



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



ESCUELA TÉCNICA
SUPERIOR INGENIEROS
INDUSTRIALES VALENCIA

TRABAJO FIN DE GRADO EN INGENIERÍA BIOMÉDICA

DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA PARA LA SEGMENTACIÓN NUCLEAR EN IMÁGENES HISTOLÓGICAS MULTI-ÓRGANO MEDIANTE UNA RED NEURONAL PROFUNDA ENCODER- DECODER

AUTOR: GALIANA BORDERA, ADRIÁN

TUTOR: NARANJO ORNEDO, VALERIANA

Selección COLOMER GRANERO, ADRIÁN

Curso Académico: 2018-19

AGRADECIMIENTOS

En primer lugar, quiero agradecer este proyecto a mis tutores. A Valery, por su espontaneidad, por ser persona antes que profesora. Por ese “pero si hace 2 días ibas perdidísimo, ¿iqué has hecho!? Me encanta ver que ya lo vas pillando todo” que me dijo cuando hice con ella y Adri, mi cotutor, el curso de *Deep learning*. Y a Adri, por todos esos skypes empezados por: “Hola Adri! // Hola Adri!” y por confiar tanto en mí y tratarme de igual a igual, haciéndome ver que podía hacerlo todo sin ayudarme en cada pasito.

A mi familia que, aunque todavía no entiendan bien qué es lo que se supone que hago, siempre está ahí, apoyándose.

A mi segunda familia, mis chicos, la **KAME HOUSE**, pues no hay mejores compañeros de piso ni de vida.

A Pedromi, que se ha tragado mil y una veces mis rollos del TFG y cómo se me iban ocurriendo nuevas ideas que probar en la red neuronal. Y sin su constante apoyo, nada hubiese sido posible.

A Mari, la chica que me hizo engancharme a esto de programar y, también, a ella. No lo olvides, tenemos el **gen script**.

Por supuesto, no puedo olvidarme de la escuela de verano JOCUS, en la que llevo siendo monitor desde hace 7 años y en los que tengo amigos a los que confiaría mi vida. Y a todos los niños que también piensan que estoy loco por todo lo que les cuento que se pude llegar a alcanzar con la carrera que he estudiado. Y me motiva muchísimo a seguir adelante.

Y, aunque este año nos ha dejado, a mi Rusita, esa perrita ha sido muy importante en mi vida y no concibo noches enteras de estudio sin ella cerquita.

RESUMEN

La segmentación nuclear en imágenes histológicas puede permitir la extracción de características de alta calidad para morfometría nuclear y demás análisis en patología computacional. Un algoritmo que segmente con precisión los núcleos en bases de datos que abarquen una gran variedad de pacientes, órganos y estados de cáncer puede contribuir significativamente al desarrollo de software de investigación clínica y médica. Esto es debido a que una vez segmentados de forma precisa los núcleos, se pueden extraer características morfométricas y de apariencia nucleares, como la densidad, la relación núcleo-citoplasma, el tamaño promedio y el pleomorfismo. Dichos rasgos pueden emplearse tanto en la identificación de los distintos grados de cáncer como en la predicción de la efectividad de un determinado tratamiento.

Palabras clave: cáncer, núcleos, imagen histológica, red neuronal, inteligencia artificial, *deep learning*, convolución, UNet, ResNet, AJI

RESUM

La segmentació nuclear en imatges histològiques permet extreure característiques d'alta qualitat per morfometria nuclear i altres anàlisis en patologia computacional. Un algoritme que segmente amb precisió els nuclis de les imatges de la nostra base de dades, feta a partir d'una gran varietat de pacients, òrgans i estats de càncer, pot contribuir significativament al desenvolupament de programari de recerca clínica i mèdica. Això és degut al fet que la segmentació precisa dels nuclis permet extreure característiques morfomètriques i d'aparença nuclears, com la densitat, la relació nucli-citoplasma, la grandària mitjana i el pleomorfisme. Aquests trets poden emprar-se tant en la identificació dels diferents graus de càncer com en la predicció de l'eficàcia d'un determinat tractament.

Paraules clau: càncer, nuclis, imatge histològica, xarxa neuronal, intel·ligència artificial, *deep learning*, convolució, UNet, ResNet, AJI

ABSTRACT

Nuclear segmentation in histological images can allow the extraction of high-quality characteristics for nuclear morphometry and other analyses in computational pathology. An algorithm that precisely segments the nuclei of the images in our database, made from a wide variety of patients, organs and cancer states, can contribute significantly to the development of clinical and medical research software. This is because once the nuclei are precisely segmented, nuclear morphometric and appearance characteristics can be extracted, such as density, nucleus-cytoplasm ratio, average size and pleomorphism. These traits can be used both in the identification of the different degrees of cancer and in the effectiveness's prediction of a certain treatment.

Keywords: cancer, nuclei, histological image, neural network, artificial intelligence, *deep learning*, convolution, UNet, ResNet, AI

ÍNDICE GENERAL

PARTE I. MEMORIA

PARTE II. PRESUPUESTO

Parte I

Memoria

ÍNDICE DE LA MEMORIA

CAPÍTULO 1	5
1. INTRODUCCIÓN	5
1.1. EL CÁNCER	6
1.2. HISTOLOGÍA	8
1.2.1. Preparación de la muestra histológica	8
1.2.2. Tinción Hematoxilina & Eosina	15
1.2.3. Análisis de las imágenes	15
1.3. MACHINE LEARNING, DEEP LEARNING E INTELIGENCIA ARTIFICIAL	16
1.3.1. Inteligencia Artificial	16
1.3.2. Machine learning	17
1.3.3. Deep learning	18
1.4. MARCO DEL PROYECTO	19
1.4.1. SICAP	19
1.4.2. Challenge MoNuSeg	19
1.5. ESTADO DEL ARTE	19
CAPÍTULO 2	21
2. OBJETIVOS	21
CAPÍTULO 3	23
3. MATERIAL Y MÉTODOS	23
3.1. MATERIAL	23
3.1.1. Base de datos	23
3.1.2. Equipo y entorno de programación	25
3.2. METODOLOGÍA	25
3.2.1. Extracción de regiones de interés y etiquetado	25
3.2.2. Obtención de las máscaras ternarias y de instancias	27
3.2.3. Normalización de color	28
3.2.4. Redes neuronales	32
3.2.5. Arquitectura Red neuronal convolucional	41
3.2.6. Arquitectura autoencoder (encoder – decoder)	46
3.2.7. UNet	47
3.2.8. Desarrollo de la red UNet	49
3.2.9. Postprocesado	50
CAPÍTULO 4	53
4. RESULTADOS	53
4.1. RESULTADOS CUALITATIVOS SOBRE LAS IMÁGENES DE TEST	53
4.2. RESULTADOS CUANTITATIVOS Y COMPARATIVA	58
4.2.1. Aggergated Jaccard Index	58

4.2.2.	<i>Comparativa de las dos redes propuestas</i>	59
4.2.3.	<i>Comparativa de resultados con MoNuSeg</i>	60
CAPÍTULO 5		63
5.	CONCLUSIONES Y LÍNEAS FUTURAS	63
5.1.	CONCLUSIONES	63
5.2.	LÍNEAS FUTURAS	64
BIBLIOGRAFÍA		67

1. INTRODUCCIÓN

El siguiente proyecto pretende abordar cómo la inteligencia artificial se puede aplicar a la histología y ver los alcances a los que se puede llegar en la ayuda al diagnóstico histopatológico.

La histología ha sido una herramienta importante en el diagnóstico y pronóstico del cáncer durante más de un siglo. Los patólogos anatómicos evalúan las características histológicas, como la atipia nuclear, la actividad mitótica, la densidad celular y la arquitectura tisular, incorporando detalles citológicos y patrones de orden superior para clasificar las lesiones. Aunque el pronóstico se basa cada vez más en biomarcadores genómicos que miden alteraciones genéticas, expresión genética y modificaciones epigenéticas, la histología sigue siendo una herramienta importante para predecir el curso futuro de la enfermedad de un paciente.

Por su parte, los sensores fotográficos han ido avanzando hasta el punto de poder obtener imágenes digitalizadas de muestras micrométricas con una magnífica resolución. Estas imágenes se pueden tratar con el fin de extraer características y patrones que los patólogos analizan en sus laboratorios. Las muestras histológicas digitalizadas es el punto de inicio de este trabajo de fin de grado.

A continuación, se va a detallar a grandes rasgos, qué es el cáncer, la histología y las tinciones histológicas, haciendo hincapié en la hematoxilina & eosina, dado que es la más ampliamente usada y con la que se ha tratado en el proyecto. Además de una breve introducción a la inteligencia artificial.

1.1. El cáncer

a) Definición del cáncer

La Organización Mundial de la Salud (OMS) define el cáncer como “*proceso de crecimiento y diseminación incontrolados de células*” [1]. Es decir, es un crecimiento anormal de células que se caracteriza por alterar la expresión de múltiples genes. Esto conduce a la desregulación del programa celular normal para su división y diferenciación. Un desequilibrio en la replicación y la muerte celular favorece el crecimiento de una población de células tumorales. En última instancia, esta población de células es capaz de invadir otros tejidos y hacer metástasis en localizaciones alejadas al foco de crecimiento [2].

b) Descripción del cáncer

Las principales características de un tumor maligno son su capacidad de propagarse a los ganglios linfáticos y metastatizar a órganos distantes en el cuerpo, cosa que no ocurre en los tumores benignos [1], [2].

Clínicamente, el cáncer se asemeja a muchas enfermedades diferentes con diversas características fenotípicas. A medida que progresa el crecimiento cancerígeno, la deriva genética en la población celular produce heterogeneidad, es decir, el genoma va degenerando. Esto complica su reparación y da lugar a poblaciones cada vez más diferentes a las progenitoras. Concretamente, estos cambios afectan a características como la antigenicidad celular, invasividad, potencial metastático, tasa de proliferación celular, estado de diferenciación y respuesta a agentes quimioterapéuticos.

A nivel molecular, todos los cánceres comparten algunos rasgos, lo que sugiere que las lesiones bioquímicas definitivas que conducen a la transformación y progresión malignas pueden producirse mediante un patrón común, pero no idéntico en las alteraciones de la lectura del gen [2].

En general, los tumores malignos causan una morbilidad significativa y serán letales para el huésped si no se tratan. Las excepciones a esto parecen ser los cánceres latentes e indolentes que pueden permanecer clínicamente indetectables (o *in situ*), lo que permite al huésped tener una esperanza de vida estándar [3].

c) Datos estadísticos

En 2017, se estima que 9.6 millones de personas murieron a causa de varias formas de cáncer. El Instituto de Métricas y Evaluación de la Salud (IHME) estableció unos márgenes de error relativamente pequeños en torno a esta cifra global. Las estimaciones inferiores y superiores se extienden de 9.2 a 9.7 millones [4].

