa neuronal convolucional que processa conjuntament les mostres de les biòpsies a distintes resolucions, i combina les seues característiques per a realitzar una classificació més precisa. Esta xarxa, ha sigut comparada amb models a resolució fixa. En l'etapa experimental es demostra que el mètode multi-resolució millora els resultats, aconseguint un kappa quadràtic de Cohen de 0.5430.

Paraules clau: càncer de pròstata, *Deep learning*, xarxa neuronal convolucional, imatges histològiques, multi-resolució

Abstract

Prostate cancer is one of the most frequent diagnosed types of cancer in the world. Currently, the definitive diagnosis of this disease is precedent by the visual examination of the biopsied tissue samples by the pathologist, who assigns a score of the histological structures according to the Gleason score. This manual inspection consumes large amounts of time for the doctor and also presents a certain degree of subjectivity in each pathologist.

In recent years, the need has arisen to develop diagnostic support systems to reduce the large workload and the level of variability between the diagnoses of different specialists, based on automatic computer vision systems. It should be noted that most of these systems use a fixed image resolution. However, pathologists analyse the tissue sample under the microscope using different magnifications, since, in order to determine a diagnosis, they need not only information about the region of interest at the local level, but also information about the context in which that region is located. For this reason, the objective of this Master Thesis is to develop multi-resolution approaches by applying deep learning algorithms for the detection of cancer patterns in WSIs, according to the Gleason scale, taking into account the information from the images at different resolutions.

In the present Master Thesis, it is proposed the development of a convolutional neuronal network that processes together the biopsy samples at different resolutions, and combines its characteristics to make a more precise classification. This network has been compared with fixed resolution models. In the experimental stage, it is demonstrated that the multi-resolution method improves the results, reaching a Cohen's square Kappa of 0.5430.

Keywords: prostate cancer, deep learning, convolutional neuronal network, histological images, multi-resolution

Índice General

I. Memoria

II. Presupuesto

Parte I

Memoria

Índice de la memoria

Introducción.....	3
1.1. Cáncer	3
1.1.1. Cáncer de próstata	5
1.1.2. Escala de Gleason.....	8
1.2. Imágenes histológicas	10
1.2.1. Preparación muestras histológicas en Whole Slide Images (WSI)	10
1.2.2. Tinción Hematoxilina & Eosina.....	11
1.3. Análisis de las muestras	13
1.4. Marco del Proyecto	14
1.5. Estado del arte.....	15
Objetivos.....	18
Material y Metodología	20
3.1. Material.....	20
3.1.1. Base de datos	20
3.1.2. Software empleado	22
3.1.3. Hardware empleado.....	22
3.2. Metodología	24
3.2.1. Preparación de la base de datos	24
3.2.2. Creación Ground Truth.....	26
3.2.3. Partición de los datos.....	27
3.2.4. Generador de datos	28
3.2.5. Entrenamiento.....	31
3.2.6. Evaluación	40

Resultados y discusión	42
Conclusión y líneas futuras.....	49
5.1. Conclusión	49
5.2. Líneas futuras.....	50
Bibliografía.....	51

Capítulo 1

Introducción

En este capítulo se va a explicar la problemática del cáncer de próstata por su incidencia y método de diagnóstico. El cáncer de próstata es uno de los más comunes entre los hombres hoy en día. A pesar de que a lo largo de los años su pronóstico ha mejorado, es imprescindible detectarlo a tiempo.

El tratamiento se realiza en función del avance de la enfermedad y por ello es importante la gradación del tumor, en el caso del cáncer de próstata se gradada en función de una escala llamada Escala de Gleason. Este trabajo es realizado normalmente por patólogos, es un trabajo muy tedioso y subjetivo lo que genera una alta variabilidad. Por ello, surge la necesidad de desarrollar sistemas de ayuda a la decisión que podrían reducir los tiempos de diagnóstico con una mayor fiabilidad.

El uso del *Deep Learning* en estos sistemas ha aumentado los últimos años, y particularmente las redes neuronales convolucionales (CNN) han demostrado tener una alta precisión para la segmentación y gradación del cáncer de próstata como se puede observar en la literatura científica observada los últimos años.

1.1. Cáncer

El cáncer es el nombre que recibe un conjunto de enfermedades relacionadas en las que se observa un proceso descontrolado en la división de células del cuerpo. Cuando un número de células presentan mutaciones u otras alteraciones, el propio cuerpo activa una

serie de mecanismos con la finalidad de inducir las a la muerte celular o apoptosis. En cambio, las células cancerosas desarrollan mutaciones que no son reparadas y pierden capacidad para morir (Puente y de Velasco, 2019).

El cáncer es una de las principales causas de morbi-mortalidad en el mundo, alrededor de 18,1 millones de casos fueron diagnosticados en 2018, últimos datos disponibles. Las estimaciones poblacionales indican que esta cifra aumentará a 29,5 millones en 2040, un aumento del 63,1%. Se estima que en 2020 sean diagnosticados 277.394 casos en España, de ellos, los más frecuentes serán colon y recto (44.231 casos), próstata (35.953), pulmón (29.638) y vejiga urinaria (22.350). A mucha distancia, se encuentran otros tipos de cáncer, como se puede observar en Fig. 1, estos cinco tipos abarcan más de la mitad de los casos totales.

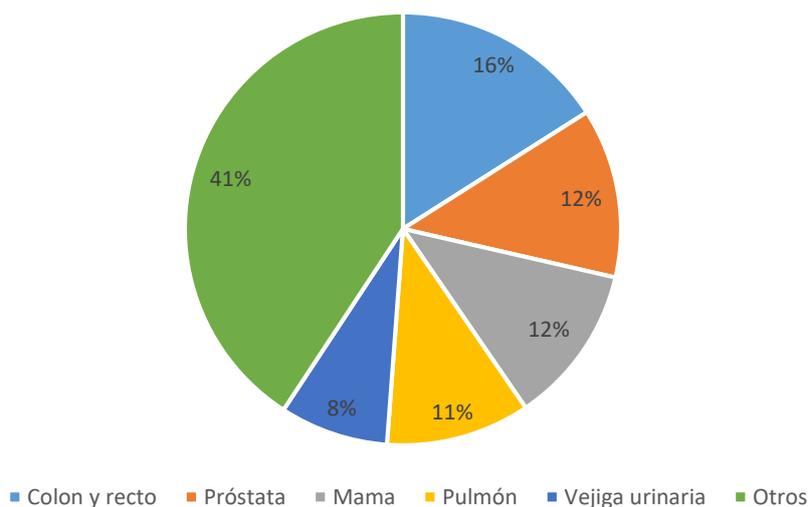


Fig. 1. Estimación del número de nuevos casos de cáncer en España para el año 2020 según el tipo tumoral. ("Cifras del cáncer 2020", 2020)

La prevalencia es la proporción de la población que está sujeta a la enfermedad en un periodo o momento determinado, siendo estas diagnosticadas previamente y sin que el paciente haya fallecido. Por lo tanto, los tumores con mayor prevalencia serán aquellos que tienen una mayor tasa de supervivencia los cuales son de próstata y de mama en hombres y mujeres respectivamente, con 25,8% de los casos de cáncer en varones y una prevalencia en 5 años de 106.941 casos. ("Cifras del cáncer 2020", 2020)

1.1.1. Cáncer de próstata

La próstata junto con las vesículas seminales y las glándulas bulbouretrales conforman las glándulas accesorias del aparato reproductor masculino. El líquido producido por la glándula prostática representa aproximadamente el 30% del semen.

Su función principal es la secreción de un líquido el cual contiene varios componentes, entre ellos ácido cítrico que sirve de nutriente para los espermatozoides, fosfatasa alcalina y fibrinolisisina que sirve para licuar el semen. Además, ayuda a expulsar el semen a través de la uretra, esta zona de la uretra atravesada por la próstata es la uretra prostática.

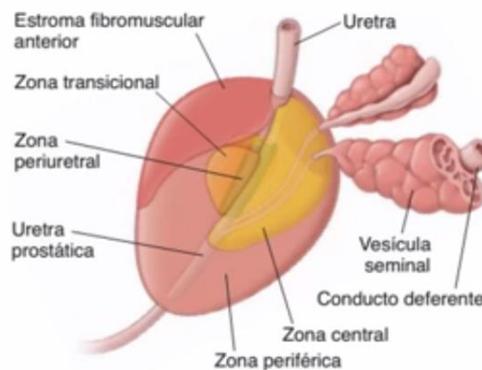


Fig. 2. Fisiología de la próstata extraído de ("*Histología de la próstata*", 2020)

Tal y como se puede observar en Fig. 3 la próstata está compuesta por varias zonas las cuales tienen una fisiología e histología diferentes:

- Zona de periuretral: compuesta por glándulas mucosas que drenan directamente en la uretra y la rodea. Esta zona puede sufrir una proliferación anormal de las células comprimiendo parcial o totalmente la uretra, lo que se conoce como BPH (hiperplasia prostática benigna).
- Zona central: compuesta por glándulas submucosas, drenan en los senos prostáticos y rodea los conductos eyaculadores. Esta zona es resistente a la inflamación y a los carcinomas.
- Zona periférica o principal: compuesta por glándulas principales que drenan en los senos prostáticos y rodea toda la próstata. Es la zona más susceptible a la inflamación y es donde suelen generarse los carcinomas prostáticos.

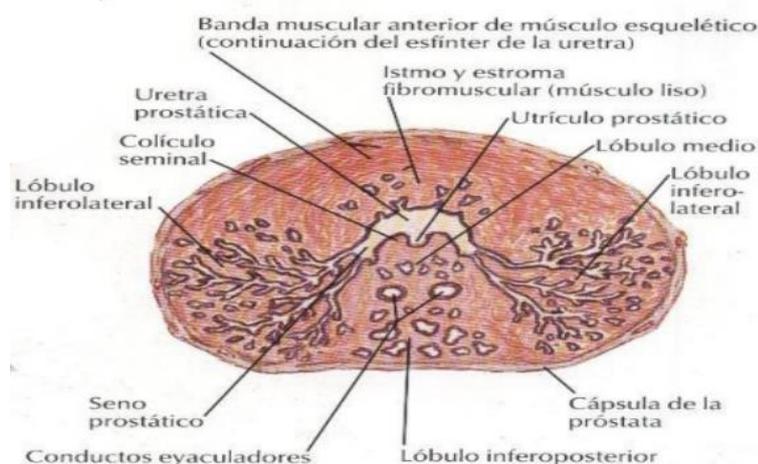


Fig. 3. Corte transversal de la próstata extraído de ("Histología de la próstata", 2020)

El cáncer de próstata es el segundo cáncer más común en hombres después del cáncer de pulmón y la quinta causa principal de muerte en Europa por cáncer. La incidencia en España según un estudio observacional realizado en 25 hospitales españoles es de 70,75 casos por 100.000 habitantes (Campá, Mar-Barrutia, Extramiana, Arróspide, y Mar, 2016).

En el mundo se registró en 2018 casi 1,3 millones de casos, particularmente en España se registraron 31.728 nuevos casos y fallecieron por esta causa 6.061 hombres. ("Cáncer de próstata", 2020). La Organización Mundial de la Salud prevé que el número de casos por año aumente a 1,8 millones en esta década.

El cáncer de próstata se produce en un 99% de los casos en células glandulares, las células que son las encargadas de producir el líquido prostático, que forma parte del semen. Puede crecer de tres maneras:

- *Local*: Se produce por crecimiento tumoral e invasión de la cápsula prostática. Si avanza, el tumor puede romper la misma e invadir los tejidos y órganos colindantes a la próstata.
- *Diseminación linfática*: Cuanto mayor es el tamaño del tumor primitivo, mayor es la probabilidad de afectación ganglionar.
- *Diseminación hematológica*: Se realiza a través de los vasos sanguíneos y viaja hacia el hueso.

A uno de cada nueve hombres le será diagnosticado cáncer de próstata, y seis de cada diez de este grupo serán mayores de 65 años. Raramente se diagnostica el cáncer de próstata antes de los 40 años, la edad media del diagnóstico es a los 66 años. ("Estadísticas importantes sobre el cáncer de próstata", 2020) También, se ha observado una fuerte predisposición genética a padecer cáncer de próstata basada en grupo racial (raza afroamericana) y por el historial familiar (Ortega Alonzo, González Páramo y Galván Losada, 2019).

La detección temprana del cáncer de próstata es algo muy importante para asegurar un mejor pronóstico de la enfermedad y calidad de vida de los pacientes. Algunos de los síntomas del cáncer de próstata son ("Cáncer de próstata - Síntomas y signos", 2020):

- Micción frecuente
- Flujo miccional débil o interrumpido, o necesidad de hacer fuerza para vaciar la vejiga, esto puede llegar a generar ardor o dolor
- Sangre en la orina o en el líquido seminal
- Aparición de disfunción eréctil
- Molestias o dolor al estar sentado, causadas por un aumento del tamaño de la próstata

Las pruebas de detección tienen por objetivo descubrir el cáncer antes de que se presente algún síntoma. La más común y fiable para la detección temprana es la prueba sanguínea del antígeno prostático específico (PSA), es una proteína producida por las células en las glándulas prostáticas, tanto normales como cancerosas. Esta proteína se encuentra principalmente en el semen pero puede aparecer también en pequeñas cantidades en sangre, el aumento de concentración sérica se correlaciona con el aumento de tamaño tumoral, por lo tanto puede ser un buen indicador.

El examen digital del recto, que se conoce como DRE, se resume en la introducción de un dedo en el recto a fin de palpar cualquier abultamiento, es menos fiable que la PSA ya que puede no ser detectado mediante el tacto ("Pruebas de detección para el cáncer de próstata", 2020). En Fig. 4 se muestra cómo se lleva a cabo en la práctica clínica la detección del cáncer de próstata.