El cáncer es una de las principales causas de morbilidad y mortalidad en el mundo. En general, la prevalencia de cáncer ha aumentado. En Europa se detectan más de 3.7 millones de casos de cáncer nuevos y causa 1.9 millones de muertes cada año, y aunque más del 40% de las muertes por cáncer se pueden prevenir, el cáncer representa el 20% de las muertes en Europa.

En escala global, el cáncer representó 8.2 millones de muertes (alrededor del 13% del total) en 2012. En *Figura 1.1* se puede observar la cantidad de personas diagnosticadas con cáncer en 2017 por países.

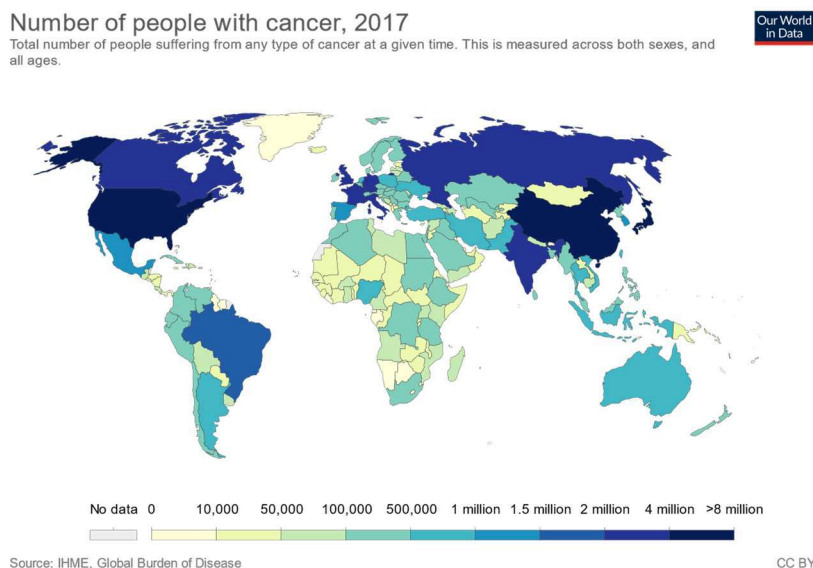


Figura 1.1. Número de personas con cáncer [4]

d) Ocurrencia

A pesar de compartir algunos rasgos unos con otros tipos, el cáncer es una enfermedad con gran variedad a nivel tisular, lo que supone un gran desafío para su diagnóstico específico y la eficacia del tratamiento. En los hombres, la mayor ocurrencia es el cáncer de próstata, seguido del de los pulmones y los bronquios, el colon y el recto y la vejiga urinaria. En las mujeres, la prevalencia de cáncer es más alta en los senos, pulmones y bronquios, colon y recto, cuerpo uterino y tiroides. En la *Figura 1.2* se puede ver la cantidad de enfermos de cada tipo de cáncer en el 2017.

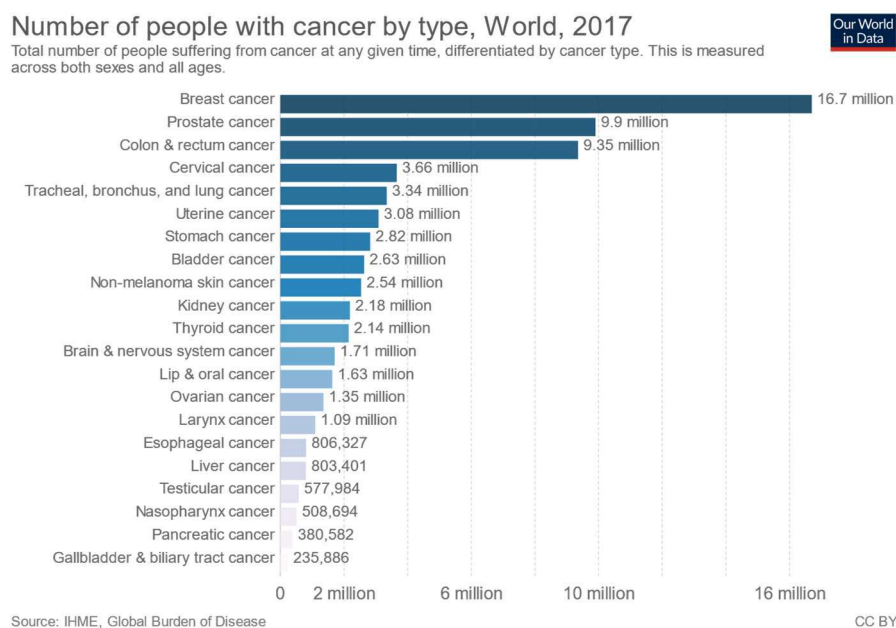


Figura 1.2. Número de personas por tipo de cáncer en el mundo, 2017 [4]

Estos datos indican que el cáncer de próstata y de mama constituyen una parte importante del cáncer en hombres y mujeres, respectivamente. En niños, los tipos más altos de enfermedad por cáncer son el cáncer de la sangre y los cánceres relacionados con el cerebro y los ganglios linfáticos.

En resumen, el cáncer se produce por una serie de mutaciones sucesivas en los genes, las cuales cambian las funciones celulares. Los compuestos químicos tienen un papel importante en la formación de mutaciones genéticas y células cancerosas. Además, hábitos como fumar involucran varios compuestos químicos cancerígenos que acaban en los pulmones y pueden ayudar a que se produzcan estas mutaciones genéticas [4].

1.2. Histología

La histología es el estudio de la microanatomía de células, tejidos y órganos a través de un microscopio.

Hay muchas razones para examinar las células y tejidos humanos bajo el microscopio. La investigación médica y biológica se basa en el conocimiento de la estructura y funciones normales de las células, tejidos, órganos y estructuras que componen el cuerpo humano. En un sujeto sano las células y otros elementos del tejido están dispuestos en patrones regulares reconocibles. Los cambios inducidos por una amplia gama de influencias químicas y físicas se reflejan por alteraciones en la estructura a nivel microscópico. Muchas enfermedades se caracterizan por anomalías estructurales y químicas típicas que difieren del estado normal. Identificar estos cambios y vincularlos con enfermedades particulares es la base de la histopatología y la citopatología [5].

Por tanto, saber cómo se ve un tejido y cómo funciona en condiciones normales es importante para poder reconocer el cáncer y nos puede ayudar a determinar cómo tratarlo y si el tratamiento ha funcionado.

1.2.1. Preparación de la muestra histológica

En primer lugar, debemos tener en cuenta que existen muchas formas diferentes de microscopía. La que se emplea con mayor frecuencia es la microscopía de "campo claro", donde la muestra se ilumina con un haz de luz que la atraviesa (a diferencia de un haz de electrones como en la microscopía electrónica). Los requisitos generales para que un espécimen sea examinado con éxito usando microscopía de campo claro son:

- Las células y el resto de los elementos que se encuentren en la muestra se deben conservar en un estado realista, a este proceso se llama "fijación".
- La muestra no puede ser opaca, ya que la luz tiene que pasar a través de ella.
- El espécimen debe ser delgado y plano, de modo que solo esté presente una sola capa de células.
- La mayoría de las células son incoloras, por lo que las secciones histológicas deben teñirse de alguna manera para que las células sean visibles.

Debido a los requisitos de microscopía, las opciones para preparar muestras se limitan a:

- **Monturas enteras**, donde todo el organismo o estructura es lo suficientemente pequeño o delgado como para colocarlo directamente en un portaobjetos de microscopio (por ejemplo, un pequeño organismo unicelular o multicelular o una membrana que se puede estirar finamente sobre un portaobjetos).
- **Preparaciones de "aplastar"**, donde las células se aplastan intencionalmente en un portaobjetos para revelar su contenido.
- **Frotis**, donde el espécimen consiste en células suspendidas en un líquido (por ejemplo, sangre, semen, líquido cefalorraquídeo o un cultivo de microorganismos), o donde las células individuales se han raspado, cepillado o aspirado (succionado) desde una superficie o desde el interior de un órgano (citología exfoliativa). Los frotis son la base de la conocida "prueba de Papanicolaou" que se utiliza para detectar el cáncer de cuello uterino en las mujeres.
- **Secciones**, donde las muestras son compatibles de alguna manera para que se puedan cortar rodajas muy finas, montarlas en portaobjetos y teñirlas. Las secciones se preparan usando un instrumento llamado "microtomo".

Los frotis proporcionan detalles sobre las células individuales y los números relativos de células, pero se pierden las relaciones estructurales. Por su parte, las secciones conservan las relaciones estructurales entre las células individuales y los componentes extracelulares. El examen microscópico de las secciones por un patólogo forma la piedra angular del diagnóstico de cáncer.

Preparación de la sección

La mayor parte del tejido fresco es muy delicado, se distorsiona y daña fácilmente y, por lo tanto, es imposible preparar secciones delgadas (rebanadas) a menos que se sostenga de alguna manera mientras se corta.

En general, hay dos tipos de estrategias a seguir:

1. *Congelación*: El tejido puede congelarse y mantenerse congelado mientras las secciones se cortan usando un microtomo de criostato (un microtomo en una cámara de congelación). Estas muestras se llaman "secciones congeladas". Tienen la ventaja de poderse preparar muy rápidamente y, por lo tanto, se usan cuando se requiere un diagnóstico intraoperatorio para guiar un procedimiento quirúrgico o cuando se debe evitar cualquier tipo de interferencia con la composición química de las células (como en algunas investigaciones histoquímicas).
2. *Formaldehído y parafina*: Alternativamente, las muestras se pueden infiltrar con un agente líquido que posteriormente se puede convertir en un sólido que tenga las propiedades físicas apropiadas y permita cortar secciones delgadas. Se pueden usar varios agentes para infiltrar y soportar muestras que incluyen resinas epoxi y metacrilato, aunque las muestras histológicas con base de cera de parafina son las más populares para la microscopía óptica. Con esto se consigue las llamadas "secciones de parafina" que se preparan, generalmente, con un microtomo rotativo. En todos los laboratorios de histopatología, las secciones de parafina se preparan rutinariamente a partir de casi todas las muestras y se utilizan en el diagnóstico.

Los siguientes párrafos describen los principales pasos en la preparación de secciones de parafina. Estos procesos dictan el diseño y el flujo de trabajo en laboratorios de histopatología donde se manejan cientos de muestras todos los días.

i. Recepción de muestras

Las muestras recibidas para el examen histológico pueden provenir de varias fuentes diferentes. Van desde especímenes muy grandes u órganos completos hasta pequeños fragmentos de tejido. Por ejemplo,

- Muestras de escisión (biopsias quirúrgicas), donde se extraen órganos enteros o áreas afectadas durante la operación.
- Muestras de biopsia incisional, donde se extrae tejido para el diagnóstico desde un área afectada.
- Biopsias por punción, donde se utilizan punzones para extraer un pequeño trozo de tejido sospechoso para su examen (a menudo de la piel).
- Biopsias de afeitado, donde pequeños fragmentos de tejido se "afeitan" de una superficie (generalmente piel).
- Curetas, donde el tejido se extrae en pequeños trozos del revestimiento del útero o el cuello uterino.
- Biopsias centrales, donde se extrae una pequeña muestra de tejido con una aguja especial a veces a través de la piel (por vía percutánea).

Las muestras generalmente se reciben fijadas (en un conservante), pero a veces llegan frescas y deben repararse de inmediato, como se ve en la Figura 1.3. Antes de que un laboratorio acepte las muestras, se verificará cuidadosamente la identificación (etiquetado) y la documentación adjunta, se registrarán todos los detalles y se iniciará el "seguimiento de muestras". Es vital que las muestras de pacientes o de investigación se identifiquen adecuadamente y se minimice el riesgo de inexactitudes.



Figura 1.3. Una muestra fresca no fijada después de la extracción quirúrgica [5].

ii. Fijación

La fijación es un paso crucial en la preparación de muestras para examen microscópico. Su objetivo es prevenir la descomposición y preservar las células y los tejidos en un estado "realista". Lo hace deteniendo la actividad enzimática, matando microorganismos y endureciendo la muestra mientras mantiene suficiente estructura molecular para permitir la aplicación de métodos de tinción apropiados (incluidos los que involucran reacciones antígeno-anticuerpo y aquellos que dependen de preservar el ADN y el ARN). Cuanto antes se inicie la fijación después de la separación de una muestra de su suministro de sangre, mejor será el resultado. El agente de fijación más popular es el formaldehído, generalmente en forma de una solución tamponada con fosfato (a menudo denominada "formalina"), que es la que se observa en la Figura 1.4. Idealmente, las muestras deben fijarse por inmersión en formalina durante seis a doce horas antes de ser procesadas.



Figura 1.4. Espécimen quirúrgico que se fija en formalina y está listo para la extracción [5]

iii. Selección del corte

La realización de los cortes, Figura 1.5, supone un examen cuidadoso y una descripción del espécimen que incluirá la apariencia, el número de piezas y sus dimensiones. Los especímenes más grandes pueden requerir una disección adicional para producir piezas representativas de las áreas apropiadas. Por ejemplo, se pueden tomar múltiples muestras de los márgenes de escisión de un tumor para asegurar que el tumor se haya eliminado por completo. En el caso de muestras pequeñas, se puede procesar toda la muestra. Los tejidos seleccionados para el procesamiento se colocarán en casetes (pequeñas cestas perforadas) y los *batches* se cargarán en un procesador de tejidos para procesarlos en cera.



Figura 1.5. espécimen quirúrgico de estómago fijado en formalina [5]

iv. Procesamiento

Cuando se procesan grandes *batches* de muestras para la preparación de la sección de parafina, se utilizan instrumentos automatizados llamados "procesadores de tejidos" (Figura 1.6). Estos instrumentos permiten que las muestras se infiltran con una secuencia de diferentes solventes que terminan en parafina fundida. Para comenzar, las muestras se encuentran en un entorno acuoso (a base de agua) y deben pasar a través de múltiples cambios de solventes deshidratantes y limpiadores (típicamente etanol y xileno) antes de que puedan colocarse en cera fundida (que es hidrófoba e inmisible con agua). La duración y los detalles del paso del "programa de procesamiento" elegido para un *batch* particular de muestras dependerán de la naturaleza y el tamaño de las muestras. Los tiempos pueden ser tan cortos como una hora para muestras pequeñas o tan largos como doce horas o más para muestras grandes. En muchos laboratorios, la mayor parte del procesamiento se lleva a cabo durante la noche. En la actualidad, existe una presión considerable sobre los laboratorios para que utilicen procesadores capaces de procesar rápidamente en un esfuerzo por mejorar el flujo de trabajo y reducir los tiempos de respuesta.



Figura 1.6. Procesador de tejidos que se carga con una canasta de casetes que contienen muestras de tejidos para su procesamiento [5]

v. Ensamblado

Después del procesamiento, las muestras se colocan en un centro de inclusión donde se retiran de sus casetes y se colocan en moldes rellenos de cera. En esta etapa, las muestras se orientan cuidadosamente, como se puede ver en la Figura 1.7, esto determinará el plano a través del cual se cortará la sección y, en última instancia, puede decidir si un área anormal será visible bajo el microscopio. El casete en el que se ha procesado el tejido lleva los detalles de identificación de la muestra y ahora se coloca en la parte superior del molde y se une mediante la adición de cera adicional. Ahora se permite que el "bloque" de la muestra se solidifique en una superficie fría y cuando se establece el molde se retira. El casete, ahora lleno de cera y formando parte del bloque, Figura 1.8, proporciona una base estable para sujetar el microtomo. El bloque que contiene la muestra ya está listo para cortar secciones.



Figura 1.7. Colocación y orientación en un molde de inclusión en la placa caliente de un centro de inclusión [5]



Figura 1.8. Muestra procesada, incrustada y lista para la preparación de la sección utilizando un microtomo [5]

vi. Seccionamiento

Las secciones se cortan en un instrumento de precisión llamado "microtomo" con cuchillas de acero extremadamente finas, podemos ver el microtomo y los cortes de secciones que genera en la Figura 1.9. Las secciones de parafina generalmente se cortan con un grosor de 3 a 5 μm , asegurando que solo una sola capa de células forme la sección (un glóbulo rojo tiene un diámetro de aproximadamente 7 μm). Una de las ventajas de la cera de parafina como agente de inclusión es que a medida que se cortan las secciones se pegarán de borde a borde, formando una "cinta" de secciones, lo cual facilita el manejo.

Las secciones ahora se colocan en agua tibia, Figura 1.10, esto se conoce como un baño de flotación que sirve para aplanarlas y luego se recogen en un portaobjetos de microscopio. Después de un secado completo, están listos para la tinción.

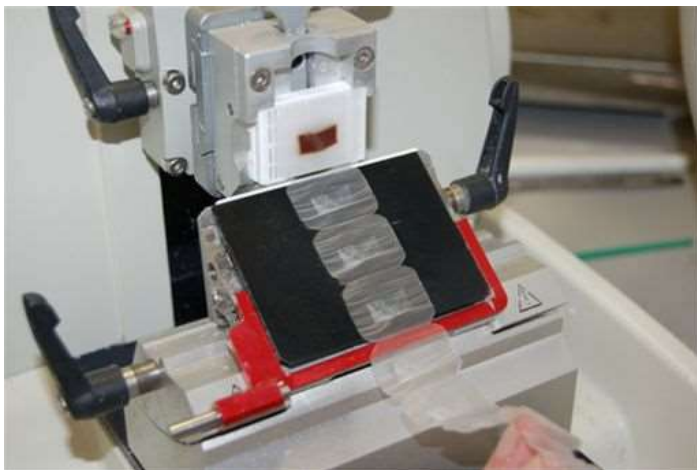


Figura 1.9. Secciones cortadas de un bloque de parafina utilizando un microtomo giratorio [5]



Figura 1.10. Sección de parafina flotando en agua tibia y montaje en un portaobjetos de microscopio [5]

vii. Tinción

Los cortes de parafina son incoloros ya que muy pocas células tienen color, de modo que es necesaria una tinción para poder ver las muestras bajo el microscopio óptico.

Las técnicas utilizadas pueden ser inespecíficas, teñir una gran cantidad de las células de la misma forma, o teñir selectivamente grupos químicos o moléculas químicas particulares dentro de las células o tejidos. La coloración se hace, generalmente aplicando dos tintes, uno que tiñe algunos de los componentes de las células de un color brillante, junto con una contratinción que tiñe el resto de la célula de un color diferente [6].

Se pueden distinguir dos componentes importantes en los colorantes: el cromógeno, que aporta color y el auxocromo, que posibilita la unión a elementos del tejido. El cromóforo es la organización molecular del cromógeno y es la responsable de absorber un espectro determinado de longitudes de onda. El auxocromo que se une al cromógeno puede influir en su coloración, muchos colorantes tienen más de un grupo auxocrómico.

El auxocromo puede ser un grupo ionizable, es decir, un grupo que reacciona de manera covalente con iones metálicos, o puede reaccionar covalentemente con el sustrato que en este caso es el tejido. Los colorantes suelen ser hidrosolubles, aunque hay colorantes que carecen de grupos ionizables y sirven para teñir sustancias grasas.

Según la naturaleza química del cromóforo hay varios tipos de colorantes: nitrosos, ozoicos, derivados de la antroquinona, derivados de la acridina, derivados de iminas quinónicas, derivados de diferrilamento y triferrilmetano, derivados del xanteno y derivados de las talocianinas.

Según la naturaleza química del radical auxocromo, los colorantes se clasifican en:

1. **Básicos:** son sales en las que la base es, normalmente, una amina, que aporta el color, mientras que la parte ácida es incolora. Es decir, son colorantes catiónicos. Esta atracción por las sustancias ácidas del tejido hace que tiñan el ADN o ciertos componentes de la matriz extracelular, como los glicosaminoglicanos. La unión es por atracción eléctrica. Así que son capaces de marcar el núcleo y el ARN, sobre todo el ARNr presente en los ribosomas por ser muy abundante, así como ciertas matrices extracelulares ricas en componentes ácidos. Algunos ejemplos de colorantes básicos son la tionina, safranina, azul de toluidina, el azul de metileno o la **hematoxilina**.
2. **Ácidos:** son sales con un anión coloreado y la base incolora. Son derivados de grupos sulfónicos, carboxilos o hidroxilos fenólicos. Tienen atracción por sustancias básicas, sobre todo por estructuras proteicas localizadas en el citoplasma celular y también por el colágeno en la matriz extracelular. La unión es por atracción eléctrica. Los ejemplos de colorantes ácidos son la fucsina ácida, verde rápido, naranja G o la **eosina**.

1.2.2. Tinción Hematoxilina & Eosina

El sistema de tinción más utilizado se llama hematoxilina & eosina (H&E). Como su nombre indica este colorante contiene los tintes de la hematoxilina (básico) y la eosina (ácido).

La hematoxilina es un colorante acidófilo. Se trata de un tinte llamado hemateína y se usa en combinación con los iones de aluminio (Al^{+3}). Se emplea para teñir estructuras ácidas, o basófilas, de un azul violáceo. En realidad, la hematoxilina no es estrictamente un tinte básico, pero se usa con un "mordiente" que hace que esta tinción actúe como un tinte básico. Las sales de aluminio se unen al tejido y la hematoxilina se une al mordiente (sales de aluminio).

La eosina es un colorante ácido, está cargado negativamente. Tiñe las estructuras básicas, o acidófilas, de un color rosado o rojizo. También se conoce como colorante 'eosinófilo'.

Esto significa que el núcleo y las partes del citoplasma que contienen ARN se tiñen de color púrpura y el resto del citoplasma se tiñe de color rosa [7].

En la siguiente imagen, Figura 1.11, se puede ver un ejemplo de la tinción H&E con la que se ha trabajado en este proyecto.