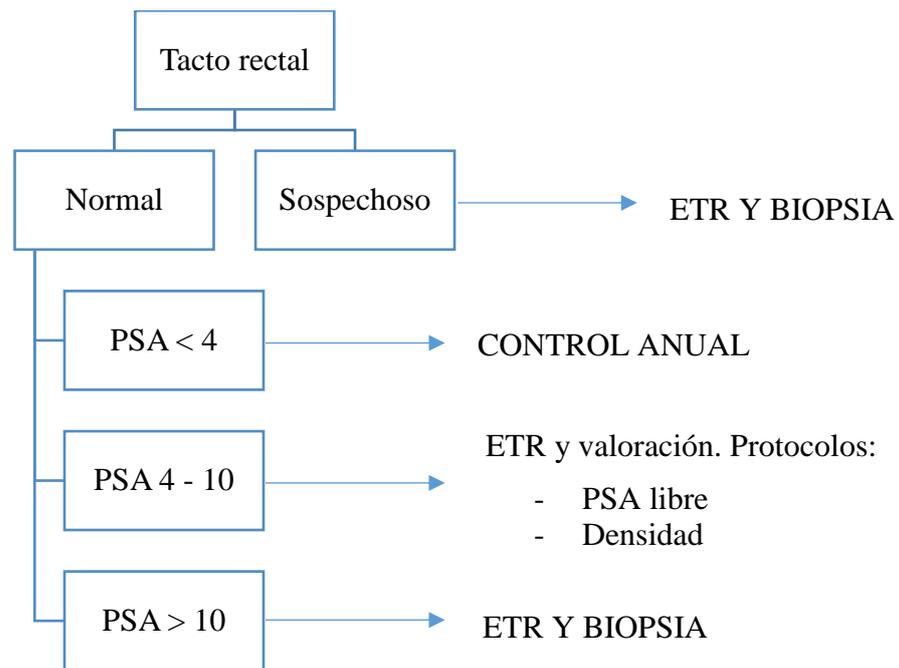
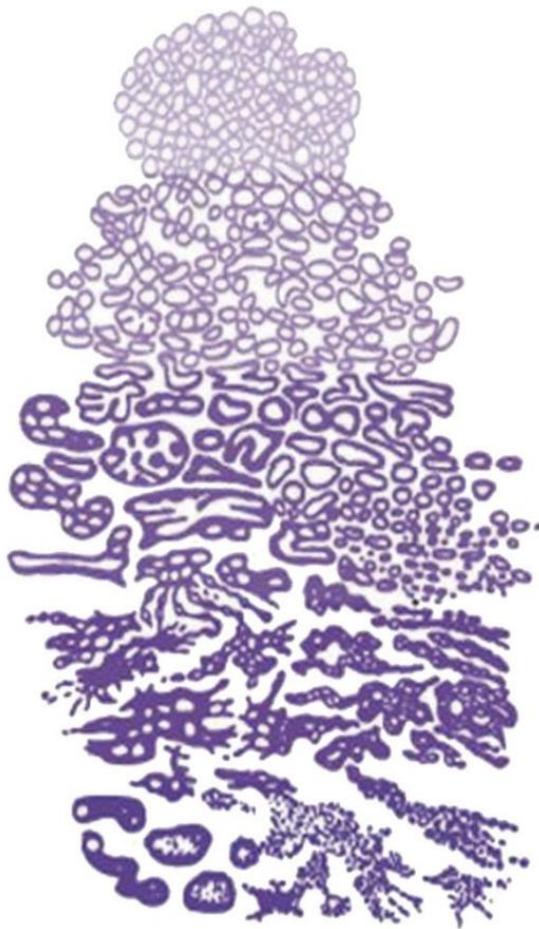


Fig. 4. Práctica clínica de la detección del cáncer de próstata extraído de ("Cáncer de próstata", 2020)

1.1.2. Escala de Gleason

La escala de Gleason es una escala numérica que asigna dos valores según el grado de diferenciación celular y el patrón glandular. Esta medida es la escala clínica que se utiliza en la práctica mediante la observación y análisis con el microscopio de las muestras, además está compuesto por dos principales grados a nivel local en la biopsia, en términos de porcentaje y severidad.

El primero hace referencia al grado más común dentro de la muestra, que puede comprender valores del 3 al 5, y el segundo al grado más alto, que también puede comprender valores entre el 3 y el 5. Aunque realmente la escala de Gleason comprenda valores entre el 1 y el 5, en cortes biopsiados bidimensionales solo se pueden apreciar grados a partir del 3 ("Escala de Gleason", 2020).



Únicamente glándulas individuales y bien diferenciadas, no se observa infiltración.

Predominio de glándulas bien formadas con un componente menor de glándulas pobremente formadas, fusionadas y/o cribiformes. Menor estroma entre las células

Predominio de glándulas pobremente formadas, fusionadas y/o cribiformes con un componente menor de glándulas bien formadas

Únicamente glándulas pobremente formadas, fusionadas y/o cribiformes (patrón 4+4)

Predominio de glándulas bien formadas con un componente menor no glandular (patrón 3+5)

Predominio de componente no glandular con un menor componente de glándulas bien formadas (patrón 5+3)

Ausencia de formación glandular (o con necrosis) con/sin glándulas pobremente formadas, fusionadas y/o cribiformes

Fig. 5. Características histológicas del cáncer de próstata según la escala de Gleason, extraído de (Macías et al., 2020)

El grado se obtiene mediante la suma de los dos valores asignados, donde se toma como valor más bajo el 6 haciendo referencia a la enfermedad menos agresiva y latente y el más alto posible el 10, que representa la enfermedad más agresiva y avanzada (Ortega Alonzo, González Páramo y Galván Losada, 2019). El pronóstico y la agresividad del tratamiento viene determinado por esta puntuación.

Por lo tanto, los grupos correspondientes a los patrones 1 – 3 de Gleason serán aquellos con un Gleason menor o igual a 6 o incluso dependiendo de las características 7, los de patrón 4 aquellos con un Gleason 7 u 8 y por último el patrón 5 que corresponderá con un Gleason 9 – 10.

Tal y como se puede observar en Fig. 5 los distintos patrones de la escala de Gleason se corresponden con distintas características histológicas de las muestras. Las glándulas pobremente formadas y la fusión de ellas es un punto clave dentro de la gradación. En Fig. 6 se puede apreciar en muestras histológicas reales estos patrones de características.

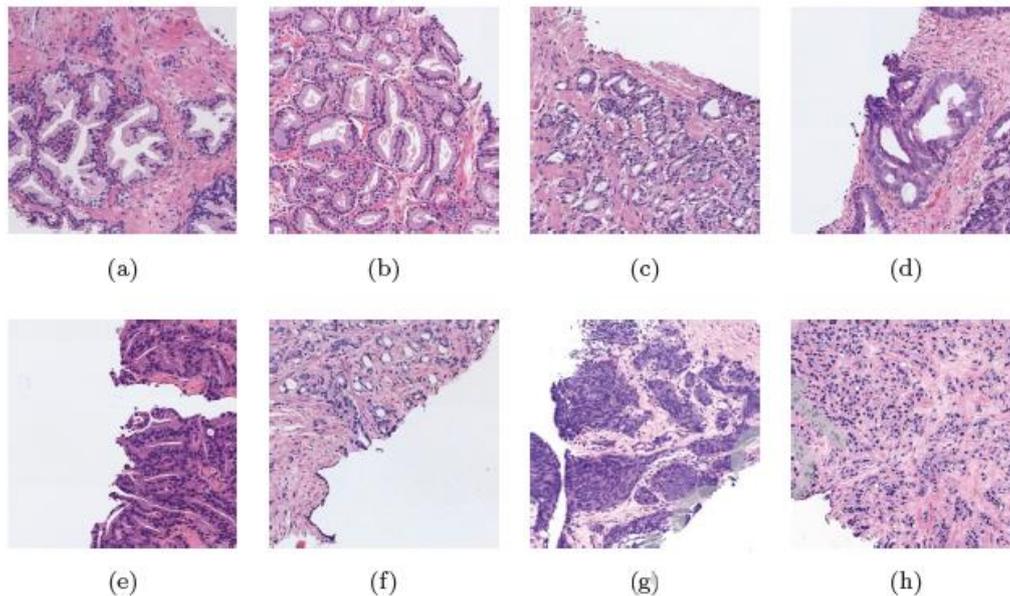


Fig. 6. Patches de muestras histológicas H&E con distintos patrones de Gleason. (a): Glandulas no cancerosas bien diferenciadas; (b): Gleason grado 3 con patrones de densidad atrófica; (c): Gleason grado 4 con grandes patrones glandulares fusionados; (d): Gleason grado 4 con patrones cribiformes; (e): Gleason grado 4 con estructuras papilares; (f): Gleason grado 4 con glándulas pobremente formadas individuales; (g): Gleason grado 5 incluyendo nidos de células sin formación de lumen; (h): Gleason grado 5 con células aisladas extraído de (Silva-Rodríguez et al., 2020).

1.2. Imágenes histológicas

Este apartado se centra en explicar la preparación de las muestras histológicas que posteriormente van a ser empleadas como base de datos del proyecto. Para ello, se muestra la preparación que nos permite observar los distintos cuerpos y características celulares del tejido.

1.2.1. Preparación muestras histológicas en Whole Slide Images (WSI)

La histología es la rama de la biología que estudia la composición, la estructura y las características de los tejidos orgánicos de los seres vivos. Está estrechamente relacionada con la anatomía microscópica y su estudio permite no solo el conocimiento de los tejidos,

sino también de las células que los componen y otros corpúsculos, que nos permite relacionarlo con la bioquímica y la citología, ciencia que estudia las células en su composición y función. (“Histología”, 2020)

Como ya se ha explicado anteriormente, una vez se tiene un diagnóstico sospechoso a través del tacto rectal y el análisis de PSA en sangre, se procede a realizar una biopsia. El trozo de tejido extraído se fija sobre una placa, donde posteriormente se realizará un corte con un micrótopo. Se ha de tener en cuenta que el corte debe ser muy fino, unos 3-5 μ m, ya que los escáneres necesarios para procesar la imagen tienen poca capacidad de enfoque (“Best practices for better digital image quality: Leica Biosystems”, 2020).

Tal y como se ilustra en Fig. 8, se coloca la muestra en el portaobjetos y se realiza la segmentación con el micrótopo, posteriormente se tiñe con hematoxilina y eosina y ya se tienen las muestras listas para ser analizadas (Megías M, Molist P, Pombal MA., 2019).



Fig. 7. Proceso de preparado de las muestras WSIs extraído de (Megías et al., 2019)

1.2.2. Tinción Hematoxilina & Eosina

La mayoría de los tejidos son incoloros y por ello es necesario teñirlos para poder observar sus características a través del microscopio óptico. La tinción con Hematoxilina y Eosina es una de las más comúnmente usadas en histología sobre cortes de parafina.

Esta tinción emplea un colorante básico y otro ácido, hematoxilina y eosina respectivamente, para teñir de diferente color a las estructuras ácidas y básicas de la célula (Megías M, Molist P, Pombal MA., 2019).

Los colorantes no son designados ácido o básico debido al pH, sino en cuanto a la proporción que prevalece en cuanto a las cargas aniónicas o catiónicas que poseen. Por

esto, la hematoxilina es considerada un colorante básico, es decir catiónico, y por ello es afín a estructuras ácidas como el núcleo de las células. Por otro lado, la eosina al ser un colorante ácido o aniónico es afín a estructuras básicas como el citoplasma celular ("Tinción de hematoxilina-eosina: características, usos, técnicas", 2020).

Para realizar esta tinción, se lleva a cabo una serie de procesos como el desparafinado, la hidratación, tinción y deshidratación utilizando distintos productos. A continuación, en Fig. 8, se describe el proceso de tinción de los cortes histológicos con hematoxilina y eosina.

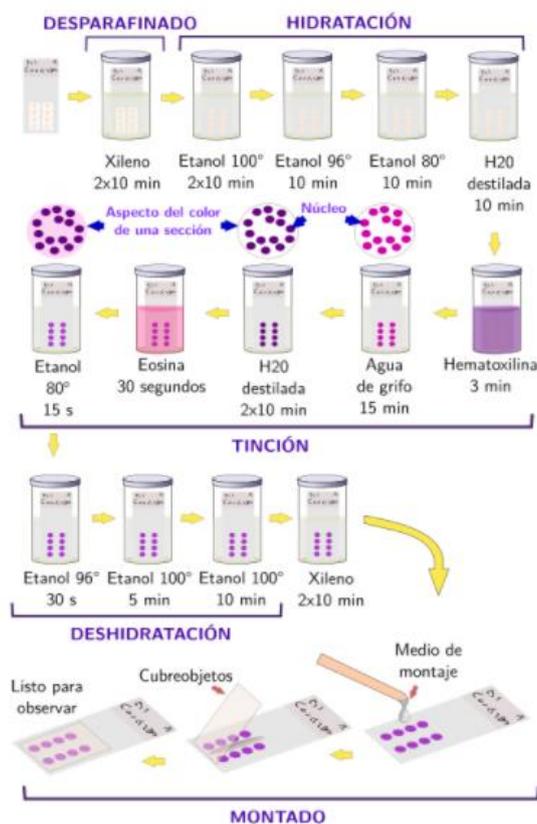


Fig. 8. Proceso de tinción con hematoxilina y eosina, extraído de (Megías et al., 2019)

Por ejemplo, el tejido prostático se reconoce por la forma de las glándulas y por los cuerpos amiláceos o concreciones prostáticas, que son restos de células epiteliales muertas y tal y como se puede observar en Fig. 9 muy eosinófilas y es más posible observarlos en adultos mayores, gracias a este tipo de tinciones se pueden observar.

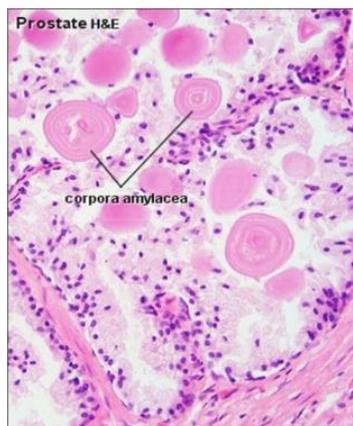


Fig. 9. Cuerpos amiláceos en un corte de tejido histológico de la próstata extraído de ("Histología de la próstata", 2020)

1.3. Análisis de las muestras

Una vez las pruebas preliminares han esbozado un resultado sospechoso, deben realizarse otra serie de pruebas más complejas e invasivas para confirmar el diagnóstico y la gradación del cáncer de próstata. Estas pruebas son la biopsia y la ecografía transrectal (ETTR), las cuales se realizan a la vez. Una vez se realiza la biopsia y se extrae una pequeña cantidad de tejido de distintas áreas de la próstata, se preparan las muestras como se ha descrito anteriormente.

El proceso de observación y gradación es realizado por los patólogos bajo un microscopio, donde trata de evaluar características en las células. El patólogo hace una valoración no solo de la zona afectada sino también del tejido que la rodea ya que también esto influye en el diagnóstico final, para ello el especialista realiza aumentos a distintas resoluciones con el fin de conseguir toda esta información. Este proceso es muy tedioso y subjetivo, tanto intra- como inter-profesionales, además de la cantidad de tiempo que se ha de invertir en cada una de las muestras.

Es por ello, que un sistema CAD podría facilitar esta labor dando un enfoque más objetivo, agilizando el proceso de diagnóstico y con una mayor precisión, además de reducir la carga de trabajo sobre los patólogos.

1.4. Marco del Proyecto

Este Trabajo Fin de Master se desarrolla en el proyecto SICAP, corresponde al acrónimo “Sistemas de interpretación de imágenes histopatológicas para la detección del cáncer de próstata”.

Este proyecto financiado por el Ministerio de Economía, Industria y Competitividad y con la colaboración de médicos especialistas del Hospital Clínic Universitari de València tiene como objetivo desarrollar un sistema de ayuda al diagnóstico para el cáncer de próstata clasificando las imágenes histopatológicas procedentes de biopsias en diferentes estados según la escala de Gleason.

En la práctica clínica, tal y como se ha explicado anteriormente, el análisis de las muestras es un trabajo muy tedioso y subjetivo entre distintos especialistas. Por ello, nace este proyecto, para diseñar y desarrollar un sistema que pueda ayudar a los patólogos a mejorar en términos de tiempo y eficacia así como reducir los niveles de discordancia.