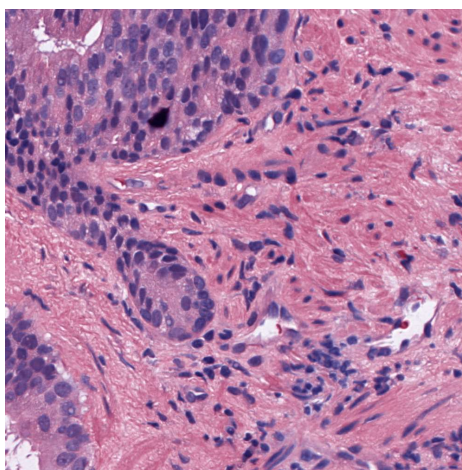


Figura 1.11. Muestra H&E, paciente 16B0008066 HE
5_060220171540_59655_119188

1.2.3. Análisis de las imágenes

El patólogo observa a microscopio las muestras histológicas con el fin de encontrar patrones que conduzcan a la determinación de si la muestra observada tiene cáncer.

La morfología y tamaño de los núcleos puede ser de ayuda en la identificación del grado de cáncer que presenta el paciente. Escanear estas muestras histológicas y analizar de manera automática las imágenes con el fin de segmentar los núcleos puede significar un gran avance en la medicina. Estas segmentaciones pueden ayudar al patólogo a buscar exactamente en las zonas de la muestra más dudosas sin perder tiempo observando toda la muestra.

La segmentación de los núcleos no resulta sencilla, ya que tienen formas bastante diversas, puede haber solapamiento de núcleos o pueden tener defectos de tinción que haga que los núcleos no sean tan reconocibles.

El presente trabajo pretende abordar este problema mediante la realización de una red neuronal convolucional de tipo *encoder-decoder*. La solución que se propone es un modelo predictivo entrenado con una base de datos dotada de una gran cantidad de imágenes con núcleos segmentados. La red neuronal será capaz de obtener 3 etiquetas diferenciadas que serán interior de núcleo, borde de núcleo y fondo. Este resultado facilitaría la labor del día a día del patólogo.

1.3. *Machine learning, Deep learning e Inteligencia Artificial*

En los últimos años, han aparecido una serie de términos para describir las nuevas técnicas de procesamiento de datos. Se trata de términos como aprendizaje automático (*machine learning*), aprendizaje profundo (*deep learning*) e inteligencia artificial.

Todos estos términos están relacionados y pueden superponerse unos a otros. No obstante, existen diferencias importantes que deben entenderse para comprender cada una de las técnicas. La forma más fácil de pensar en su relación es visualizarlos como círculos concéntricos, Figura 1.12. La idea de inteligencia artificial (IA) fue la primera en surgir, más tarde aparecieron los conceptos de aprendizaje automático y finalmente el aprendizaje profundo, que está impulsando la explosión de la IA de hoy en día.

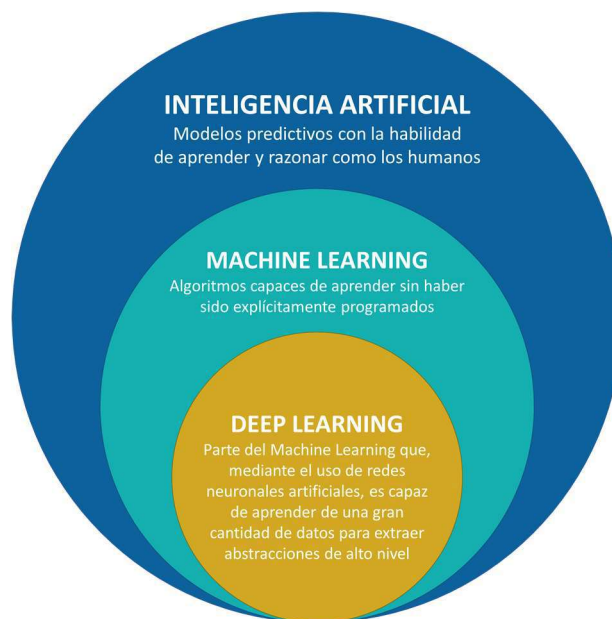


Figura 1.12. *Inteligencia artificial, machine learning y deep learning*

1.3.1. *Inteligencia Artificial*

Como se puede observar en la Figura 1.12, la inteligencia artificial engloba al *machine learning* (ML) y al *deep learning* (DL).

John McCarthy, ampliamente reconocido como uno de los padrinos de la IA, la definió como "la ciencia e ingeniería de hacer máquinas inteligentes" [8].

Aunque también es posible encontrar otras definiciones de la inteligencia artificial:

- Rama de las ciencias de la computación que trata de simular el comportamiento inteligente en los ordenadores.
- La capacidad de una máquina para imitar el comportamiento inteligente del humano.
- Sistema informático capaz de realizar tareas que normalmente requieren de la inteligencia humana, tales como la percepción visual, el reconocimiento de voz, la toma de decisiones, y la traducción entre lenguas.

Hay una gran variedad de maneras de simular la inteligencia humana, y algunos de estos métodos son más inteligentes que otros. La IA puede entenderse como una infinidad de instrucciones *if-then*, es decir, reglas explícitamente programadas por un humano. Esto podría ser el caso de los asistentes virtuales que se encuentran en gran parte de las empresas al acceder al apartado de contacto, tanto al llamar por teléfono, donde se debe especificar con un número o una frase al departamento con el que nos queremos comunicar, como en los chats virtuales (*bots*) que se encuentran en las páginas webs. Por otro lado, se pueden encontrar otros ejemplos de IA, como los asistentes personales: Cortana, Siri o el asistente de Google son algunos de ellos, y se puede encontrar en la inmensa mayoría de móviles y ordenadores. Hoy en día, la frase inteligencia artificial, se usa de manera amplia y, en general, para referirse a cualquier tipo de modelo predictivo de aprendizaje automático.

La inteligencia artificial se ha ido optimizando para intentar solventar el problema de tener que programar explícitamente los algoritmos "inteligentes". Concretamente el ML es un conjunto de técnicas que permite resolver problemas basados en IA. Pero no se puede olvidar que, además del ML existen otras técnicas como, por ejemplo, las heurísticas para resolver problemas de IA.

1.3.2. *Machine learning*

El *machine learning*, aprendizaje automático, en su nivel más básico, se refiere a cualquier tipo de programa de computación que pueda 'aprender' por sí mismo sin tener que ser explícitamente programado por un humano. Su origen tiene lugar en el artículo seminal de Alan Turing "*Computing Machinery and Intelligence*" de 1950, que incluía una sección de su famosa "Máquina de aprendizaje" que podría engañar a un ser humano para que crea que es real [9].

Actualmente, el aprendizaje automático es un término ampliamente utilizado que abarca muchos tipos de programas. Se puede encontrar en el análisis de *big data* y la minería de datos. Encontramos algoritmos de aprendizaje automático en los programas predictivos que utilizamos cada día, incluidos los filtros de spam, los recomendadores de productos y los detectores de fraude.

A su vez, existen diferencias entre el aprendizaje automático supervisado y el aprendizaje automático no supervisado, así como con el modelado en conjunto, que utiliza una combinación de técnicas de abordaje, y el aprendizaje semi-supervisado, que combina enfoques supervisados y no supervisados.

i. Aprendizaje supervisado

El **aprendizaje supervisado** es aquel en el usuario entrena el modelo predictivo para generar una respuesta basada en un conjunto de datos conocido y etiquetado. Los algoritmos de clasificación y regresión, incluidos los *random forest*, los árboles de decisión y los *support vector machine (SVM)*, se utilizan comúnmente para tareas de aprendizaje supervisadas.

ii. Aprendizaje no supervisado

El **aprendizaje automático no supervisado** es aquel en el que los algoritmos generan respuestas en forma de datos desconocidos y sin etiquetar. Los científicos de datos comúnmente usan técnicas no supervisadas para descubrir patrones en nuevos conjuntos de datos. Los algoritmos de agrupación en clústeres, como *K-means*, se utilizan a menudo en el aprendizaje automático no supervisado.

Es posible programar algoritmos de aprendizaje automático utilizando una variedad de tecnologías y lenguajes, incluidos Java, MATLAB, Python, Scala y otros. También pueden usar marcos de aprendizaje de máquina pre-construidos para acelerar el proceso.

1.3.3. *Deep learning*

El *Deep learning*, aprendizaje profundo, es una forma de aprendizaje automático que puede utilizar algoritmos supervisados, no supervisados, o ambos. Si bien, no es necesariamente nuevo, el aprendizaje profundo se ha hecho popular en los últimos años. Es capaz de acelerar la solución de ciertos tipos de problemas informáticos difíciles, especialmente en los campos de la visión por computador y el procesamiento del lenguaje natural.

Las redes neuronales pueden tener muchas capas ocultas. El aprendizaje profundo se basa en la rama de aprendizaje de representación (o aprendizaje de características) de la teoría del aprendizaje automático.

Al extraer abstracciones complejas de alto nivel como representaciones de datos a través de un proceso de aprendizaje jerárquico, los modelos de aprendizaje profundo producen resultados más rápidamente que los enfoques estándar de aprendizaje automático. En un lenguaje sencillo, un modelo de aprendizaje profundo aprenderá las características que son importantes por sí mismas, en lugar de requerir que el científico de datos seleccione manualmente las características que a su modo de ver describen el problema bajo estudio de manera relevante.

Las red neuronales convolucionales (CNNs) surgen con el objetivo de analizar patrones a nivel local, maximizando así el rendimiento en problemas que incluyen la imagen como dato de entrada.

El *Deep learning* es tan popular hoy en día debido a que las CNN funcionan mucho más rápido y de manera eficiente en las Graphic Processing Units (GPUs), la tecnología en este sector ha avanzado considerablemente en los últimos años. Por otro lado, debido a esta expansión, recientemente han nacido diversas librerías de *Deep learning*. TensorFlow es un ejemplo de un marco de desarrollo de software, creado por Google, que está experimentando un gran interés; Keras, Caffe, Torch y Theano son otros ejemplos.

El *Deep learning* es, sin duda, la piedra angular donde va a sustentarse el trabajo que estamos desarrollando.

1.4. Marco del proyecto

1.4.1. SICAP

SICAP es el acrónimo de “Sistema de interpretación de imágenes histopatológicas para la detección del cáncer de próstata”. En este proyecto participan el grupo de investigación CVBLab (*Computer Vision and Behaviour Analysis Lab*) perteneciente a la Universidad Politécnica de Valencia e *investigadores* del Hospital Clínico Universitario de Valencia. SICAP busca la creación de una base de datos privada con el fin de trabajar la segmentación automática y en la extracción de rasgos basados tanto en la estructura de los tejidos (ayudándose de la experiencia de los médicos) como en las técnicas de filtrado (sin necesidad de apoyo médico). Estas características extraídas serán utilizadas como entrada de clasificadores que permita la detección del cáncer. Por otra parte, también se implementan técnicas de aprendizaje profundo basadas en Deep Learning para la tarea de clasificación y segmentación automática. Cabe destacar que en este proyecto se tendrá que lidiar con ciertos problemas que presentan las dimensiones de las imágenes de muestras histológicas, ya que se trabaja con unas dimensiones de tamaño considerablemente grandes y una alta resolución.

En conclusión, SICAP es un proyecto compuesto por un equipo coordinado y multidisciplinar en el que colaboran médicos, ingenieros y científicos, procedentes de distintas universidades y centros de investigación con la finalidad de proporcionar un software (con vistas a la integración en la empresa) para ayuda al diagnóstico del cáncer.

1.4.2. Challenge MoNuSeg

El challenge MoNuSeg es un evento oficial de MICCAI 2018 (*International Conference on Medical Image Computing and Computer Assisted Intervention*).

Este reto facilitó un total de 30 imágenes multiórgano con sus respectivas etiquetas de núcleos a la comunidad científica. En él se pretendía encontrar las mejores técnicas de segmentación automática de núcleos de imágenes histológicas H&E obtenidas de diferentes hospitales que abarcan múltiples pacientes y órganos basadas en Deep learning.

El CVBLab se presentó a este concurso y consiguió el puesto número 26 de 32 participantes, siendo el primer clasificado de los representantes españoles.

1.5. Estado del arte

Una vez hecha esta introducción teórica acerca de la histología, el cáncer y la inteligencia artificial, vamos a ver cómo se unen estos campos.

La información fenotípica presente en la histología refleja el efecto agregado de las alteraciones moleculares en el comportamiento de las células cancerosas y proporciona una lectura visual conveniente de la agresividad de la enfermedad. Sin embargo, las evaluaciones humanas de histología son altamente subjetivas y no son repetibles. Por lo tanto, el análisis computacional de imágenes de histología ha recibido una atención significativa. Ayudados por los avances en microscopios de barrido de diapositivas y computación, se han desarrollado una serie de algoritmos de análisis de imágenes para la clasificación y la identificación de metástasis de ganglios linfáticos en múltiples tipos de cáncer.

Las aplicaciones de análisis de imágenes médicas se han basado, en gran medida, en los enfoques de ingeniería de características, donde se utilizan líneas de algoritmos para delinear explícitamente estructuras de interés utilizando algoritmos de segmentación para medir características predefinidas de estas estructuras, que se cree que son predictivas, y usar estas características para entrenar modelos capaces de predecir los resultados de un paciente. En Septiembre de 2008, Mahmood et al [10], proponen que el uso de redes neuronales convolucionales han sido probadas como robustas y personalizables.

El paradigma de aprendizaje de características de las redes neuronales convolucionales, CNN, aprende de forma adaptativa a transformar imágenes en características altamente predictivas para un objetivo de aprendizaje específico. Las imágenes y las etiquetas de los pacientes se presentan a una red compuesta de capas interconectadas de filtros convolucionales que resaltan patrones importantes en las imágenes, y los filtros y otros parámetros de esta red se adaptan matemáticamente para minimizar el error de predicción.

El aprendizaje de características evita una definición de características sesgada a priori y no requiere el uso de algoritmos de segmentación que a menudo se confunden con artefactos y variaciones naturales en el color y la intensidad de la imagen. Si bien el aprendizaje de características se ha convertido en el paradigma dominante en las tareas generales de análisis de imágenes, las aplicaciones médicas presentan desafíos únicos. No obstante, se necesitan grandes cantidades de datos etiquetados para capacitar a las CNN a predecir las estructuras, y las aplicaciones médicas a menudo sufren déficits de datos que limitan el rendimiento. A pesar de estos desafíos, las CNN se están utilizando con éxito para el análisis de imágenes médicas [11].

Las redes neuronales convolucionales profundas (CNN) se han convertido en una importante herramienta de análisis de imágenes. La capacidad de las CNN para aprender características predictivas a partir de datos de imágenes simbolizan un cambio de paradigma que presenta oportunidades interesantes en la imagen médica [11].

Desde 2014, se han propuesto numerosos métodos de segmentación de imágenes basados en redes neuronales convolucionales. El artículo *Fully Convolutional Networks for Semantic Segmentation* [12] fue el primero en introducir una red neuronal totalmente convolucional (FCN) para la segmentación semántica. En comparación con los modelos anteriores, se demuestra que el algoritmo FCN es mucho más eficiente y preciso. La conversión de capas completamente conectadas en redes neuronales convolucionales hace posible predecir el mapa de probabilidades de los objetos en la imagen segmentada. En 2015 se propone una arquitectura de red basada en FCN, U-net [13], que ganó el gran desafío para la detección automatizada por computador de caries en radiografías. Más tarde, en 2016, una arquitectura de red neuronal introduce, por primera vez, bloques residuales [14] aplicándolos para fusionar diferentes niveles de información semántica.

El presente trabajo propone una red neuronal convolucional basándose en la U-net [13] para el problema de segmentación de núcleos. Además, se pretende generar otra red U-net que incluya bloques residuales (ResNet) [14] con el fin de estudiar las diferencias de la segmentación entre ambas.

2. OBJETIVOS

El principal objetivo de este trabajo fin de grado es el de crear un modelo de segmentación de núcleos basado en aprendizaje profundo.

- Conformar una base de datos compuesta por regiones de interés (ROI) provenientes de una base de datos pública de imagen histológica multiórgano y una base de datos de imágenes hitológicas de prostata adquiridas en el Hospital Clínico de Valencia. Todas las imágenes tendrán un tamaño de 1000 x 1000 píxeles.
- Estudiar los efectos de preprocesado sobre el modelo de segmentación de núcleos.
- Diseñar y analizar arquitecturas secuenciales y residuales *encoder – decoder* para la segmentación de núcleos.
- Diseñar y desarrollar un algoritmo de postprocesado capaz de diferenciar los diferentes núcleos de una imagen histológica a nivel de instancia.
- Hacer una evaluación de los resultados obtenidos por las redes propuestas, extraer conclusiones a todos los niveles del proceso. Comprobar si el objetivo principal del trabajo se ha alcanzado con el éxito esperado y analizar el comportamiento del sistema determinando sus fortalezas y debilidades, finalmente, en base a las conclusiones extraídas, ver las posibles líneas futuras de trabajo que propicien una mejora de las prestaciones del sistema y permitan abarcar nuevos ámbitos más allá de los perseguidos en el presente trabajo.

3. MATERIAL Y MÉTODOS

3.1. Material

A continuación, vamos a detallar los diferentes recursos que se han utilizado en este proyecto.

3.1.1. Base de datos

La base de datos con la que se ha trabajado es la unión del dataset alojado en la web del challenge MoNuSeg [15], que cuenta con 30 muestras de diversos órganos, junto con 16 nuevas imágenes que se generaron a partir de muestras histológicas de biopsias de próstata procedentes del departamento de anatomía patológica del Hospital Clínico Universitario de Valencia. Este *batch* de imágenes provenientes del hospital está conformado por pacientes sanos y pacientes enfermos de cáncer, por lo que se encuentran núcleos mucho más irregulares que en *batch* de MoNuSeg. En la subsección 1.2.1 se ha explicado detalladamente los procesos por los que pasa una biopsia hasta convertirse en una imagen histológica como las que se han empleado en el proyecto.

Por tanto, este trabajo se ha realizado a partir de 46 imágenes, todas ellas digitalizadas a 40x de zoom óptico. Esta información queda desglosada en la Tabla 3.1.

Tabla 3.1. Base de datos

Base de datos				
MoNuSeg	30 imágenes	6 de pecho 6 de riñón 6 de hígado 6 de próstata 2 de vejiga 2 de colon 2 de estómago	22 entrenamiento 8 test	30 entrenamiento 16 test
Hospital Clínico Universitario de Valencia	16 imágenes	16 de próstata	8 entrenamiento 8 test	

En la tabla podemos ver que las 46 imágenes se han dividido en dos grupos, concretamente 30 se han destinado para entrenamiento y 16 para testeo. Cabe destacar que comúnmente se destina el 80% de la base de datos al entrenamiento, por lo que debería entrenarse con unas 36-37 imágenes, pero en este proyecto se ha pretendido emular las condiciones del challenge MoNuSeg [15], donde se tenían 30 imágenes de entrenamiento.

Con el fin de que la red neuronal convolucional no aprenda a segmentar mejor las imágenes procedentes del sensor que escaneó el dataset MoNuSeg y también se entrenase con imágenes que tienen cáncer, se han extraído 8 de las 30 imágenes y se han sustituido por 8 del Clínico. De este modo la red entrena con imágenes del sensor que se tiene en el Hospital Clínico de Valencia y con formas nucleares cancerígenas, a la vez que seguía manteniendo imágenes del resto de órganos, y no solo de próstata. Este hecho posibilita la generación de un modelo segmentador multiórgano e independiente del dispositivo de adquisición.

Las imágenes procedentes del departamento de patología se han recortado en patches de regiones de interés (ROI), dado que el tamaño original de las imágenes es del orden de $[50000, 90000] \times [90000, 190000]$ píxeles, ocupando desde 300 MB hasta 1.5 GB.

Por otro lado, la base de datos se ha complementado con las máscaras (ternaria y de instancia) que contienen las etiquetas de todos los núcleos. En la Figura 3.1 se puede ver un esquema que ilustra perfectamente cómo se conforma el dataset final. En el apartado de Metodología explicaremos cómo se logra la creación de patches, normalización de color, extracción de los *.xml* y la creación de las máscaras.