Este TFM pretende implementar técnicas de aprendizaje profundo basadas en *Deep Learning* para la clasificación de las imágenes. Para ello, se trabaja con imágenes a distintas resoluciones para tener en cuenta no solo el tejido patológico en sí, si no también aquel que le rodea. Así, se pretende acercar los sistemas de inteligencia artificial al modo en el cual los patólogos analizan las muestras en la práctica clínica.

1.5. Estado del arte

Los algoritmos de visión computacional han sido ampliamente usados para analizar imágenes histológicas de cáncer de próstata. En este apartado se lleva a cabo un análisis de trabajos presentados anteriormente en la detección del cáncer de próstata por CAD.

La literatura se clasifica en función de tres factores: el tipo de imágenes incluidas en la base de datos analizada, los objetivos dirigidos por los sistemas CAD, y las técnicas propuestas para conseguirlos. Respecto a las imágenes, son usadas principalmente tres tipos de imágenes histológicas: *Whole Slide Images* (WSIs), prostatectomías y Tissue Micro Arrays (TMAs). En este capítulo se centrará en el trabajo generado para la gradación de WSIs.

Tal y como se ha comentado anteriormente, este TFM se lleva a cabo en una de las fases finales del proyecto SICAP, previamente se han realizado distintos análisis de las WSIs. Se realiza la identificación de glándulas y segmentación de las mismas como en García et al. (2018) donde se emplea una técnica de segmentación LWTC, nunca empleada anteriormente en este tipo de imágenes, y García et al. (2019) donde se emplean redes neuronales convolucionales para una posterior clasificación. En García, Colomer y Naranjo (2019), se emplean arquitecturas *fine-tune* para realizar la clasificación de glándulas en estadios tempranos de la enfermedad.

Doyle et al., (2007) describe la problemática en el análisis visual para la gradación del cáncer por la variabilidad intra- e inter-observacional. Para solucionarlo se extraen 102 características morfológicas y de textura basadas en las imágenes, estas características son intrducidas en un *Support Vector Machine* (SVM) para clasificar las imágenes en cuatro clases diferentes: epitelio benigno, estroma benigno, adenocarcinoma de grado Gleason 3 y adenocarcinoma de grado Gleason 4. Los mejores resultados se obtuvieron entre la diferenciación del estroma y el adenocarcinoma de grado 3 con una precisión de 92,8% y los peores entre el adenocarcinoma de grado 3 y 4 con una precisión del 76,9%.

Posteriormente, Gertych et al. (2015) genera mediante la deconvolución de las imágenes de H&E en aquellas formadas por los tejidos teñidos con hematoxilina y eosina respectivamente, para capturar las diferencias estructurales entre el estroma y las glándulas. Aquí se empleó un SVM seguido de un árbol de decisión o *Random Forest*

(RF) para digitalizar la estratificación del tejido prostático en estroma, epitelio benigno y cáncer de próstata.

Jiménez del Toro et al. (2017) empieza a introducir arquitecturas de *Deep Learning* con redes neuronales convolucionales (CNN) como LeNet, AlexNet o GoogLeNet para identificar regiones de interés (ROIs), estas arquitecturas tienen pesos predeterminados que mejoran la eficiencia del modelo lo que se conoce como *fine-tune*. Los resultados mostraron que se podía conseguir una precisión del 73,52% con GoogLeNet y *Data Augmentation*.

Por otro lado, en Ren et al. (2017) se genera un método de clasificación que comienza con una segmentación mediante CNN, y luego una gradación en función de la segmentación se extraen una serie de características y la gradación entre grado 3 y 4 se hace mediante una regresión con RF y un *cross-validation*. Este método de clasificación obtuvo un porcentaje de precisión del 83%.

Ing et al. (2018) demuestra la idoneidad de las CNN para la detección y gradación del cáncer de próstata. En este estudio se prueban cuatro arquitecturas de CNN a partir de imágenes clasificadas manualmente, los resultados de estas pruebas esbozan resultados cercanos al 0.9 de precisión. Una vez ya se ha demostrado la eficiencia de CNN, Esteban et al. (2019) propone distintos modelos para obtener mejores resultados de clasificación en imágenes histológicas de próstata y aquel que obtiene mejores resultados de precisión es el modelo VGG19.

Finalmente, en Lucas et al. (2019) se muestra que es posible con CNN diferenciar entre muestras de grado 3 y 4, aunque encontramos mayor precisión entre aquellas no patológicas y malignas, esta diferencia se relaciona con las anotaciones realizadas por distintos patólogos. Esto demuestra que las CNN pueden ser una valiosa herramienta para los sistemas de ayuda al diagnóstico por ordenador.

Por lo tanto, tal y como nos muestra la literatura, en múltiples estudios se ha mostrado la capacidad de los modelos basados en CNN para generar sistemas de ayuda al diagnóstico, para ello tenemos modelos pre-entrenados que nos aporta unos pesos predefinidos antes del entrenamiento y no aleatorios, factor que mejora el rendimiento del modelo. Además, otras técnicas como el aumento del set de datos, o *Data Augmentation*, y también la validación cruzada puede mejorar la clasificación de los sistemas de diagnóstico.

Lo que también se observa en la literatura reciente es que en algunos casos como por ejemplo Esteban et al. (2019) estudian distintos modelos para diferentes resoluciones y cual aporta mejores resultados. El problema es que en la práctica clínica se analiza una misma imagen a distintas resoluciones ya que no solo importa la zona patológica o afectada, sino también el contexto para dar un diagnóstico preciso.

En los estudios más recientes como Rodríguez-Silva et al. (2020) se desarrollan distintos modelos aplicando arquitecturas *fine-tuning* con WSIs, se observa como entre los modelos predefinidos el que mejor se ajusta al set de datos es VGG19 con un índice kappa de 0.7474.

Por lo tanto, en base a todo lo observado anteriormente y planteando la hipótesis de que el patólogo realiza sus predicciones analizando las muestras a distintas resoluciones para tener detalles tanto globales como locales de la muestra, se decide generar una red multi-resolución cuyas entradas sean las imágenes a distintas resoluciones y así generar modelos más precisos y fiables.

Capítulo 2

Objetivos

El objetivo último de este Trabajo Fin de Master es el diseño y desarrollo de un sistema de diagnóstico a distintas resoluciones para conseguir mejorar la precisión en la gradación del cáncer de próstata a partir de muestras de biopsias en *Whole Slide Images* (WSIs).

Se desarrollan una serie de objetivos secundarios que permiten llegar al objetivo final:

1. Entender la problemática del diagnóstico del cáncer de próstata y revisar la literatura científica actual para conocer el estado del arte de los sistemas de diagnóstico por imagen histológica del cáncer de próstata y particularmente de los sistemas formados por CNN.
2. Generación de la base de datos a partir de parches de muestras de biopsias que forman parte del proyecto SICAP, para poder trabajar con una base de datos a distintas resoluciones.
3. Creación del *Ground Truth* a partir de las anotaciones de los patólogos mediante máscaras. Estas especifican las zonas donde se encuentra un patrón patológico.
4. Partición de los datos en distintos conjuntos para entrenamiento, validación y testeo.
5. Estudiar distintas formas de diseñar la red neuronal, técnicas y arquitecturas utilizadas en la práctica, así como los parámetros a tener en cuenta y que efecto tienen estos en la arquitectura y entrenamiento de la red.
6. Desarrollo de redes neuronales formadas por imágenes de una única resolución, tratando de optimizar los valores de los hiperparámetros para conseguir el equilibrio entre precisión y tiempos de compilación de los datos.

7. Desarrollo de un generador de datos sintético que permita agrupar las imágenes de las tres resoluciones en función de una única etiqueta. Este generador de datos además debe poseer sus propios atributos y funciones adaptadas al problema.
8. Desarrollo de la red multi-resolución, teniendo como entrada las imágenes a distintas resoluciones y teniendo en cuenta lo que eso supone dentro de la red. Además, se debe tratar de optimizar lo máximo posible los hiperparámetros.
9. Evaluación de los resultados obtenidos tras el entrenamiento de los modelos mediante distintos parámetros y coeficientes que nos permitan evaluar la precisión y robustez del modelo.
10. Identificar los problemas y las limitaciones observadas, además de proponer posibles mejoras solventar estos problemas encontrados.

Capítulo 3

Material y Metodología

3.1. Material

En esta sección se explican los materiales empleados en este TFM. Principalmente, consta de la base de datos del proyecto SICAP, los distintos *softwares* que van a ser empleados junto con la librerías descritas a lo largo del capítulo, así como el *hardware* que permite compilar dicho proyecto.

3.1.1. 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 imágenes se encuentran digitalizadas y almacenadas en una base de datos privada cuyo objetivo es garantizar la confidencialidad del paciente.

Esta base de datos está compuesta por 18.784 parches de 512x512 píxeles (1.99 GB), estas imágenes corresponden con 181 biopsias diferentes de distintos pacientes, las cuales están anotadas por especialistas de anatomía patológica mediante 8.064 máscaras de las mismas imágenes. Por consiguiente, se dispone de imágenes anotadas e imágenes no anotadas. Se puede observar como son las máscaras que identifican las regiones patológicas de las muestras en Fig. 10.

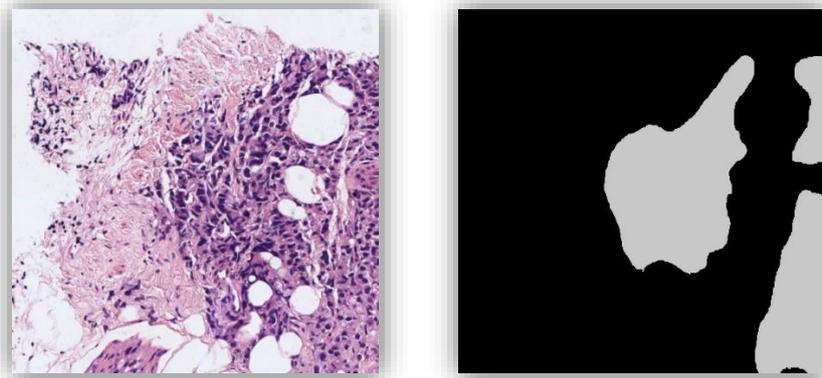


Fig. 10. (a) Ejemplo imagen de la base de datos de 512x512 píxeles (b) Máscara asociada a la imagen anterior

De estas imágenes que se encuentran anotadas la mayoría corresponden a aquellas de grado 4, el 57,7%, le prosiguen aquellas de grado 3 con un 29,1% de las muestras y tan solo el 11,7% de las muestras corresponden a aquellas de grado 5. Únicamente un 1,6% de las muestras no contienen información patológica.

Grado	Número de parches
No patológico	37
Grado 3	2.344
Grado 4	4.653
Grado 5	1.028

Tabla 1. Número de parches anotados con los distintos grados de la escala de Gleason

Del resto de los parches que se encuentran no anotados, únicamente se emplean aquellos cuya biopsia haya sido no patológica. Esta información se encuentra en la base de datos del proyecto SICAP y aumenta el número total de parches no patológicos a 4454. Por lo tanto, el modelo va a ser entrenado con un total de 12.480 parches.

3.1.2. Software empleado

Para el desarrollo y ejecución del algoritmo se emplean dos softwares diferentes. Primero, se emplea MATLAB® v.R2020a para la preparación de la base de datos, la creación del *Gound Truth* y el post-procesado. Posteriormente, para el diseño y desarrollo del algoritmo se utiliza Pycharm v.2020.1.3 junto con MobaXterm v.20.2 para generar una conexión remota con el servidor de CVBLab.

MATLAB® v.R2020a es un programa computacional que ejecuta operaciones y tareas matemáticas. Su nombre significa “MATrix LABoratory” y en primer lugar fue diseñado para trabajar con vectores y matrices. Es una de las aplicaciones más empleadas en ingeniería por ser interactivo, compatible con una gran cantidad de software, posee una gran cantidad de herramientas, llamadas *Toolboxes* y permite elaborar programas y aplicaciones que puedan ejecutarse en cualquier computador. (“¿Qué es MATLAB? – ACAPMI Blog”, 2020)

Pycharm v.2020.1.3 es un *Integrated Development Environment* (IDE) de Python, esto es una aplicación destinada a dar una serie de herramientas para desarrolladores que permite generar códigos de una forma más sencilla y didáctica. (“PyCharm: uno de los mejores IDE para Python — Escuela de Python”, 2020) Python contiene una amplia cantidad de paquetes y librerías que soportan múltiples funciones, muy útiles en *Deep Learning*.

Algunas de las librerías que se van a emplear son, principalmente *keras* para el desarrollo del algoritmo pero también se utilizarán otras como *pandas*, *numpy*, *matplotlib.pyplot* y por último *sklearn* para la evaluación de los resultados. MobaXterm v.20.2 sirve para realizar una conexión remota SSH entre el *interpreter*, Pycharm, y el servidor externo para poder tener acceso a la base de datos y librerías.

3.1.3. Hardware empleado

El Proyecto se ha desarrollado y ejecutado con un ordenador portátil HP Notebook 15-ac006ns, con un procesador Intel Core i5-5200U con una velocidad de 2,20 GHz. Este dispositivo posee una memoria RAM de 8,00 GB (DDR3L-1600 SDRAM) y con gráficos de vídeo Intel HD Graphics 5500. Para el entrenamiento de los modelos también se ha hecho uso de los servidores de computación de CVBLab, los cuales cuentan con distintas GPU Titan V.

Es importante conocer las características del dispositivo utilizado para el desarrollo del proyecto, ya que los tiempos de ejecución, la velocidad y el rendimiento dependen de la capacidad de las características internas del *hardware*.

3.2. Metodología

Para llevar a cabo los objetivos descritos en el capítulo anterior, se debe seguir un flujo de trabajo a fin de conseguir el objetivo último (véase Fig. 11). A partir del generador de datos hay que tener en cuenta que se debe realizar tanto para las tres redes de resoluciones únicas como para la red multi-resolución, por lo tanto, en cada apartado se irán explicando paralelamente ambos casos.

Los tres primeros pasos, preparación de la base de datos, creación del *ground truth* y partición de los datos, se lleva a cabo con MATLAB® v.R2020a. Por su parte, el diseño, desarrollo, entrenamiento y evaluación de los resultados se realiza con Pycharm v.2020.1.3.