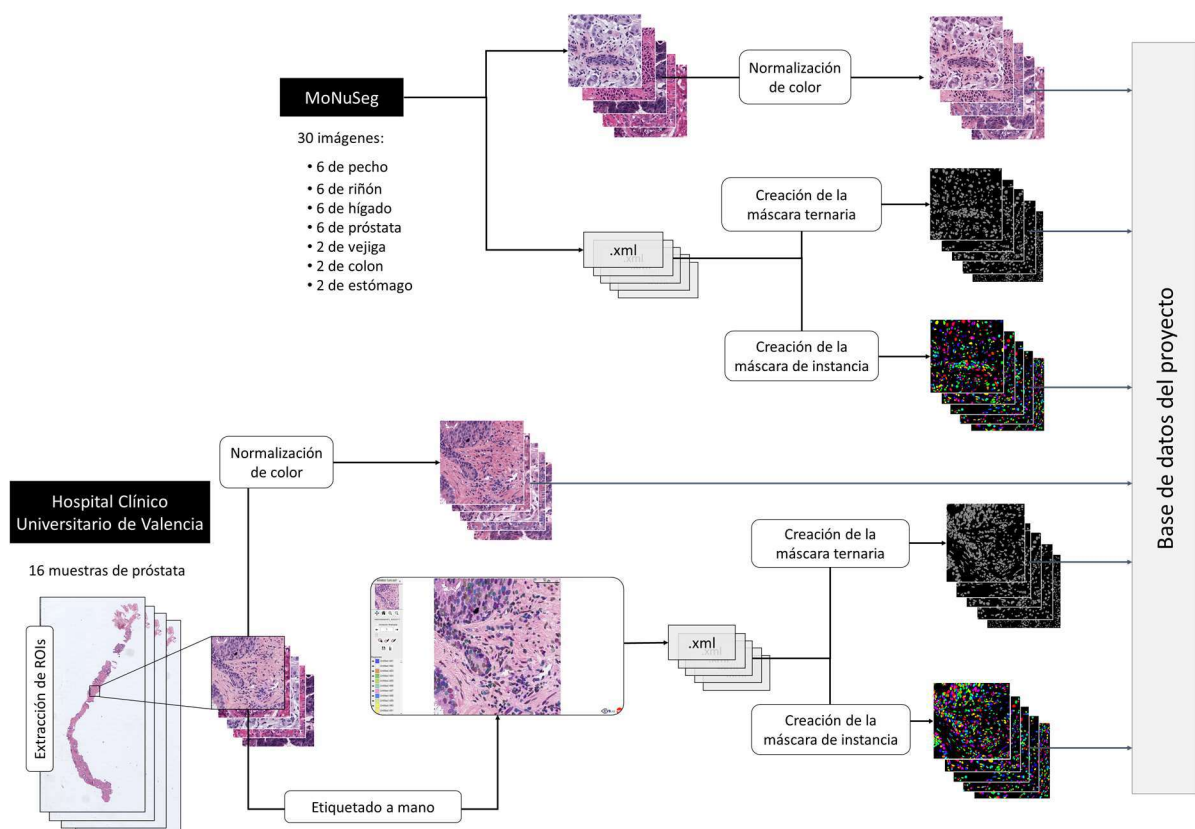


Figura 3.1. Conformación de la base de datos

3.1.2. Equipo y entorno de programación

El presente proyecto se ha desarrollado en una máquina DELL XPS 15 9530 que monta un procesador Intel® Core™ i7-4712HQ CPU @ 2.30GHz (8 CPUs), cuenta con 16 GB de memoria RAM¹, con una tarjeta gráfica integrada Intel® HD Graphics 4600 y una tarjeta gráfica dedicada NVIDIA GeForce GT 750M con 2 GB de VRAM². El sistema operativo es Windows 10 Home de 64 bits. El portátil monta una pantalla táctil QHD con la que se segmentó a mano los núcleos.

Para poder implementar los algoritmos de *deep learning* se ha necesitado el lenguaje de programación Python en su versión 3.6 con la IDE JetBrains Pycharm 2019.1.3x64 y *frameworks* numpy, keras (con *bakend* de TensorFlow), cv2, scipy.io y matplotlib.

Tanto la parte de pre-procesado como la de postprocesado y la evaluación de los modelos de segmentación resultantes se han desarrollado y ejecutado en MATLAB^R v.R2018b, The MathWorks, Inc. (Natick, Massachusetts, Estados Unidos). Este programa combina un entorno de escritorio perfeccionado para el análisis iterativo y los procesos de diseño con un lenguaje de programación que expresa las matemáticas de matrices y arrays directamente [16].

Para las etapas del aprendizaje profundo empleadas, se ha hecho uso de un servidor de computación de altas prestaciones compuestos por un procesador Intel i7 @4.20GHz, 32GB de RAM y tarjetas gráficas NVIDIA Titan V, que se encuentran disponibles en el CVBLab. Adicionalmente disponen de un servidor (Synology DS416) en el que se almacenan los datos referentes al proyecto.

3.2. Metodología

Este apartado abordaremos la metodología empleada en el proyecto. En primer lugar, se va a desarrollar detalladamente los procesos vistos en la Figura 3.1, es decir, el preprocesado necesario para conformar la base de datos. Tras esto, se analizará en profundidad los fundamentos de las redes neuronales para poder detallar, finalmente, la arquitectura que se ha creado en este trabajo. Por último se detallará el postprocesado necesario para la creación de las máscaras (ternaria y de instancia) de núcleos.

3.2.1. Extracción de regiones de interés y etiquetado

Como se ha comentado en la Sección de la base de datos, el primer paso ha sido recortar las imágenes ofrecidas por el departamento de patología en regiones de interés (ROI). Las imágenes originales tienen un peso de 1 GB aproximadamente y están en formato *.tif* con un tamaño del orden de [50000, 90000] × [90000, 190000] píxeles. Tras llevar a cabo el proceso de extracción de ROIs, se obtienen imágenes con un tamaño de 1000 x 1000 píxeles manteniendo los canales rojo, azul y verde (RGB), este fenómeno se ilustra en la Figura 3.2.

El recorte se ha realizado con MATLAB y ha sido necesario MATLAB Support for MinGW-w64 C/C++ Compiler con la librería Openslide. Esto es debido a que las imágenes de muestras histológicas son de gran tamaño y de una alta resolución. Leer estas imágenes usando las herramientas estándar no es

¹ *Random Access Memory (RAM)*: Memoria de acceso aleatorio

² *Video Random Access Memory (VRAM)*: Memoria de vídeo de acceso aleatorio

posible, dado que están diseñadas para imágenes que se pueden descomprimir en la memoria RAM. Openslide solventa este problema reduciendo el tamaño y comprimiendo la resolución global cuando se visualiza la imagen, permitiendo así elegir la zona de recorte de las ROIs.

Por tanto, tras instalar Openslide ha sido posible cargar y visualizar las imágenes provenientes del Hospital Clínico de Valencia, tras eso, con la función `ginput`³ se han generado las ROIs. Además, con esta función es posible almacenar las coordenadas de dónde se encuentra la región de interés en la imagen original.



Figura 3.2. Extracción de ROIs

Una vez conformada la base de datos del Hospital Clínico se ha procedido al etiquetado de las imágenes. Cabe destacar que esta es una labor muy costosa, cada imagen cuenta con una media de 400 núcleos que se han tenido que segmentar a mano. La dificultad de delinear manualmente con precisión y fiabilidad estas imágenes es muy alta, ya que debe atenderse a la forma de cada núcleo. Algunos ejemplos con los que puede encontrarse el experto que ejecuta esta labor son la existencia de solapamiento con otros núcleos, en este caso deberán separarse consecuentemente en dos etiquetas. Otro posible caso es que la célula se encuentra en algún proceso biológico, como la mitosis⁴, donde la cápsula que delimita el material genético se disuelve para proceder a la conformación de las dos células hijas, ante este fenómeno el profesional podría verse en la tesitura de considerar 1 o 2 núcleos. En la Figura 3.3 pueden verse una serie de ejemplos que ayudan a ilustrar mejor esta idea. Por último, los núcleos patológicos, es decir, aquellos en los que se pueden apreciar irregularidades asociables al cáncer, también suponen un gran reto para la persona que segmenta a mano, porque suele ser muy difícil distinguir la forma que tienen dichos núcleos. Estos son, sin duda, de los más

³ `ginput` permite identificar las coordenadas de n puntos. Para elegir el punto se mueve el cursor a la ubicación deseada y se pulsa un botón del ratón o una tecla en el teclado.

⁴ **Mitosis:** Proceso de reproducción de una célula que consiste, fundamentalmente, en la división longitudinal de los cromosomas y en la división del núcleo y del citoplasma; como resultado se constituyen dos células hijas con el mismo número de cromosomas y la misma información genética que la célula madre.

importantes para la base de datos, dado que se pretende que la red neuronal sea capaz de distinguir todo tipo de núcleo, tanto sanos, solapados, en proceso de mitosis o cancerígenos.

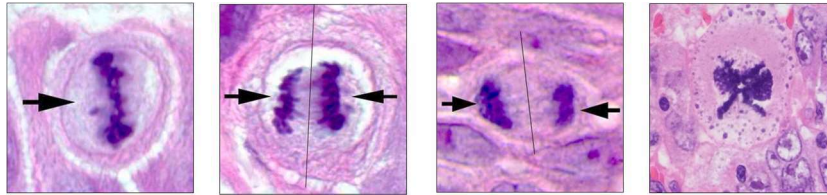


Figura 3.3. Muestras histológicas en proceso de mitosis [17]

Para poder generar y almacenar esta gran cantidad de etiquetas ha sido necesaria una aplicación de anotación desarrollada sobre una librería de código abierto y adaptada por los ingenieros del CVBLab a las necesidades de los patólogos. Con ella es posible dibujar en las imágenes y anotar los grados de patología. En lo referente a este proyecto, se ha utilizado esta plataforma para delinear los núcleos y guardar las anotaciones de cada uno de ellos por separado. La plataforma web, denominada *MicroDraw* y un ejemplo de la segmentación manual de núcleos se puede ver en la Figura 3.4.

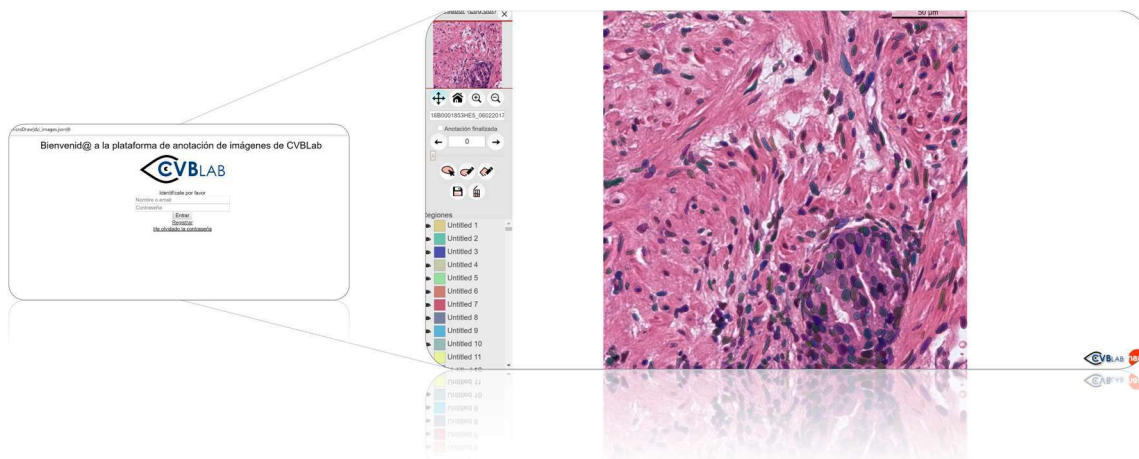


Figura 3.4. Plataforma *MicroDraw*

3.2.2. Obtención de las máscaras ternarias y de instancias

El resultado de la segmentación manual de las imágenes de la base de datos que se ha generado mediante la plataforma *MicroDraw* se va a procesar para conformar el *ground truth*⁵. Con esto se tiene la información necesaria para realizar un proceso de aprendizaje supervisado y posteriormente evaluar lo exactas que son las segmentaciones automáticas que se generan con el modelo predictivo propuesto. El procesamiento de la segmentación manual se realiza a dos niveles.

En primer lugar, tiene que establecerse una conexión con la base de datos de *MicroDraw* y descargar un *.xml* para cada imagen segmentada. Estos *.xml* contienen las coordenadas de los bordes donde se encuentra cada núcleo de la imagen, segmentado previamente mediante la herramienta

⁵ **Ground truth** hace referencia a la segmentación manual de las imágenes de estudio hecha por un experto. Esta segmentación es la que se va a utilizar para comparar con los resultados obtenidos del modelo predictivo que se está desarrollando.

MicroDraw. El algoritmo que se implementa en MATLAB descarga y convierte el XML en una estructura y guarda un archivo *.mat* que contiene tantas estructuras como imágenes segmentadas.

El siguiente algoritmo recorre cada una de las regiones donde se encuentra un núcleo. Durante este proceso se reconstruyen los puntos que forman el borde del núcleo teniendo en cuenta si falta algún píxel que forme parte del núcleo. De ser así, se genera una unión entre los píxeles más cercanos consiguiendo cerrar el borde del núcleo. Una vez con el núcleo cerrado se rellena su interior y se fijan las siguientes intensidades: Borde a 2, interior a 1 y fondo a 0. Esto se hace para todos los núcleos que conforman la imagen y se guarda la máscara, a la que denominaremos máscara ternaria, que servirá para entrenar la red. Por otro lado, se da una intensidad a cada uno de los núcleos (mismo valor para interior y borde de núcleo) y con esto se genera una nueva máscara que tendrá cada núcleo a una intensidad distinta, a esta máscara la que denominaremos máscara de instancia, y es la que utilizaremos como *ground truth*. En la siguiente Figura 3.5 podemos ver el resultado de la máscara ternaria, donde se han normalizado las intensidades (borde a 255, interior a 128 y fondo a 0) y la máscara de intensidades que en la que se ha coloreado cada valor de núcleos de un color distinto.

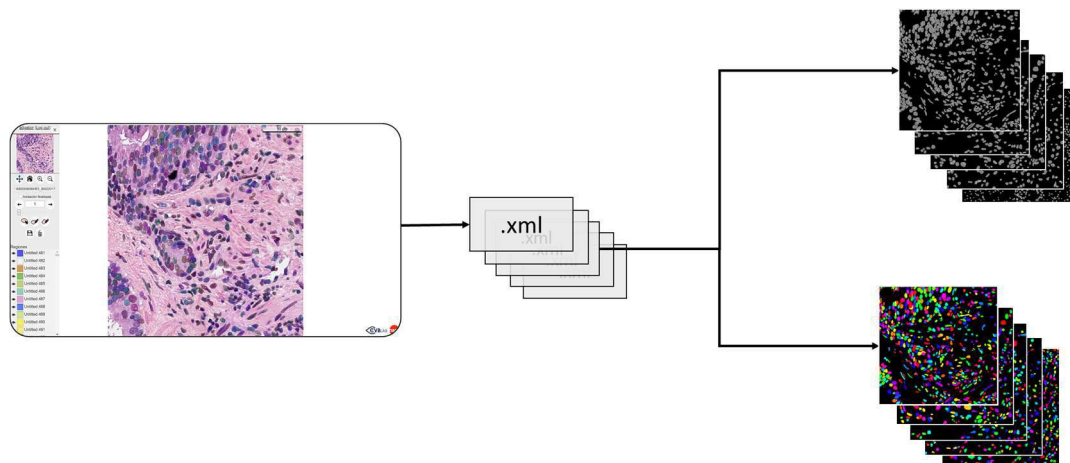


Figura 3.5. Creación de máscara ternaria y máscara de instancia

3.2.3. Normalización de color

Uno de los desafíos para la segmentación de tejidos es la gran variabilidad en color existente en las muestras debido a los reactivos H&E, tiempo de tinción y la forma en la que toma la imagen el sensor.

Se ha demostrado que la normalización de color mejora la segmentación de las imágenes [18], por lo que este recurso es necesario cuando las imágenes a segmentar proceden de distintos tejidos biológicos, tiempos de tinción y sensores de captura.

Existen distintos algoritmos de normalización de colores capaces de conseguir que las variaciones de color entre las distintas imágenes se reduzcan.

En este caso se ha empleado un algoritmo que preserva la estructura biológica basándose en el modelado de mezclas de colores en factorización de matriz no negativa dispersa (SNMF) [18]. Esta técnica primero evalúa un mapa de densidad de tinciones para una imagen dada de la siguiente manera.

Las imágenes teñidas absorben luz, esto hace que la relación entre la tinción y su absorción siga la Ley de Beer-Lambert. Sea ζ los diferentes canales de color en el espacio RGB, es decir, $\zeta \in R, G, B$. Cada D_ζ es el vector de densidad óptica (**OD**) correspondiente a los diferentes canales de color. De acuerdo con la Ley de Beer-Lambert, D_ζ de cada canal de color para el píxel p se expresa en términos de una absorbancia que se define como

$$D_\zeta(p) = -\ln\left(\frac{I_\zeta(p)}{I_0(p)}\right) = \varepsilon \times \ell \times \eta \quad (3.1)$$

donde ε , ℓ y η representan respectivamente una capacidad de absorción del atenuador de la muestra, la distancia que recorre la luz a través de la muestra y la concentración de tinción en la muestra. I_0 es la intensidad de la luz que ingresa a la muestra e I_ζ es la intensidad de la luz detectada en el canal ζ después de la absorción cuando I_0 pasa la muestra.

De la ecuación (3.1) se puede concluir que las intensidades $I_\zeta \in \{I_R, I_G, I_B\}$ no dependen linealmente de la concentración de tinción η . Por lo tanto, estos valores de intensidad de la imagen RGB no se pueden usar directamente para la descomposición de cada una de las tinciones. Sin embargo, al aplicar la Ley de Beer-Lambert en (3.1), la densidad óptica correspondiente D_ζ para cada canal es lineal con η . Por lo tanto, antes de emplear SNMF para la descomposición, la imagen de intensidad RGB original se transforma en los vectores correspondientes D_ζ .

i. Deconvolución de color para la mezcla de color y sus limitaciones asociadas

La relación entre la cantidad de cada tinción [18] es

$$\phi(p) = [\phi_R(p), \phi_G(p), \phi_B(p)]^T$$

y **OD**

$$D(p) = [D_R(p), D_G(p), D_B(p)]^T$$

Para un píxel p puede expresarse como

$$D(p) = S\phi(p) \quad (3.2)$$

Donde

$$S = \begin{pmatrix} S_{R,1} & S_{G,1} & S_{B,1} \\ S_{R,2} & S_{G,2} & S_{B,2} \\ S_{R,3} & S_{G,3} & S_{B,3} \end{pmatrix} \quad (3.3)$$

es la matriz de tinción que define la asociación entre **OD** y la cantidad de cada tinción en el píxel p . Cada fila en S corresponde a cada canal de tinción.

Para todos los píxeles podemos reescribir (3.2) como la matriz

$$D = S\phi \quad (3.4)$$

donde

$$D = \begin{pmatrix} D_R(1) & D_R(2) & \cdots & D_R(n) \\ D_G(1) & D_G(2) & \cdots & D_G(n) \\ D_B(1) & D_B(2) & \cdots & D_B(n) \end{pmatrix} \quad (3.5)$$

y

$$\phi = \begin{pmatrix} \varphi_R(1) & \varphi_R(2) & \cdots & \varphi_R(n) \\ \varphi_G(1) & \varphi_G(2) & \cdots & \varphi_G(n) \\ \varphi_B(1) & \varphi_B(2) & \cdots & \varphi_B(n) \end{pmatrix} \quad (3.6)$$

La cantidad de tinte ϕ se puede obtener resolviendo la solución de forma cerrada de ($D = S\phi$) (3.4).

Este enfoque tiene limitaciones, dado que la matriz de tinción estimada \mathbf{S} será particular para el conjunto de datos en consideración, \mathbf{S} podría no ser óptimo para todas las imágenes en el conjunto de datos.

ii. *Sparse Non-negative Matrix Factorization (SNMF)*

Se puede demostrar que la matriz de datos \mathbf{D} en (3.5) es una matriz no negativa [18]. SNMF tiene como objetivo descomponer una matriz no negativa en una matriz base y una matriz de coeficientes. En general, PCA⁶, ICA⁷ y NMF/SNMF pueden considerarse enfoques de factorización matricial, con diferentes opciones de función objetivo y correspondientes restricciones. Dada una matriz no negativa $D \in R^{d \times n}$, donde n es el número de puntos de datos (o píxeles en la imagen) y d es la dimensión de cada punto de datos, es fácil ver que el rango deseado de la matriz \mathbf{D} es $k \leq \min d, n$. El algoritmo SNMF descompone \mathbf{D} en dos matrices no negativas $B \in R^{d \times k}$ y $X \in R^{k \times n}$ para que

$$D \approx B \times X \quad (3.7).$$

La solución de \mathbf{D} puede obtenerse resolviendo el siguiente problema de optimización:

$$\begin{aligned} \min_{B, X} \frac{1}{2} \|D - BX\|_F^2, \\ \text{s.t. } B, X \geq 0, \end{aligned} \quad (3.8)$$

donde \mathbf{B} y \mathbf{X} representan la matriz base y la matriz de coeficientes, respectivamente. Para evitar el problema de la varianza de escala y las soluciones no únicas, el algoritmo NMF estándar (3.8) se puede extender al algoritmo NMF disperso (SNMF) agregando un término de regularización y un término de dispersión de la siguiente manera:

$$\begin{aligned} \min_{B, X} \frac{1}{2} \|D - BX\|_F^2 + \frac{\alpha_1}{2} \|B\|_F^2 + \frac{\alpha_2}{2} \|X\|_F^2 + \frac{\beta}{2} \sum_{i=1}^n \|X(:, i)\|_1^2 \\ \text{s.t. } B, X \geq 0 \end{aligned} \quad (3.9)$$

⁶ PCA: Análisis de Componentes Principales

⁷ ICA: Análisis de Componentes Independientes

donde α_1 , α_2 y β son constantes positivas. El segundo y el tercer término representan la restricción de "suavidad" que tiene como objetivo penalizar grandes valores de los elementos en las matrices de base y coeficientes. El cuarto término de (3.9) es la restricción de escasez que se impone en la matriz de coeficientes utilizando la minimización de la norma L_1 ⁸. La restricción de escasez en la ecuación (3.9) impuesta en la matriz de coeficientes tiene como objetivo utilizar la dimensión deseada del espacio de color para representar mejor los diferentes componentes para una representación más significativa. La restricción de suavidad impuesta tanto en las matrices de base como en las de coeficiente tiene como objetivo suavizar la imagen y eliminar los ruidos blancos.

El principio de SNMF se puede aplicar a una imagen teñida con H&E para mostrar la mezcla de colores. En la Figura 3.6, cada canal de imagen RGB original se descompone en los componentes constitutivos y se expresa como $I = [I_{RT}; I_{GT}; I_{BT}]$. Al aplicar la Ley de Beer-Lambert en la ec. (3.1), la imagen de intensidad I puede ser representada en OD donde D está en el espacio RGB como $D = [D_{RT}; D_{GT}; D_{BT}]$. Aquí usamos D y D_z para representar la matriz OD y un vector de fila de D , respectivamente. D se descompone en $d \times k$ con base matriz B y unos coeficientes $k \times n$ de la matriz X basada en las ecuaciones ((3.7)–(3.9)). Se debe tener en cuenta que para dos imágenes teñidas (H&E), $k = 2$.