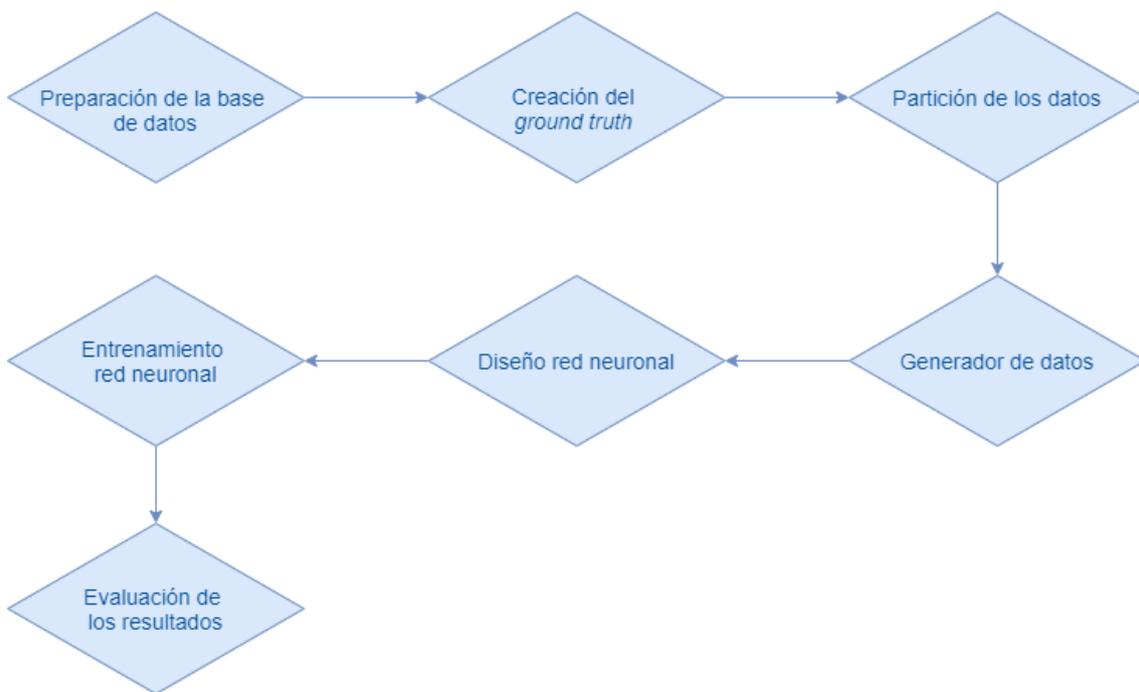


Fig. 11. Diagrama de flujo de la metodología end-to-end llevada a cabo para la consecución de los objetivos planteados en este proyecto.

3.2.1. Preparación de la base de datos

Las imágenes histológicas en WSIs con una resolución 512x512x3 son tomadas, y con las anotaciones generadas por los patólogos, se realiza una gradación de las muestras donde se anota el grado mayoritario. Para ello, con MATLAB® v.R2020a se extrae el

histograma de las máscaras, en función del grado de gris mayoritario (sin tener en cuenta el valor 0 correspondiente al fondo de la imagen) se asigna el grado de la muestra. En Fig. 12, se muestra un ejemplo de una muestra cuyo *ground truth* es 5, ya que el nivel de gris predominante toma un valor de 185 que, según la Tabla 2, corresponde con un grado 5 (derivado del subpatrón 5+4).

Grado	Nivel de gris	
3	50	} Grado 3
3+4	65	
3+5	75	
4+3	85	} Grado 4
4	100	
4 cribiform	150	
4+5	165	} Grado 5
5+3	175	
5+4	185	
5	200	
Other	255	

Tabla 2. Gradación de la escala de grises según las máscaras

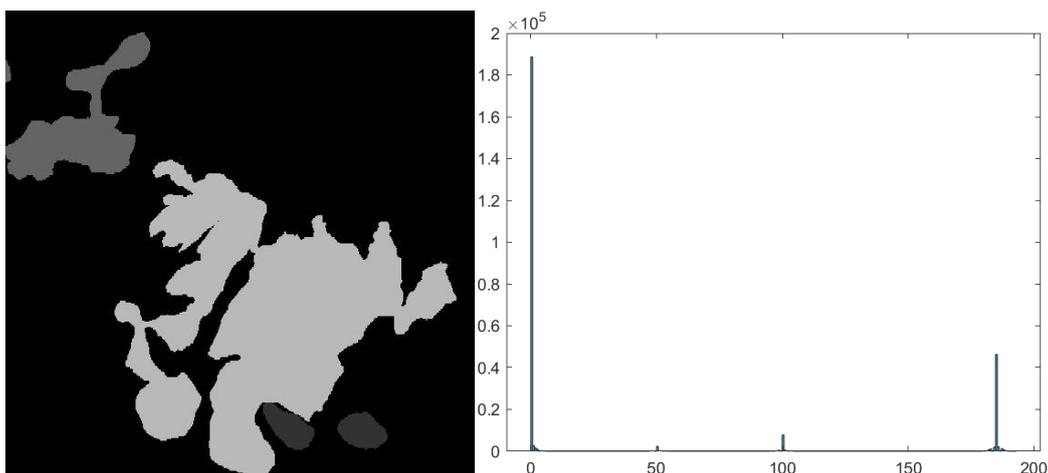


Fig. 12. Máscara con las anotaciones en función de la escala de grises de la Tabla 2 (b) Histograma asociado con la máscara anterior

Es importante señalar que, de cara a implementar la perspectiva de multi-resolución, cada imagen de 512x512 ha sido dividida en 4 y 16 subimágenes con un tamaño de 256x256 y 128x128, respectivamente (véase Fig. 13). Siguiendo el procedimiento anterior, a cada subimagen (parche) se le asignó la etiqueta del groundtruth en base al nivel de gris predominante de su máscara correspondiente.

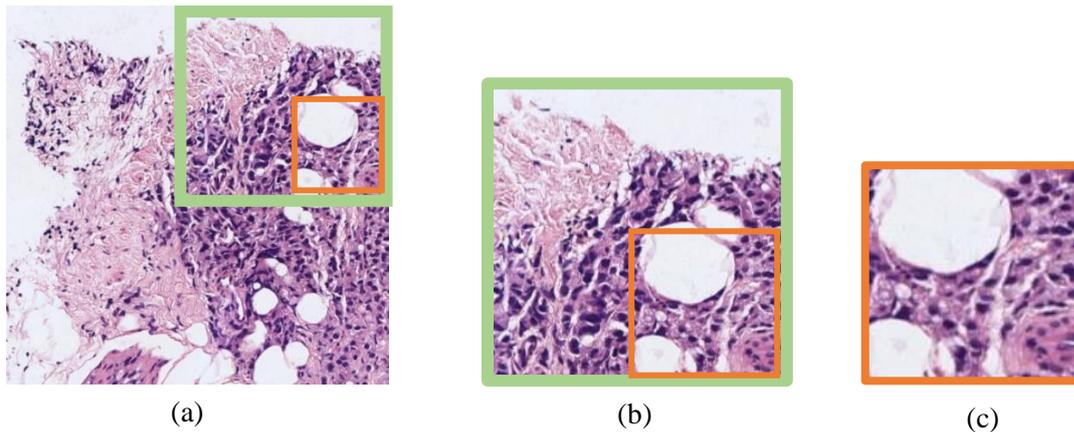


Fig. 13. Ejemplos de las tres resoluciones de imagen que componen el framework de este trabajo. (a) Imagen resolución 512x512. (b) Parche asociado a la imagen (a) de resolución 256x256. (c) Parche asociado a (a) y (b) de resolución 128x128

3.2.2. Creación Ground Truth

A parte de la base de datos en sí, es importante preparar el *DataFrame*¹ ("Intro to data structures — pandas 1.1.1 documentation", 2020) que relacionará los parches con sus etiquetas correspondientes. En este TFM se tienen 4 etiquetas: 'No patológico', 'G3', 'G4' y 'G5'.

Estas 4 etiquetas consisten en datos categóricos, cuyo problema es que algunos algoritmos no pueden operar con ellos directamente, es necesario un *input* y un *output* numérico. Esto significa que todos los datos categóricos (tanto las etiquetas del *ground truth* como las predicciones de los modelos) tienen que ser pasados a forma numérica. Existen dos formas de convertir los datos categóricos a datos numéricos, *Integer Encoding* y *One-Hot Encoding*.

El primero, *Integer Encoding*, consiste en una relación entre el valor numérico y el valor categórico (véase Tabla 3). El problema es que el algoritmo puede interpretar un orden

¹ Un *DataFrame* es una serie de datos estructurados bidimensionalmente y etiquetados por columnas.

natural entre las categorías y puede desencadenar en predicciones a medio camino entre categorías.

Para que este no suceda, *One-Hot Encoding* puede ser aplicado para la codificación de números enteros. En este caso, se elimina la variable codificada como entero y se agrega una nueva variable binaria para cada valor entero único. Por ejemplo, como se puede observar en la Tabla 4 se coloca un valor “1” en la variable binaria para el color y valores “0” para los otros colores. (Brownlee, 2020)

<i>Color</i>	<i>Valor</i>
<i>Red</i>	1
<i>Green</i>	2
<i>Blue</i>	3

Tabla 3. Ejemplo codificación con Integer Encoding

<i>id</i>	<i>Red</i>	<i>Green</i>	<i>Blue</i>
<i>1</i>	1	0	0
<i>2</i>	0	1	0
<i>3</i>	0	0	1

Tabla 4. Ejemplo codificación con One-Hot Encoding

El *ground truth* que se asumirá para la red multi-resolución será el de las imágenes de mayor resolución. Por lo tanto, las etiquetas de las 3 resoluciones, se relacionarán con la de menor tamaño respectivamente, es decir, 128x128.

3.2.3. Partición de los datos

El conjunto de los datos es dividido en tres *DataFrames*, que corresponden con los conjuntos de entrenamiento, validación y test. Esta partición se realiza para evitar el sobreajuste u “*overfitting*”. Esto se produce cuando el modelo está sobre-entrenado se ajusta tan detalladamente al conjunto de datos de entrenamiento que pierden gran parte

de su poder predictivo y es robusto para predecir sobre otro conjunto de datos. ("Datos de entrenamiento vs datos de test - Think Big Empresas", 2020)

Un 64% de la base de datos conforma el conjunto de datos de entrenamiento, que es aquel que se emplea durante el proceso de aprendizaje y se utiliza para ajustar los distintos parámetros, como por ejemplo los pesos de la red o los hiperparámetros del modelo como el *learning rate*, el *batch size*, el optimizador o el número de épocas, entre otros. Un 19% del conjunto de datos corresponde con los datos de validación, estos datos sirven para determinar la bondad de los hiperparámetros definidos y controlar el *overfitting* del modelo. Por último, el conjunto de datos de test, que corresponde con el 17% de los totales, sirve para realizar predicciones con un grupo de datos con la misma distribución probabilística que aquellos de entrenamiento y, por lo tanto, hace una evaluación del modelo. ("Training, validation, and test sets", 2020)

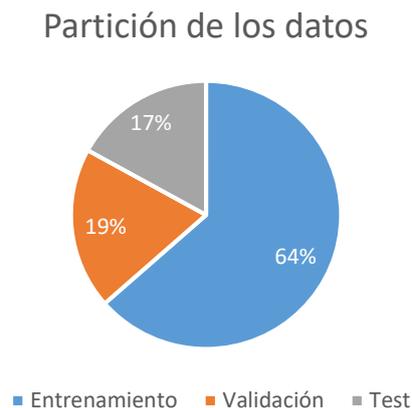


Fig. 14. Partición del conjunto de datos en entrenamiento, validación y test

3.2.4. Generador de datos

La generación de datos artificiales o más comúnmente conocido como *Data augmentation* sirve para crear imágenes sintéticas de manera artificial (con la misma etiqueta que la imagen original de la que proceden) a fin de mejorar la precisión, la generalización, y controlar el *overfitting*. Esta técnica ha sido empleada en los modelos de resoluciones únicas para aumentar el set de datos mediante pequeñas modificaciones en la imagen como se puede observar en Fig. 15.

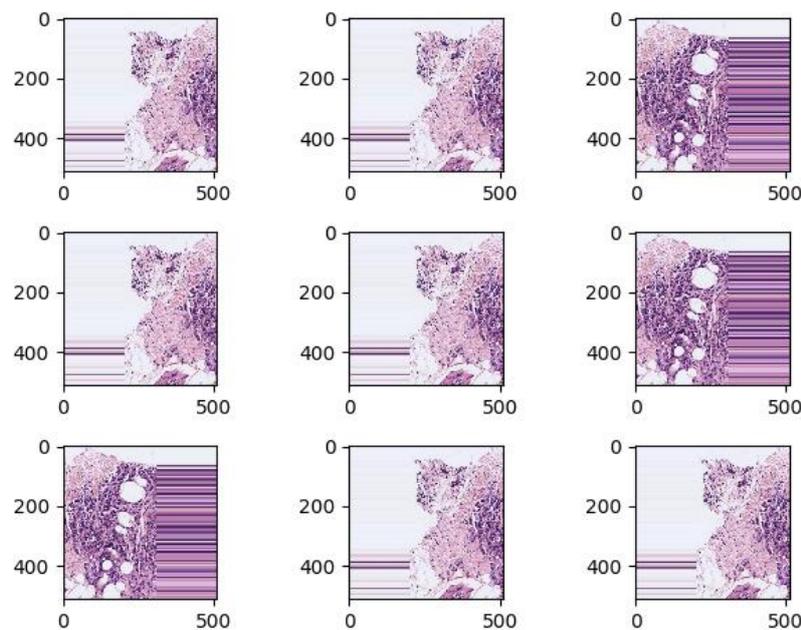


Fig. 15. Ejemplo Data Augmentation Random Horizontal Shift (Desplazamiento horizontal aleatorio)

En el diseño y desarrollo de modelos generados con imágenes de una única resolución, han sido implementados varios cambios en las imágenes para aumentar el set de datos. Primero, se normalizan las imágenes y se aplican distintos cambios como rotación, desplazamiento horizontal y vertical, transformación de corte –inclinación de la forma de la imagen – y un *flip* horizontal a las imágenes.

También, ha sido reducido el tamaño de las imágenes para obtener menores tiempos de ejecución a la hora de entrenar el modelo, siendo estos 256, 128 y 64 píxeles para aquellas de tamaño 512, 256 y 128, respectivamente.

Para la red multi-resolución ha sido creado un generador de datos para poder agrupar los parches de las distintas resoluciones que van a ser empleados. Cada imagen de menor tamaño, 128x128, se relaciona con su correspondiente parche de 256x256 y con el parche de 512x512.

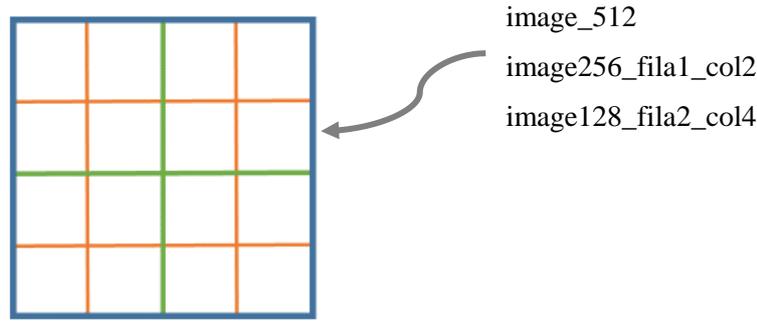


Fig. 16. Visualización de la agrupación de las imágenes en función de su posición

Tal y como se puede observar en Fig. 16 se asignan las imágenes a una mayor en función de su posición dentro de ella. Es decir, una imagen de un cuadrado naranja, 128x128, dentro de uno verde asociado, 256x256 y dentro de uno azul asociado, 512x512. Estas tres imágenes tienen asociado una etiqueta, que corresponde con la etiqueta de la imagen de mayor resolución, es este ejemplo, con la etiqueta de image128_fila2_col4.

$$\left(\begin{array}{l} [\text{image512}, \text{image256_fila1_col1}, \text{image128_fila1_col1}], [1,0,0,0] \\ [\text{image512}, \text{image256_fila1_col1}, \text{image128_fila1_col2}], [0,1,0,0] \\ [\text{image512}, \text{image256_fila1_col2}, \text{image128_fila1_col3}], [0,0,1,0] \\ [\text{image512}, \text{image256_fila1_col2}, \text{image128_fila1_col4}], [0,0,0,1] \end{array} \right)$$

El generador de los datos sintético incluye un atributo propio llamado ‘modo’ que permite separar los datos en función de la partición que se le haya asignado previamente y aparezca dentro del *DataFrame*.

Una vez se tienen asociadas estas imágenes, son cargadas por el generador de datos, las imágenes son normalizadas para corregir efectos en la imagen como bajo contraste o el exceso o falta de luz. También, se reduce el tamaño de las imágenes para mejorar los tiempos de ejecución, aun asumiendo perder cierto detalle de la imagen.

Esta reducción será en 128, 64 y 64 píxeles en las de tamaño 512, 256 y 128 píxeles respectivamente. Se satura el tamaño de la reducción de los parches a 64, ya que si con

los parches de 128 se hiciera la reducción proporcional al resto de resoluciones, tendrían una resolución de 32 píxeles y se puede perder mucha información de la imagen.

3.2.5. Entrenamiento

El *deep learning* es utilizado para distintas aplicaciones que se acercan íntimamente al modo de funcionamiento del sistema nervioso humano. Esto es porque la propia red neuronal se basa en un aprendizaje a partir de ejemplos, es decir, extrae las características a partir de las muestras y con ellas realiza la clasificación aportándole a cada característica un peso determinado para cada etiqueta.

Para entrenar la red neuronal se han de seguir dos pasos: *feedforward* y *backforward*. Durante el *feedforward* se introduce un *input* a la red y posteriormente se calcula el *output* de esta. Una vez la red extrae un resultado este se compara con el resultado real y se calcula cuanto de lejos está del extraído mediante la función de pérdidas, el objetivo es tratar de minimizar este resultado. Posteriormente, se propaga el error y se actualizan los pesos de los nodos y *bias*, esto es lo que se conoce como *backpropagation*. (García, 2019)

Capas de la red neuronal

La red neuronal está formada por tres tipos diferentes de capas: las de entrada, las ocultas y las de salida. En *deep learning* es característico encontrar un mayor número de capas ocultas, ya que no solo realiza la clasificación en función de las características extraídas manualmente sino que también las extrae de manera automática mediante operaciones de convolución. A continuación, se van a explicar las capas que han sido utilizadas para este TFM de la librería *keras* y, particularmente, cómo y porqué han sido empleadas así.

- *Input*: Es la capa de entrada y tiene la misma cantidad de neuronas que el número de píxeles que forma la imagen, sirve para instanciar un tensor de *keras*. Por lo tanto, el tamaño de entrada debe ser el mismo que el de las imágenes que se van a emplear.

- *Capas convolucionales*: Esta capa crea un *kernel* de convolución que se convoluciona con la capa de entrada para producir un tensor de salidas. Estas capas permiten extraer características de bajo nivel a la entrada del y, en capas más profundas, características de más alto nivel, por ejemplo objetos o formas. ("Combinación de capas para redes neuronales convolucionales - sitiobigdata.com", 2020)

Se introducen distintos parámetros para conseguir extraer las características, entre ellos encontramos:

- *Filters*: Número *kernels* que serán aplicados en la imagen.
- Tamaño del *kernel*: Dimensión de la matriz que va a ir corriendo a lo largo de la imagen para extraer las características que se buscan. Por lo tanto, a menor tamaño mayor detalle se tendrá, así que los *kernels* que se aplican son de tamaño 3x3. En Fig. 17 se observa cómo se extraen las características y el resultado de la salida.

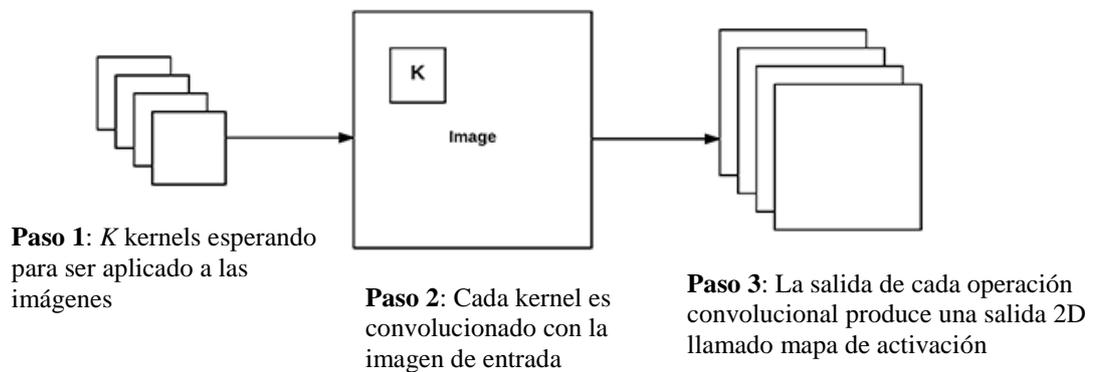


Fig. 17. Representación del funcionamiento de las capas convolucionales en las imágenes y el output de estas, extraído de (Rosebrock, 2020)

- *Padding*: *Padding* significa relleno, y puede tener dos tipos de valores “*valid*” o “*same*”. *Valid* es cuando permite que se modifique el tamaño original de la imagen, en el caso de escoger *same*, después de aplicar el *kernel* se rellenará con ceros hasta conseguir el tamaño original de la imagen. En la red que se va a trabajar se emplea *padding same*.
- *Activación*: Esta función puede llevarse a cabo mediante una capa propia o mediante un argumento dentro de la capa. El parámetro de activación aplica a

la salida de la convolución una función que no modifica la dimensión de la capa. Las más comunes entre otras son la función sigmoide y la función de unidad simplificada (ReLU) que se encuentran representadas en Fig. 18. En la red que se genera en este trabajo se emplea la activación ReLU que elimina los valores negativos y los pone a 0. (Rosebrock, 2020)

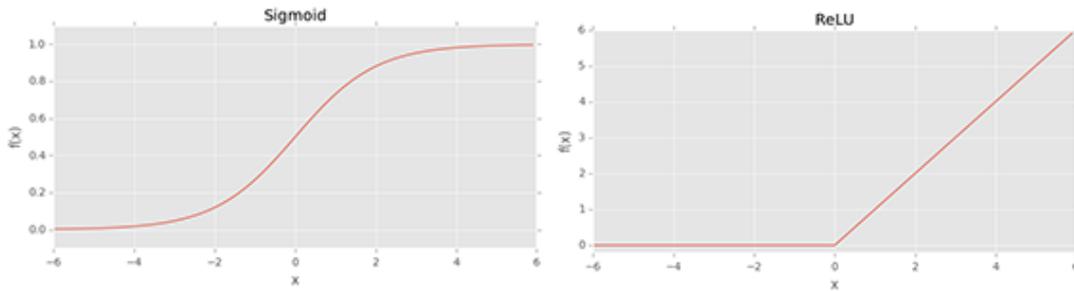


Fig. 18. Representación de las funciones de activación, extraído de (Rosebrock, 2020)

$$f(x) = \frac{1}{1 + e^{-x}} \quad (3.2.5.1)$$

$$f(x) = \begin{cases} 0, & x < 0 \\ x, & x \geq 0 \end{cases} \quad (3.2.5.2)$$

- *Capa de max pooling:* Reduce la representación de los datos de entrada tomando el valor máximo sobre la ventana definida por *pool_size* para cada dimensión a lo largo del eje de características. *Pool_size* es el tamaño de la ventana respecto al que se tomará el máximo, por ejemplo si como en este caso se emplea una ventana de 2x2, es esa agrupación se escoge el máximo valor. En Fig. 19 se puede observar la ejecución práctica de la capa. ("Keras documentation: MaxPooling2D layer", 2020)

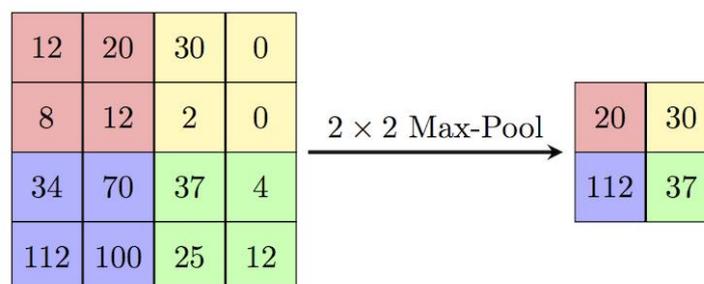


Fig. 19. Ejecución práctica de la capa max pooling extraído de ("Max-pooling / Pooling - Computer Science Wiki", 2020)

- *Capa global max pooling*: Similar a la capa anterior, sólo que el máximo se extrae de la capa total, consiguiendo finalmente reducir la dimensión de las características en una única. Como se puede observar en Fig. 20, se tiene una línea de características única, reduciendo la cantidad de pesos.

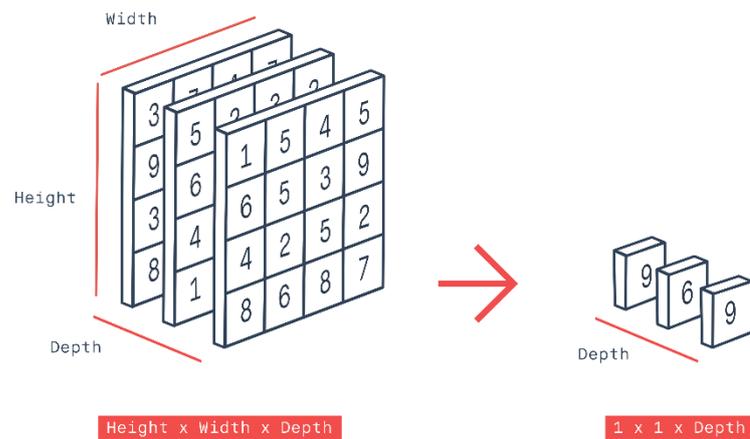


Fig. 20. Funcionamiento de la capa global max pooling a lo largo de los volúmenes de características extraído de ("Introduction to Capsule Networks | Paperspace Blog", 2020)

- *Capa de concatenación*: Esta capa se encarga de concatenar una lista de entradas. Toma como entrada una lista de tensores, todos de la misma forma excepto por el eje de concatenación que en este caso es la profundidad, y devuelve un solo tensor que es la concatenación de todas las entradas. ("Keras documentation: Concatenate layer", 2020)
- *Capa fully-connected*: Está formada por un número determinado de nodos que sirve para conectar todas las características de los parches y que aporta un peso determinado a cada una de las características para cada una de las imágenes. Por lo tanto, la capa está formada en este caso por cuatro nodos, los mismos que el número de etiquetas. La función de activación en este caso, de la capa final es *softmax*, esta función pasa los pasos de las características a probabilidades y se las asigna a cada clase (véase la Ecuación 3.2.5.3).

$$\begin{bmatrix} 1.3 \\ 5.1 \\ 2.2 \end{bmatrix} \rightarrow \frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}} \rightarrow \begin{bmatrix} 0.02 \\ 0.93 \\ 0.5 \end{bmatrix} \quad (3.2.5.3)$$

Arquitectura del modelo

Una vez explicadas las diferentes capas que pueden conformar la arquitectura del modelo, se explica a continuación la utilizada durante este proyecto. En este sentido, se pueden utilizar redes entrenadas *from scratch*, es decir, desde cero o mediante técnicas de *fine-tuning* que permiten utilizar redes pre-entrenadas.

En este trabajo, se han aplicado métodos de *fine-tuning* sobre la arquitectura VGG16, que es una de las más comunes del estado del arte. De esta manera, los pesos de la arquitectura no se inicializan de forma aleatoria, sino que aprovechan los pesos de dicha arquitectura cuando esta fue pre-entrenada sobre el conjunto de datos de ImageNet, que es una base de datos de más de 14 millones de imágenes correspondientes a 1000 clases diferentes.

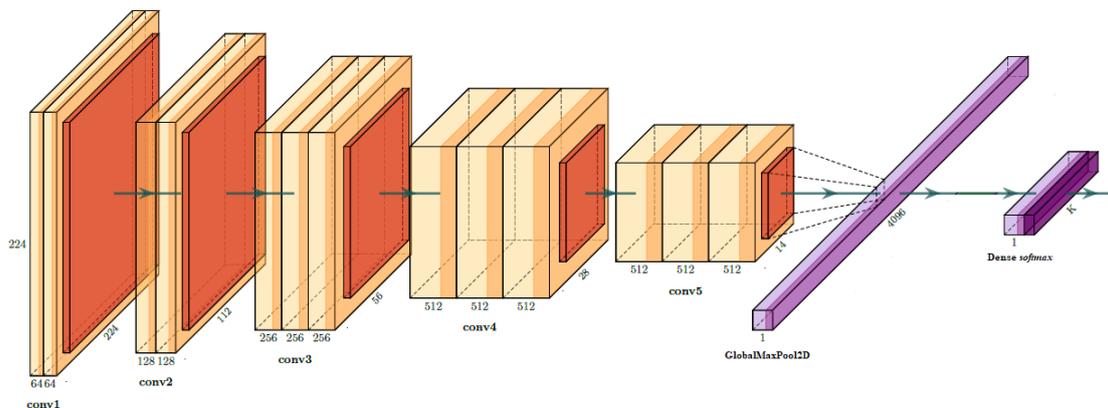


Fig. 21. Arquitectura del modelo para una única resolución

Se importa el modelo VGG16 de la librería `keras.applications.vgg16`. Estos modelos están formados como se puede observar en Fig. 21 por capas *Conv2D*, que son aquellas coloreadas de un naranja más suave de donde se obtienen los mapas de activación. Posteriormente pasan por una capa *MaxPooling2D* que son aquellas, que se encuentran coloreadas por un naranja más intenso, para reducir la dimensión de estas tal y como se ha explicado anteriormente.

Como se ha explicado previamente, se aplican técnicas de *fine-tuning*. Estas técnicas permiten no solo utilizar pesos pre-entrenados, sino también fijar (congelar) los pesos de determinadas capas convolucionales y reentrenar el resto. En concreto, para este trabajo, se dejan ‘congelados’ los pesos de los tres primeros bloques convolucionales, y se reentrenan el resto haciendo uso de las imágenes histológicas de próstata que le aportan a la red el carácter específico para abordar el problema en cuestión.

No se incluye la parte superior del modelo (*top model*), que corresponde con las capas moradas de Fig. 21, ya que va a generarse otras capas para esta parte que se describirán a continuación. Cada una de las imágenes a distintas resoluciones, pasan por el modelo anterior para extraer las características de cada una de ellas.

Una vez definido el “*base model*”, se aplica una capa de concatenación para anexionar las características extraídas por los mapas de activación de cada una de las resoluciones tras pasar por la capa de *global max pooling*. Finalmente, las características concatenadas suponen la entrada al *top model*, el cual está compuesto únicamente por una capa *Dense* con activación *softmax* para, como se ha explicado en el apartado anterior, extraer las probabilidades en cada una de las clases.

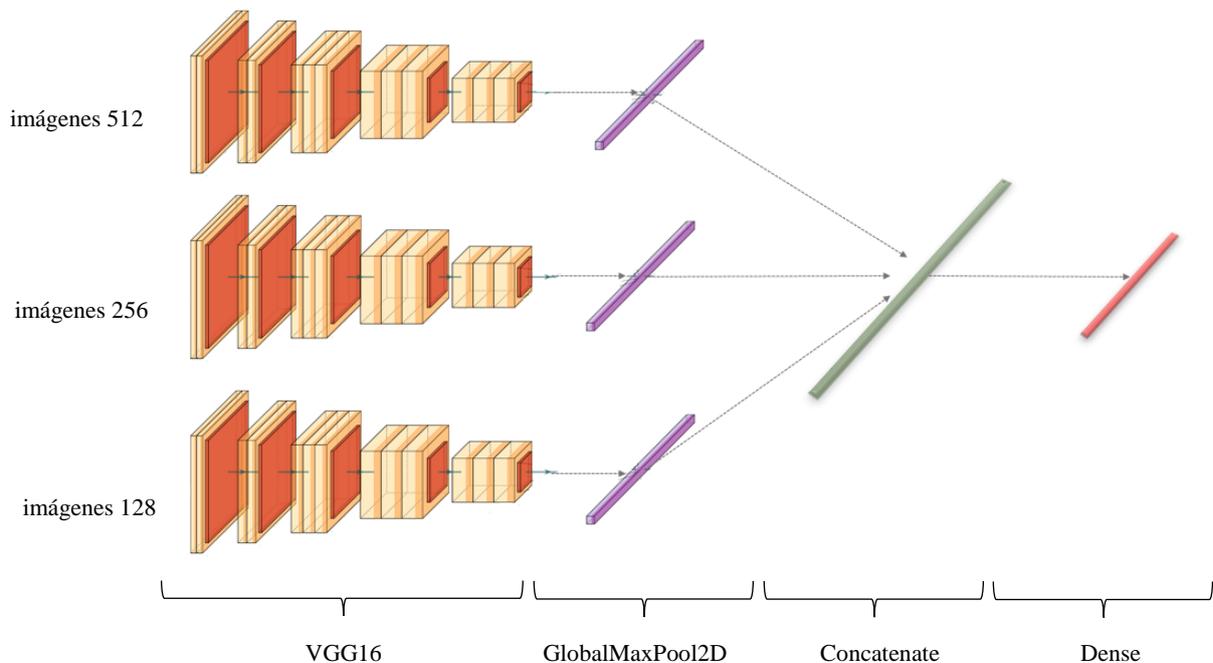


Fig. 22. Arquitectura del modelo multi-resolución

En el caso de la red multi-resolución, una vez todas las características se extraen se la arquitectura definida anteriormente, son unidas mediante una capa de concatenación y ahí se aplica la capa *fully-connected* para extraer las probabilidades de las características (véase Fig. 22).

Hiperparámetros

Los hiperparámetros son valores que tienen que ser modificados por el programador para entrenar la red. El problema del *deep learning* es que se debe encontrar el equilibrio entre optimizar los valores para obtener el mejor resultado reduciendo todo lo posible la carga computacional y los tiempos de ejecución. Estos son los principales hiperparámetros que deben ir modificándose:

- **Batch size:** Es simplemente el número de épocas que es *forward* y *backward* propagado en cada época para que el modelo actualice los pesos.
- **Learning rate:** Este hiperparámetro trata de controlar como se van ajustando los pesos en nuestra red con respecto al gradiente de pérdida. El problema es que si este parámetro es muy pequeño el algoritmo puede tardar mucho en llegar al valor óptimo, o incluso no llegar nunca por verse atascado en un mínimo local, ya que como se puede observar en la Ecuación 3.2.5.4 los pesos se van actualizando respecto al anterior en función del *learning rate* γ . Por otro lado, si el *learning rate* es excesivamente alto sobrepasaría el valor óptimo y el algoritmo podría no llegar a converger. ("Hyperparameters in Machine/Deep Learning", 2020)

$$w_1 = w_0 - \gamma \nabla Q(w) \quad (3.2.5.4)$$

- **Número de épocas:** Es el número de veces que el modelo que se realizan los pasos de *feedforward* y *backforward propagation*. La idea es buscar el número óptimo de épocas encontrando el compromiso entre la precisión del modelo y el sobreajuste de los datos.

- Optimizador: El optimizador es la función que permite ir modificando los pesos, en base a un algoritmo determinado. En el siguiente apartado se profundiza en la explicación de diversos tipos de optimizadores.
- Función de pérdidas: Esta función $J(w)$ mide cuanto de insatisfecho se está con la predicción del modelo con respecto a una respuesta correcta. Existen distintas funciones de pérdidas como *mean square error* (MSE) o *categorical cross-entropy* y la selección va en función de varios factores, entre ellos, el objetivo de nuestro modelo. En este caso, al tener un problema multiclase con valores categóricos, la función de perdidas corresponde con *categorical cross-entropy*. Además, la base de datos de la que se dispone no es equitativa, es decir, no se tiene el mismo número de imágenes de cada etiqueta. Por lo tanto, los datos se encuentran desbalanceados, así que, mediante la librería *sklearn.utils* se calculan los pesos que deben tener las imágenes de cada etiqueta. Esto significa que durante el entrenamiento tienen un mayor peso las clases con menos muestras, con pesos mayores a 1, y las clases con más muestras tendrán pesos inferiores, menor a 1. Por lo tanto, finalmente, se tiene como función de perdidas la función *weighted – categorical cross-entropy* (véase Ecuación 3.2.5.5). (Ho & Wookey, 2019)

$$J_{wcce} = -\frac{1}{M} \sum_{k=1}^K \sum_{m=1}^M w_k \times y_m^k \times \log(h_{\theta}(x_m, k)) \quad (3.2.5.4)$$

Donde:

- M : número de muestras
- K : número de clases
- w_k : peso para la clase K
- y_m^k : etiqueta de destino para el ejemplo de entrenamiento m para la clase k
- x_m : *input* para la muestra de entrenamiento m
- h_{θ} : modelo con pesos de redes neuronales θ

Optimizadores

Se han seleccionado los optimizadores más comúnmente usados y se va a explicar el funcionamiento y diferencia entre ellos y cuál ha sido elegido para este trabajo y porque.

- RMSProp: Utiliza la magnitud de gradientes recientes para normalizar los gradientes y siempre se utiliza un promedio móvil sobre los gradientes de la raíz cuadrada media. El problema es que tiene mayores problemas de convergencia en algunos casos. (Dauphin et al., 2015)
- Adadelta: Su optimización se basa en el método estocástico de gradiente descendiente que a su vez se basa en el *learning rate* adaptativo por dimensión para abordar dos inconvenientes. El primero, es la continua decadencia de las tasas de aprendizaje a lo largo de la formación y, el segundo, la necesidad de asignar una tasa de aprendizaje global manualmente.
Adadelta es una extensión más robusta de otro optimizador – Adagrad – que adapta los distintos *learning rates* en función de una ventana móvil de actualizaciones de gradientes, en lugar de acumular los anteriores. (Ruder, 2020)
- SGD: Este método de optimización surge a partir de otros métodos de descenso de gradiente como el *batch gradient descent*, el problema de este método es que realiza cálculos redundantes para grandes conjuntos de datos, ya que vuelve a calcular gradientes para ejemplos similares antes de cada actualización de los parámetros. SGD elimina esta redundancia al realizar una actualización a la vez, por lo tanto, suele ser mucho más rápido. Además realiza actualizaciones frecuentes que hacen que la función objetivo fluctúe mucho. Finalmente, a partir de realizar pruebas con distintos optimizadores, se ha llegado a la conclusión que el que mejor hace converger los resultados es este optimizador. (Ruder, 2020)

3.2.6. Evaluación

Una vez el modelo ha sido entrenado, siempre se deben evaluar los resultados que obtiene el modelo prediciendo sobre el set de datos de test. Para ello, hacemos referencia a una serie de métricas que permiten obtener información acerca de la precisión y robustez del modelo. Estas métricas son:

- **Matriz de confusión:** La matriz de confusión es una herramienta que permite visualizar del desempeño de un algoritmo. Las columnas representan las predicciones que se realizan de cada clase, mientras que cada fila representa a la etiqueta real de las muestras. ("Matriz de confusión", 2020)

Esta matriz ayuda a evaluar la robustez del modelo, ya que no es lo mismo confundir entre dos clases contiguas, por ejemplo clase 3 y 4, que clasificar una muestra no patológica como grado 5 o viceversa.

- **Coefficiente Kappa de Cohen:** Este coeficiente es una medida estadística que ajusta el efecto del azar en la proporción de la concordancia observada para variables categóricas. Se considera una medida más robusta, ya que tiene en cuenta la clasificación que puede ser acertada por azar. Por otro lado, también se puede considerar una medida excesivamente conservadora, puesto que puede subestimar la eficiencia del modelo.

La ecuación 3.2.7.1 es la fórmula para obtener el coeficiente Kappa, donde $\Pr(a)$ es la probabilidad relativa de que distintos observadores se pongan de acuerdo y $\Pr(e)$ es la probabilidad hipotética de que se pongan de acuerdo dos observadores por azar, utilizando el set de datos. Si los observadores están completamente de acuerdo κ es igual a 1, por el contrario κ igual 0 significa que estas clasificaciones se realizan de acuerdo al azar. ("Coeficiente kappa de Cohen", 2020)

$$\kappa = \frac{\Pr(a) - \Pr(e)}{1 - \Pr(e)} \quad (3.2.7.1)$$

- **F1-Score:** F1 se utiliza para combinar las medidas de sensibilidad (recall) y valor predictivo positivo (precisión) en un solo valor. Esta medida es más práctica, ya que permite comparar el rendimiento combinado de la precisión (Ecuación

3.2.7.2) y la sensibilidad (Ecuación 3.2.7.3) en un problema multi-clase. La puntuación F1 se puede interpretar como un promedio ponderado de los valores de precisión y sensibilidad, donde 1 sería el mejor valor y 0 el peor. (Heras, 2020)

$$\textit{Precisión} = \frac{TP}{TP+FP} \quad (3.2.7.2)$$

$$\textit{Recall} = \frac{TP}{TP + FN} \quad (3.2.7.3)$$

$$F_1 = 2 \cdot \frac{\textit{precisión} * \textit{recall}}{\textit{precisión} + \textit{recall}} \quad (3.2.7.4)$$

Capítulo 4

Resultados y discusión

Tal y como se puede observar en las curvas de entrenamiento (Fig. 23-26) de las gráficas de aprendizaje de los modelos desarrollados para las resoluciones individuales, a medida que va aumentando la resolución de la imagen se pierde la capacidad del modelo para identificar las características que aporta cada una de las imágenes. Durante el entrenamiento de los modelos, se determinó el número de épocas haciendo uso de métodos de "*Early-Stopping*" que permiten extraer el modelo en la mejor época, es decir, aquella en la que las pérdidas de validación alcanzan el mínimo global. La técnica de *early stopping* permite detener el modelo en la mejor época para que no se produzca *overfitting*.

La red formada por imágenes de tamaño 512 (Fig. 23) finaliza el entrenamiento con un valor de precisión del 87,09%, mientras que la red formada por imágenes de tamaño 256 (Fig. 24) y la red formada por imágenes de tamaño 128 (Fig. 25) finalizan con un valor de 71,31% y 69,23% respectivamente. Como cabía esperar, a medida que disminuye el tamaño de la imagen, los resultados de precisión son peores, ya que el modelo está dotado con menos información contextual de las imágenes.

Por este motivo, el presente trabajo pretende abordar una perspectiva de multi-resolución que permita combinar la información local con la información del contexto, para obtener mejores resultados a nivel local.



Fig. 23. Curva de aprendizaje red resolución 512

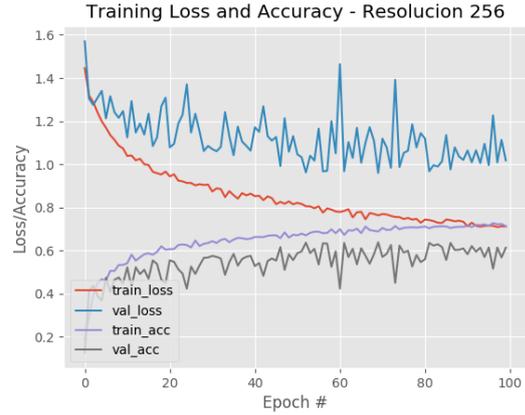


Fig. 24. Curva de aprendizaje red resolución 256

Por otro lado, el modelo multi-resolución (Fig. 26) sí que tiene una curva de aprendizaje similar al modelo de la red formada por imágenes de tamaño 512, ya no es solo que crezca o no, sino que consigue una amplia diferencia durante el entrenamiento entre los valores de pérdidas y la precisión. Tal y como se puede observar con respecto a la red de 128, los resultados son mucho más satisfactorios.

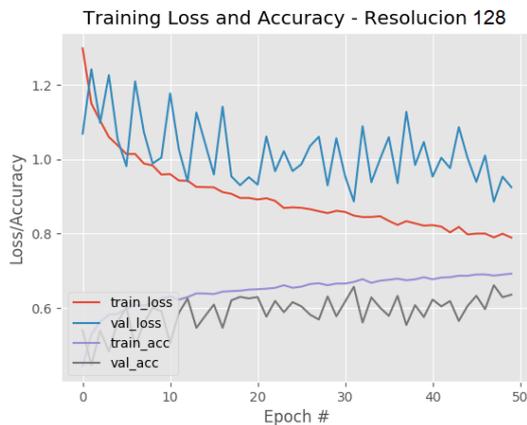


Fig. 25. Curva de aprendizaje red resolución 128

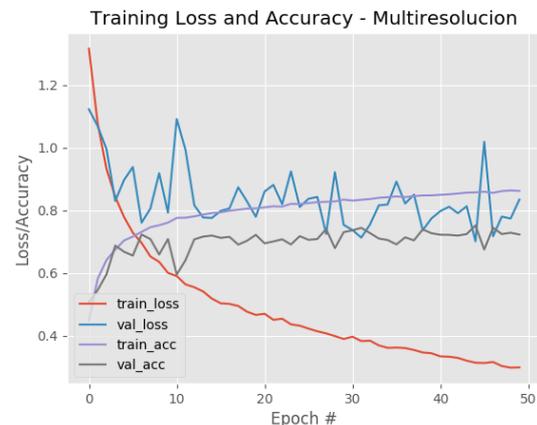


Fig. 26. Curva de aprendizaje red multi-resolución

La matriz de confusión de la red formada por imágenes de tamaño 512 que expone en Fig. 27, es bastante robusta tal y como se puede observar, pocas veces confunde las imágenes de grado 5 con imágenes no patológicas o viceversa. Pero sí que confunde en la predicción imágenes de grado 3 con imágenes de grado 4 en un 34,27% de las predicciones. También, se predice como grado 4 parches de grado 5 casi la mitad de las veces, exactamente 45,69% de las veces. Por lo tanto, se puede definir el modelo como

robusto pero no exactamente preciso, también se conocerán los datos exactos de precisión en las tablas que se mostrarán a continuación.

Por otro lado, la matriz de confusión de la red formada por imágenes de tamaño 256, Fig. 28, no es tan robusta como la anterior. Predice muy bien las imágenes no patológicas pero también predice como no patológicas otras que sí lo son. Además, predice como grado 3 imágenes de grado 4 un 32,18% de las veces. En este modelo las imágenes que mejor se encuentran predichas son aquellas que corresponden con grado 5.

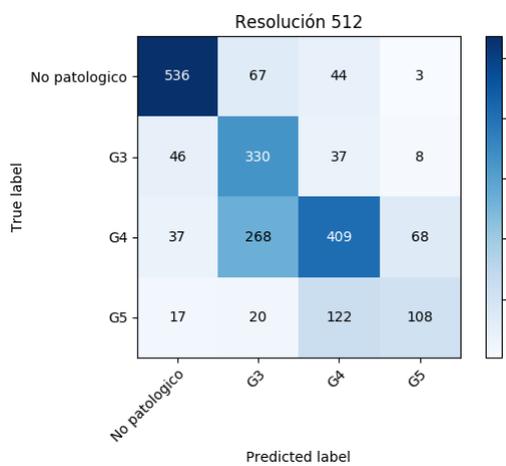


Fig. 27. Matriz de confusión red resolución 512

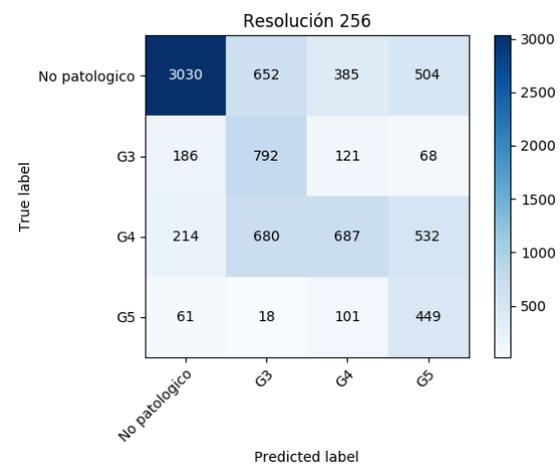


Fig. 28. Matriz confusión red resolución 256

En el caso de la matriz de confusión de la red formada por imágenes de tamaño 128 (Fig. 29) la clasificación es mucho peor que las anteriores. Tan solo predice con relativa exactitud las muestras no patológicas. También clasifican muchas muestras no patológicas como grado 4 y 5, especialmente comete este error clasificándolas como grado 4. Por otro lado, también la segunda etiqueta mejor clasificada son las muestras de grado 4 aunque también clasifica un 25,29% de las muestras como grado 5. Las etiquetas de grado 3 y 5 son las peor clasificadas en este modelo, muchos parches, el 29%, de grado 3 son clasificados como grado 4.

Con respecto a la matriz de confusión de la red multi-resolución (Fig. 30) esta situación cambia y se notan mejoras en la clasificación de las clases con respecto a cuándo solo se emplean imágenes de tamaño 128x128, pudiendo observarse como mejora la clasificación a nivel local. No predice tantas muestras no patológicas como muestras de

grado 5, lo cual aporta cierta robustez al modelo. Además, mejora la precisión en la clasificación de los parches de grado 3 considerablemente. También, mejora la clasificación de las imágenes de grado 4 aunque sigue asignando a un amplio número de imágenes no patológicas la etiqueta de grado 5 y, por lo tanto, lo que se observa es que respecto al grado 5 ha empeorado relativamente el clasificador.

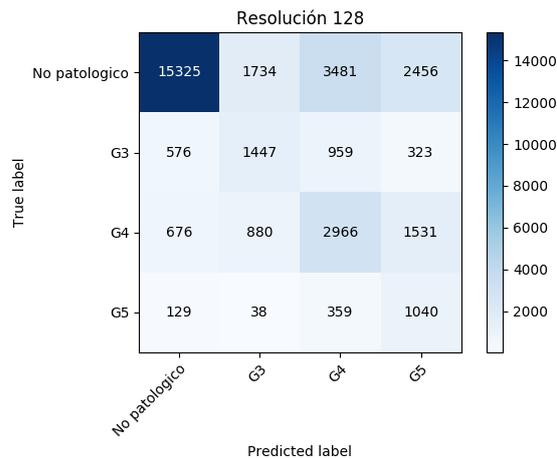


Fig. 29. Matriz confusión red resolución 128

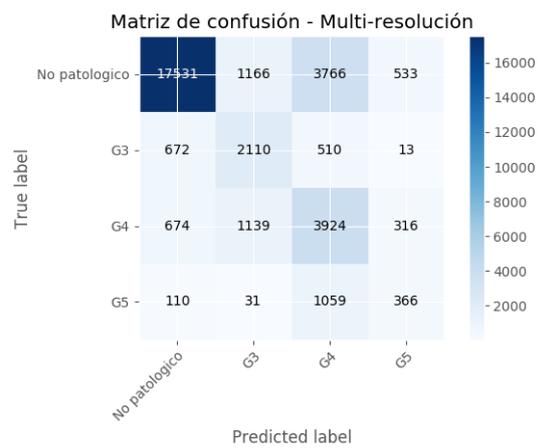


Fig. 30. Matriz confusión red multi-resolución

En las tablas se puede apreciar con más detalle lo observado en las matrices de confusión y realizar un análisis más exhaustivo. Se puede observar el informe del clasificador del modelo cuya entrada son parches de tamaño 512, y lo primero que se observa es que las muestras mejor clasificadas son las no patológicas, esto puede ser por tener un mayor número de muestras de esta clase. En general, son mejores los valores de *recall* o sensibilidad que los de precisión, ya que este hace referencia a las muestras que han sido clasificadas correctamente mientras que la precisión a cuanto de los clasificados como positivos en una clase, han sido correctamente clasificados.

	Red 512	Red 256	Red 128	Multi-resolución
No patológico	0.8428	0.8679	0.9173	0.9233
Grado 3	0.4818	0.3697	0.3530	0.4746
Grado 4	0.6683	0.5309	0.3820	0.4238
Grado 5	0.5775	0.2891	0.1944	0.2980
Avg/Total	0.6733	0.6725	0.7334	0.7616

Tabla 5. Precisión en la clasificación de las muestras a distintos niveles de resolución

	Red 512	Red 256	Red 128	Multi-resolución
No patológico	0.8246	0.6629	0.6664	0.7623
Grado 3	0.7838	0.6787	0.4378	0.6384
Grado 4	0.5230	0.3251	0.4900	0.6483
Grado 5	0.4045	0.7138	0.6641	0.2337
Avg/Total	0.6524	0.5847	0.6126	0.7055

Tabla 6. Sensibilidad en la clasificación de las muestras a distintos niveles de resolución

	Red 512	Red 256	Red 128	Multi-resolución
No patológico	0.8336	0.7517	0.7720	0.8351
Grado 3	0.5967	0.4787	0.3909	0.5444
Grado 4	0.5868	0.4033	0.4293	0.5125
Grado 5	0.4758	0.4115	0.3008	0.2620
Avg/Total	0.6505	0.6021	0.6520	0.7228

Tabla 7. F1-Score en la clasificación de las muestras a distintos niveles de resolución

El peor clasificador es el de parches de tamaño 256 tal y como se puede observar en la Tabla 7, aunque con respecto al clasificador de los parches de 128 realiza una mejor clasificación de las muestras de grado 5. El clasificador de muestras de tamaño 512 es el que mejores resultados aporta para las clases de grado 3, 4 y 5, pero, el clasificador multi-resolución es el que obtiene mejores resultados globales con respecto a todas las clases.

Especialmente en la red multi-resolución se puede observar que la clase mejor clasificada en la no patológica, por lo tanto, se puede afirmar que el modelo tiene suficiente capacidad para predecir correctamente entre una muestra patológica y una muestra no patológica.

Además, el coeficiente Kappa (Tabla 8) muestra que el modelo que presenta mayor concordancia es de tamaño de parches 512 con un coeficiente del 0.7138, seguido por el modelo multi-resolución, con un coeficiente del 0.5430. Este coeficiente lo que muestra es el grado de reproducibilidad en la clasificación más allá del azar, por lo tanto, lo que se puede extraer es que en el modelo multi-resolución se encuentra cierto grado de discordancia. Por lo tanto, se puede concluir que a nivel local el clasificador multi-resolución es mejor que el clasificador de imágenes de tamaño 128, y por lo tanto los resultados del TFM son satisfactorios.

	Kappa Score
Resolución 512	0.7138
Resolución 256	0.5063
Resolución 128	0.4265
Multi-resolución	0.5430

Tabla 8. Coeficiente Kappa de los distintos modelos

Además, lo que se puede observar del modelo multi-resolución es que hay un buen clasificador pero que tiene cierto grado de discordancia. Observando el informe del clasificador, la clase que peor predice y además con unos valores muy bajos son las clases de grado 5. Realizando un post-análisis de los datos obtenidos, se pueden plantear distintas hipótesis de porqué sucede esto.

La primera, es que las muestras de grado 5, al ser muestras necrosadas con ausencia de formación glandular, se convierten en su totalidad en estroma. Esto puede confundirse con parte de las muestras no patológicas, ya que en estas las zonas formadas por estroma son no patológicas. En el capítulo 1 se explicó que en la mayoría de los casos de cáncer de próstata (exactamente en el 95%) este se desarrolla en la zona periférica o principal la cual corresponde con la zona glandular de la próstata.

La segunda es que algunas imágenes, al ser recortadas en distintos parches de menor tamaño, pueden resultar en que estos parches contengan apenas o ninguna muestra de tejido. Esto puede generar discordancia en el clasificador ya que se están tomando algunas muestras no patológicas y algunas con apenas información histológica como no patológicas, pudiendo alterar los pesos del modelo.

Capítulo 5

Conclusión y líneas futuras

5.1. Conclusión

Haciendo referencia a los objetivos descritos en el Capítulo 2 se puede concluir que el modelo multi-resolución clasifica mejor las muestras que los modelos cuya resolución es estática. Esto se debe a que la clasificación que realizan los patólogos no solo se realiza analizando localmente la región de interés, sino también el contexto y todo lo que rodea a la muestra.

Así que tal y como se observa en el capítulo anterior, el objetivo de este TFM se cumple satisfactoriamente ya que los resultados de la red multi-resolución son mucho más precisos que los de la red formada únicamente con imágenes de tamaño 128x128. Por lo tanto, se demuestra que para realizar clasificaciones a nivel local es necesario obtener información del contexto.

El *deep learning* o aprendizaje profundo trata de aprender a identificar las muestras al igual que lo harían las redes neuronales cerebrales. Por lo tanto, la intención de este TFM es plasmar la propia ejecución de los patólogos en un modelo. Pasando una imagen de mayor resolución con su etiqueta y además imágenes de resoluciones menores que incluyan a la anterior, se consigue que el modelo entre también a partir del contexto de la imagen que ha sido clasificada.

Por lo tanto a pesar de presentar cierto grado de discordancia, se puede concluir que este modelo, el modelo multi-resolución es mejor clasificador introduciendo como entradas

distintos tamaños de imagen, que aquellos modelos entrenados con imágenes de un único tamaño.

5.2. Líneas futuras

Tal y como se ha explicado en el capítulo anterior, se pueden realizar distintas mejoras en el clasificador para obtener mejores resultados respecto al modelo. La primera es poner un umbral de muestra histológica que debe haber en la imagen para que la muestra sea aceptada dentro del modelo. Por ejemplo, se podría fijar un umbral de un 25% para descartar aquellas muestras que tengan un porcentaje de píxeles de tejido menor al umbral estipulado. Si no, se puede clasificar una muestra prácticamente vacía con una clase y, por lo tanto, no le es posible al modelo converger en la clasificación de características.

La segunda es, tal y como se ha observado en la literatura, se encuentran distintos clasificadores que también poseen un sistema de segmentación y diferencian las distintas partes del tejido, como el estroma, o la detección de lúmenes, tal y como se puede observar en las primeras fases del proyecto SICAP en García et al. (2018). Esto puede servir sobre todo para la clasificación en estadios temprano, como un grado 3, esto se puede encontrar en fases anteriores del proyecto SICAP como en García, Colomer y Naranjo (2019) o García et al. (2019). El problema se encuentra en las muestras de grado 5, donde el tejido se encuentra tan alterado que no es posible diferenciar estas estructuras.

Además, se han encontrado limitaciones a la hora de entrenar el modelo, sobretodo el modelo final, ya que al pasar una gran cantidad de parches, los tiempos computacionales han sido muy elevados y no se ha podido aumentar el número de épocas. Con un *hardware* más potente se podrían obtener mejores resultados, ya que se reduciría el tiempo computacional y podrían ser entrenados parches de mayor resolución y tamaño de imagen, sin tener que diezmar por problemas de memoria.

Capítulo 6

Bibliografía

- [1] Puente, J. y de Velasco, G., (2019). *¿Qué Es El Cáncer Y Cómo Se Desarrolla? - SEOM: Sociedad Española De Oncología Médica* © 2019. Seom.org. Consultado el 22 de Junio 2020, de: <https://seom.org/informacion-sobre-el-cancer/que-es-el-cancer-y-como-se-desarrolla>
- [2] Cifras del cáncer 2020. (2020). Consultado el 29 Junio 2020, de [https://seom.org/seomcms/images/stories/recursos/Cifras del cancer 2020.pdf](https://seom.org/seomcms/images/stories/recursos/Cifras_del_cancer_2020.pdf)
- [3] Campá, J., Mar-Barrutia, G., Extramiana, J., Arróspide, A., y Mar, J. (2016). *Supervivencia del cáncer de próstata avanzado en España según escala de Gleason, edad y estadio*. Actas Urológicas Españolas, 40(8), 499–506.
- [4] Cáncer de próstata. (2020). Consultado el 29 Junio 2020, de <https://www.aecc.es/es/todo-sobre-cancer/tipos-cancer/cancer-prostata>
- [5] Estadísticas importantes sobre el cáncer de próstata. (2020). Consultado el 1 Julio 2020, de <https://www.cancer.org/es/cancer/cancer-de-prostata/acerca/estadisticas-clave.html>
- [6] Ortega Alonzo, S., González Páramo, J., y Galván Losada, S. (2019). *PREM de Cirugía: capacitación para el ENARM*. Editorial Digital del Tecnológico de Monterrey.
- [7] Cáncer de próstata - Síntomas y signos. (2020). Consultado el 8 Agosto 2020, de <https://www.cancer.net/es/tipos-de-cancer/cancer-de-prostata/sintomas-y-signos>

-
- [8] Pruebas de detección para el cáncer de próstata. (2020). Consultado el 22 Agosto 2020, de <https://www.cancer.org/es/cancer/cancer-de-prostata/deteccion-diagnostico-clasificacion-por-etapas/pruebas-de-deteccion-para-el-cancer-de-prostata.html>
- [9] Cancer de prostata. (2020). Consultado el 28 Agosto 2020, de <https://www.slideshare.net/katherinegoavil/cancer-de-prostata-68248683>
- [10] Escala de Gleason. (2020). Consultado el 16 Julio 2020, de https://es.wikipedia.org/wiki/Escala_de_Gleason
- [11] Macías, D., Villeda, L., Díaz, H., Gomez, J., Sáez, P., & Zapata, C. et al. (2020). *Examen De Orina En Lugar De Biopsia Para Detectar Cáncer De Próstata* | Nación Farma: Salud Y Medicina Para Todos. Consultado el 29 Agosto 2020, de <https://nacionfarma.com/examen-de-orina-detectar-cancer-de-prostata/>
- [12] Histología de la próstata. (2020). Consultado el 19 Julio 2020, de <https://www.youtube.com/watch?v=5eachIx-ns0>
- [13] Histología. (2020). Consultado el 2 Agosto 2020, de <https://es.wikipedia.org/wiki/Histolog%C3%ADa>
- [14] Megías M, Molist P, Pombal MA. (2019). *Atlas de histología vegetal y animal. Técnicas histológicas*. Consultado el 22 Septiembre 2020, de <http://mmegias.webs.uvigo.es/6-tecnicas/1-introduccion.php>
- [15] Tinción de hematoxilina-eosina: características, usos, técnicas. (2020). Consultado el 23 Septiembre 2020, de <https://www.lifeder.com/tincion-hematoxilina-eosina/>
- [16] García, G.; Colomer, A.; Naranjo, V. (2019) *First-Stage Prostate Cancer Identification on Histopathological Images: Hand-Driven versus Automatic Learning*. *Entropy*, 21 (4), 356.
- [17] García J.G., Colomer A., Naranjo V., Peñaranda F., Sales M.Á. (2018) *Identification of Individual Glandular Regions Using LCWT and Machine Learning Techniques*. In: Yin H., Camacho D., Novais P., Tallón-Ballesteros A. *Intelligent Data Engineering and Automated Learning – IDEAL 2018*. IDEAL 2018. Lecture Notes in Computer Science, vol 11314. Springer, Cham.
- [18] García J. G., Colomer A., López-Mir F., Mossi J. M. and Naranjo V. (2019). *Computer Aid-System to Identify the First Stage of Prostate Cancer*

-
- Through Deep-Learning Techniques*. 27th European Signal Processing Conference (EUSIPCO), A Coruna, Spain, pp. 1-5
- [19] Doyle, S., Hwang, M., Shah, K., Madabhushi, A., Feldman, M., & Tomaszewski, J. (2007). *Automated grading of prostate cancer using architectural and textural image features*. 4th IEEE International Symposium on Biomedical Imaging: From Nano to Macro.
- [20] Gertych, A., Ing, N., Ma, Z., Fuchs, T. J., Salman, S., Mohanty, S., Knudsen, B. S. (2015). *Machine learning approaches to analyze histological images of tissues from radical prostatectomies*. *Computerized Medical Imaging and Graphics*, 46, 197–208.
- [21] Jiménez del Toro, O., Atzori, M., Otálora, S., Andersson, M., Eurén, K., Hedlund, M., & Müller, H. (2017). *Convolutional neural networks for an automatic classification of prostate tissue slides with high-grade Gleason score*. *Medical Imaging: Digital Pathology*.
- [22] Ren, J., Sadimin, E., Foran, D. J., & Qi, X. (2017). *Computer aided analysis of prostate histopathology images to support a refined Gleason grading system*. *Medical Imaging: Image Processing*.
- [23] Ing, N., Ma, Z., Li, J., Salemi, H., Arnold, C., Knudsen, B. S., & Gertych, A. *Semantic segmentation for prostate cancer grading by convolutional neural networks*. *Digital Pathology Conference 1 (June) (2018)* 46.
- [24] Esteban, Á. E., López-Pérez, M., Colomer, A., Sales, M. A., Molina, R., & Naranjo, V. (2019). *A new optical density granulometry-based descriptor for the classification of prostate histological images using shallow and deep Gaussian processes*. *Computer Methods and Programs in Biomedicine*, 178, 303–317.
- [25] Silva-Rodríguez, J., Colomer, A., Sales, M. A., Molina, R., & Naranjo, V. (2020). *Going Deeper through the Gleason Scoring Scale: An Automatic end-to-end System for Histology Prostate Grading and Cribriform Pattern Detection*. *Computer Methods and Programs in Biomedicine*, 105637.
- [26] Lucas, M., Jansen, I., Savci-Heijink, C. D., Meijer, S. L., de Boer, O. J., van Leeuwen, T. G., & Marquering, H. A. (2019). *Deep Learning for automatic Gleason pattern classification for grade group determination of prostate biopsies*. *Virchows Archiv*.

-
- [27] ¿Qué es MATLAB? – ACAPMI Blog. (2020). Consultado el 1 Octubre 2020, de <http://acapmi.com/blog/2017/09/18/que-es-matlab/>
- [28] PyCharm: uno de los mejores IDE para Python — Escuela de Python. (2020). Consultado el 2 Octubre 2020, de <https://www.escuelapython.com/pycharm-uno-de-los-mejores-ide-para-python/>
- [29] ¿Qué es Deep Learning? (2020). Consultado el 4 Octubre 2020, de https://www.sas.com/es_es/insights/analytics/deep-learning.html
- [30] Intro to data structures — pandas 1.1.1 documentation. (2020). Consultado el 14 Agosto 2020, de https://pandas.pydata.org/pandas-docs/stable/user_guide/dsintro.html
- [31] Brownlee, J. (2020). Why One-Hot Encode Data in Machine Learning? Consultado el 13 Agosto 2020, de <https://machinelearningmastery.com/why-one-hot-encode-data-in-machine-learning/>
- [32] Datos de entrenamiento vs datos de test - Think Big Empresas. (2020). Consultado el 6 Octubre 2020, de <https://empresas.blogthinkbig.com/datos-entrenamiento-vs-datos-de-test/>
- [33] Training, validation, and test sets. (2020). Consultado el 4 Octubre 2020, de https://en.wikipedia.org/wiki/Training,_validation,_and_test_sets
- [34] Brownlee, J. (2020). How to Configure Image Data Augmentation in Keras. Consultado el 29 Septiembre 2020, de <https://machinelearningmastery.com/how-to-configure-image-data-augmentation-when-training-deep-learning-neural-networks/>
- [35] Combinación de capas para redes neuronales convolucionales - sitiobigdata.com. (2020). Consultado el 13 Octubre 2020, de <https://sitiobigdata.com/2019/05/01/combinacion-capas-redes-neuronales-convolucionales/>
- [36] Rosebrock, A. (2020). Keras Conv2D and Convolutional Layers - PyImageSearch. Consultado el 15 Octubre 2020, de <https://www.pyimagesearch.com/2018/12/31/keras-conv2d-and-convolutional-layers/>
- [37] Keras documentation: MaxPooling2D layer. (2020). Consultado el 15 Octubre 2020, de https://keras.io/api/layers/pooling_layers/max_pooling2d/

-
- [38] Max-pooling / Pooling - Computer Science Wiki. (2020 Consultado el 15 Octubre 2020, de <https://computersciencewiki.org/index.php/Max-pooling / Pooling>
- [39] Introduction to Capsule Networks | Paperspace Blog. (2020). Consultado el 20 Octubre 2020, de <https://blog.paperspace.com/capsule-networks/>
- [40] Keras documentation: Concatenate layer. (2020). Consultado el 24 Octubre 2020, de https://keras.io/api/layers/merging_layers/concatenate/
- [41] VGG16 - Convolutional Network for Classification and Detection. (2020). Consultado el 1 Octubre 2020, de <https://neurohive.io/en/popular-networks/vgg16/>
- [42] POTENCIANDO CONVOLUCIONES CON TRANSFORMACIONES AFINES. (2020). Consultado el 1 Octubre 2020, de <https://www.datahack.es/convoluciones-transformaciones-afines-2/>
- [43] Hyperparameters in Machine/Deep Learning. (2020). Consultado el 26 Octubre 2020, de <https://medium.com/@jorgesleonel/hyperparameters-in-machine-deep-learning-ca69ad10b981>
- [44] Ho, Yaoshiang & Wookey, Samuel. (2019). *The Real-World-Weight Cross-Entropy Loss Function: Modeling the Costs of Mislabeling*. IEEE Access.
- [45] Dauphin, Yann & Vries, Harm & Chung, Junyoung & Bengio, Y. (2015) *RMSProp and equilibrated adaptive learning rates for non-convex optimization*. arXiv. 35.
- [46] Ruder, S. (2020). An overview of gradient descent optimization algorithms. Consultado el 26 Octubre 2020, de <https://ruder.io/optimizing-gradient-descent/>
- [47] Matriz de confusión. (2020). Consultado el 23 Octubre 2020, de https://es.wikipedia.org/wiki/Matriz_de_confusi%C3%B3n#:~:text=En%20el%20campo%20de%20la,se%20emplea%20en%20aprendizaje%20supervisado.
- [48] Coeficiente kappa de Cohen. (2020). Consultado el 23 Octubre 2020, de https://es.wikipedia.org/wiki/Coeficiente_kappa_de_Cohen#cite_note-1
- [49] Heras, J. (2020). Precision, Recall, F1, Accuracy en clasificación - IArtificial.net. Consultado el 23 Octubre 2020, de <https://www.iartificial.net/precision-recall-f1-accuracy-en-clasificacion/#F1>

Parte II

Presupuesto

Índice del presupuesto

1. Presupuestos parciales	1
1.1. Coste de mano de obra	1
1.2. Coste de materiales	2
1.3. Coste de software y hardware	2
2. Presupuesto total	4

Presupuesto

El objetivo de este capítulo es realizar una valoración económica del gasto del proyecto para valorar cuánto costaría el diseño y desarrollo de un sistema automático para la gradación del cáncer de próstata aplicando técnicas multi-resolución con algoritmos de *deep learning* en Whole-Slide Images. Primero, se cifra el coste de los presupuestos parciales y, posteriormente, con el recuento de estos se realiza el presupuesto total de proyecto.

1. Presupuestos parciales

En este apartado se irán realizando los presupuestos parciales de los distintos aspectos que entraña el proyecto. Para ello, se segmenta el presupuesto en tres presupuestos parciales: el coste de mano de obra, el coste de los materiales empleados y el coste de *software* y *hardware*.

1.1. Coste de mano de obra

Esta sección hace referencia al coste de recursos humanos del TFM, esta sección tiene en cuenta las horas de dedicación de cada uno de los participantes junto con la remuneración que se contabiliza en unidades por hora. Por tanto, debe considerarse la contribución de: Dña. Valery Naranjo Ornedo (como directora del proyecto), D. Julio José Silva Rodríguez y D. José Gabriel García Pardo (como cotutores) y Dña. Elena Nieto González (como estudiante y autora del proyecto).

Concepto	Uds.	Cantidad (Uds.)	Precio unitario (€/Uds.)	Total (€)
Tutora	h.	16	42,00 €	672,00 €
Cotutor 1	h.	36	17,20 €	619,20 €
Cotutor 2	h.	36	17,20 €	619,20 €
Estudiante	h.	600	13,80 €	8.280,00 €
			Total	10.190,40 €

Tabla 9. Coste mano de obra

1.2. Coste de materiales

El coste de los materiales se corresponde con la cantidad de biopsias realizadas que será igual al número de pacientes y, por otro lado, la cantidad de muestras que se obtienen y deben ser digitalizadas para su posterior análisis.

Concepto	Uds.	Cantidad (Uds.)	Precio Unitario (€/Uds.)	Total (€)
Biopsias	u	111	600,00 €	66.600,00 €
Muestras	u	181	10,00 €	1.810,00 €
			Total	68.410,00 €

Tabla 10. Coste de materiales

1.3. Coste de software y hardware

Este apartado corresponde con los costes de *software* y *hardware* necesarios para llevar a cabo el proyecto. Se han necesitado las licencias de Office 365, MATLAB[®] v.R2020a y Pycharm v.2020 1.3, estas licencias al ser proporcionadas por la Universidad Politécnica de Valencia el coste ha sido nulo. Se hace uso de las licencias proporcionadas por la UPV ya que este trabajo es experimental y no podría aplicarse a la práctica clínica ya que en estos momentos no se encuentra validado.

En segundo lugar, se ha llevado el entrenamiento de varias redes neuronales, como los tiempos han sido muy variados, se va a estimar que se han empleado 120 horas del servidor de CVBLab para llevar a cabo esta labor.

Por último, para realizar los cálculos de amortización del hardware, al no haber sido adquirido únicamente para la realización de este proyecto, se ha de tener en cuenta el tiempo de amortización durante la duración del proyecto. Si el ordenador HP Notebook 15-ac006ns tiene una vida útil de 5 años, lo que equivale a 72 meses, y su coste inicial fue de 669,00 €, el precio unitario es de 9,29 €.

Concepto	Uds.	Cantidad (Uds.)	Precio Unitario (€/Uds.)	Periodo de amortización (meses)	Intervalo amortizado (meses)	Total (€)
Licencia Office 365	u	1	- €	12	6	- €
Licencia MATLAB[®]	u	1	- €	12	6	- €
Licencia Pycharm IDE	u	1	- €	12	6	- €
Servidor de CVBLab HP Notebook 15-ac006ns	h	120	0,04 €	indefinido	6	4,80 €
	u	1	9,29 €	72	6	55,75 €
Total						60,55 €

Tabla 11. Coste de software y hardware

2. Presupuesto total

Para el presupuesto total es necesario calcularlo en base a la suma de todos los presupuestos parciales generados anteriormente. Además, se añadirá el porcentaje de gastos generales (13%) y el porcentaje asociado al beneficio industrial (6%). Por último, se le añadirá al precio total bruto el impuesto del IVA (21%), obteniendo finalmente el presupuesto para la ejecución de este Trabajo Fin de Master.

CAPÍTULOS	IMPORTE (€)
1. Coste de mano de obra	10.190,40 €
2. Coste de materiales	68.410,00 €
3. Coste de software y hardware	60,55 €
PRESUPUESTO DE EJECUCIÓN DE MATERIAL	78.660,95 €
13% de gastos generales	10.225,92 €
6% de beneficio industrial	4.719,66 €
PRESUPUESTO DE EJECUCIÓN POR CONTRATA	93.606,53 €
21% de IVA	19.657,37 €
PRESUPUESTO TOTAL	113.263,90 €

Tabla 12. Coste total de la ejecución del proyecto

El presupuesto total de Trabajo Fin de Master es de ciento trece mil doscientos sesenta y tres euros con noventa céntimos.