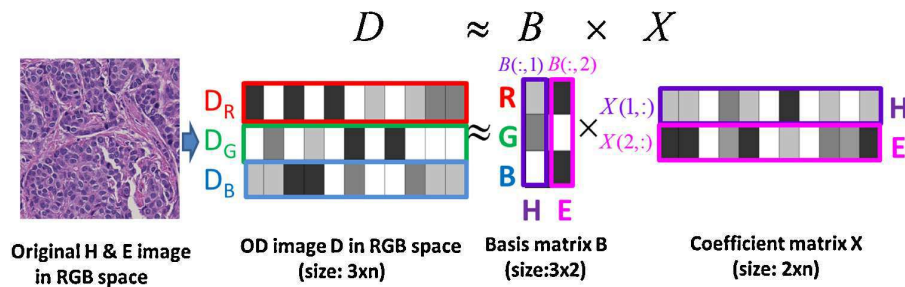


Figura 3.6. La ilustración de SNMF para descomposición de imagen teñida con H&E [18]

En resumen, lo que pasa en la figura es que la imagen original de H&E se transforma en un vector de fila de los datos de densidad óptica (OD). Luego, los datos OD , D , se descomponen en los componentes constitutivos: matriz base B y matriz coeficiente X . Permittiéndonos aplicar la normalización de color en función a la SNMF, como vemos en la Figura 3.7.

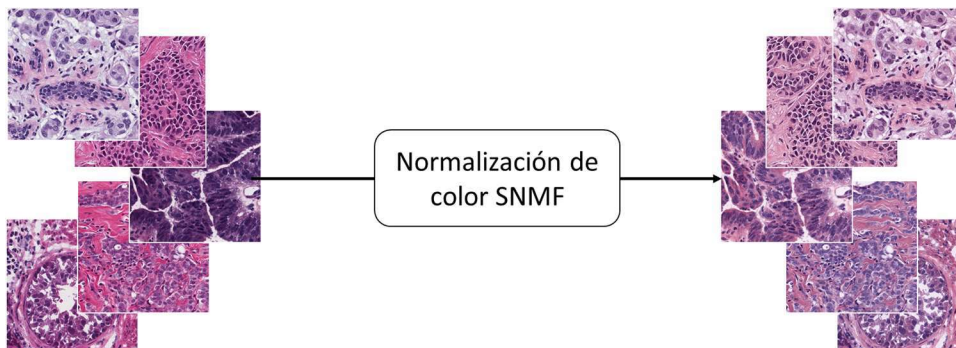


Figura 3.7. Normalización de color SNMF

⁸ Norma L_1 : $\|x\|_1 = \sum_{i=1}^n |x_i|$

3.2.4. Redes neuronales

Como se ha comentado en la introducción los dos paradigmas clave de aprendizaje son el aprendizaje supervisado y no supervisado. El aprendizaje supervisado es aprender a través de entradas pre-etiquetadas, que actúan como objetivos. Para cada muestra de entrenamiento se dispondrá de un conjunto de valores de entrada (vectores) y uno o más valores de salida asociados. El objetivo de esta forma de entrenamiento es reducir el error de clasificación general de los modelos, a través del cálculo correcto del valor de salida del ejemplo de entrenamiento.

El aprendizaje no supervisado difiere en que el conjunto de entrenamiento no incluye ninguna etiqueta. El éxito generalmente se determina por si la red puede reducir o aumentar una función de costo asociada. Sin embargo, es importante tener en cuenta que la mayoría de las tareas de reconocimiento de patrones centradas en imagen (tarea asociada al presente trabajo) generalmente dependen de la clasificación mediante el aprendizaje supervisado.

i. Redes neuronales artificiales (ANNs)

Las redes neuronales artificiales (ANNs) son sistemas de procesamiento computacional que están fuertemente inspiradas por la forma en que operan los sistemas nerviosos biológicos (como el cerebro humano). Las ANNs se componen principalmente de una gran cantidad de nodos computacionales interconectados (denominados neuronas), de los cuales el trabajo se entrelaza de manera distribuida para aprender colectivamente de la entrada a fin de optimizar su salida final.

La estructura básica de una ANN se puede modelar como se muestra en la Figura 3.8. La primera de las capas es la capa de entrada y generalmente recibe los datos en forma de vector unidimensional, dicha capa es la encargada de distribuir los datos de entrada, x , a la primera capa oculta multiplicándolos por ciertos pesos, que se suelen denotar por w o θ , y finalmente se puede un *bias*, b . Las operaciones que se llevan a cabo de forma sucesiva en cada neurona de las capas ocultas son procesos estocásticos que harán que cada una de las neuronas perjudique o mejore el resultado final. Esto es lo que se conoce como proceso de aprendizaje de la red. Tener varias capas ocultas apiladas unas sobre otras se conocen comúnmente como aprendizaje profundo [19].

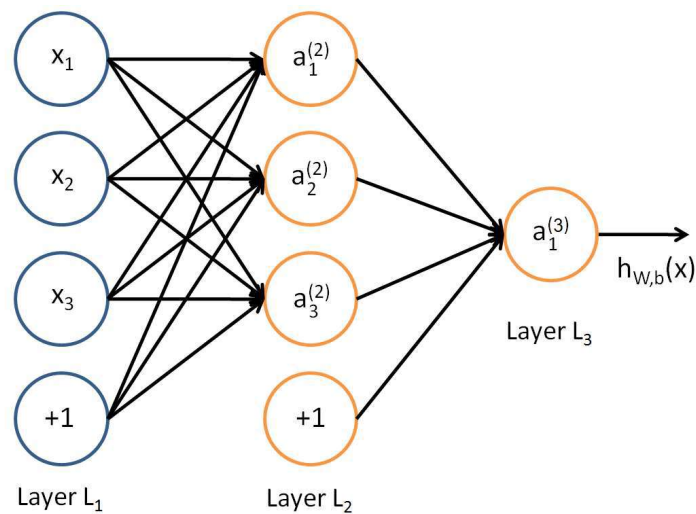


Figura 3.8. Red neuronal simple [20]

ii. Procesos matemáticos de la red neuronal

Para poder entender las ecuaciones matemáticas que tienen lugar en una red neuronal, se va a tomar como ejemplo un el modelo que se presenta en la Figura 3.8, en la que se tienen 4 nodos de entrada (3 + 1 *bias*), una capa oculta con 4 nodos (3 + 1 *bias*) y un nodo de salida.

Los nodos *bias* se van a representar como x^0 y a^0 respectivamente para la capa de entrada y la capa oculta. De este modo, los nodos de entrada pueden ser representarse como un vector x y los nodos de la capa oculta en el vector a .

$$x = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix} \quad (3.10), \quad a = \begin{bmatrix} a_0^{(2)} \\ a_1^{(2)} \\ a_2^{(2)} \\ a_3^{(2)} \end{bmatrix} \quad (3.11)$$

Los pesos (representados como flechas en la Figura 3.8), se suelen denotar como θ o w . En este caso se va a denotar como θ . Los pesos entre la entrada y la capa oculta se representan una matriz de 3×4 . Y los pesos entre la capa oculta y la capa de salida se representan con un vector de 1×4 .

Si la red tiene a unidades en la capa j y b unidades en la capa $j+1$, entonces θ_j será de dimensión $b \times (a + 1)$.

$$\theta^{(1)} = \begin{bmatrix} \theta_{10} & \theta_{11} & \theta_{12} \\ \theta_{20} & \theta_{21} & \theta_{22} \\ \theta_{30} & \theta_{31} & \theta_{32} \end{bmatrix} \quad (3.12)$$

Para calcular los nodos de activación de la capa oculta, se multiplica el vector de entrada x y la matriz de pesos θ^1 para la primera capa ($X * \theta^1$) y luego se aplica la función de activación g . Obteniendo:

$$\begin{aligned} a_1^{(2)} &= g\left(\Theta_{10}^{(1)} x_0 + \Theta_{11}^{(1)} x_1 + \Theta_{12}^{(1)} x_2 + \Theta_{13}^{(1)} x_3\right) \\ a_2^{(2)} &= g\left(\Theta_{20}^{(1)} x_0 + \Theta_{21}^{(1)} x_1 + \Theta_{22}^{(1)} x_2 + \Theta_{23}^{(1)} x_3\right) \\ a_3^{(2)} &= g\left(\Theta_{30}^{(1)} x_0 + \Theta_{31}^{(1)} x_1 + \Theta_{32}^{(1)} x_2 + \Theta_{33}^{(1)} x_3\right) \end{aligned} \quad (3.13)$$

Si se multiplica el vector de la capa oculta con la matriz de pesos θ , para la segunda capa ($a * \theta$) se obtiene la siguiente salida:

$$h_{\theta}(x) = a_1^{(3)} = g\left(\Theta_{10}^{(2)} a_0^{(2)} + \Theta_{11}^{(2)} a_1^{(2)} + \Theta_{12}^{(2)} a_2^{(2)} + \Theta_{13}^{(2)} a_3^{(2)}\right) \quad (3.14)$$

Como se ha comentado, este ejemplo es sólo para una única capa oculta de 4 nodos. Si tratamos de generalizarlo para una red neuronal con múltiples capas ocultas y varios nodos en cada una de las capas, se puede obtener la siguiente fórmula:

$$\mathbf{a}_n^L = \left[\sigma \left(\sum_m \theta_{mm}^L \left[\dots \left[\sigma \left(\sum_j \theta_{kj}^2 \left[\sigma \left(\sum_i \theta_{ji}^1 x_i + b_j^1 \right) \right] + b_k^2 \right) \right] \dots \right] + b_n^L \right) \right]_n \quad (3.15)$$

Donde tenemos L capas con n nodos y $L - 1$ capa con m nodos.

iii. Función de activación

En una red neuronal, la función de activación es la que define si el nodo en estudio debe ser activado o no. Algunas de las funciones de activación más importantes son la sigmoide, la tangente hiperbólica, la unidad lineal rectificada (ReLU) y la función Softmax.

Sigmoide

La función de activación sigmoide también se conoce como función logística. La función de activación sigmoide es una función no lineal utilizada principalmente en redes neuronales *feed-forward*⁹. Se trata de una función real, diferenciable, acotada, definida para valores de entrada reales, con derivadas positivas en todos los puntos[21]. La función Sigmoide viene dada por la relación:

$$f(x) = \left(\frac{1}{1 + e^{-x}} \right) = \frac{e^x}{e^x + 1} \quad (3.16)$$

La función sigmoide puede aparecer en las capas de salida de las arquitecturas de DL, y se utiliza para predecir resultados basados en probabilidades, como problemas de clasificación binaria, modelando tareas de regresión logística y otros dominios de las redes neuronales.

Tangente hiperbólica

La función de activación tangente hiperbólica es otro tipo de las utilizadas en DL. Se simplifica escribiendo *tanh*, se trata de una función centrada en cero, cuyo rango se encuentra entre -1 y 1, por lo tanto, la salida de la función tanh está dada por

$$f(x) = \left(\frac{e^x - e^{-x}}{e^x + e^{-x}} \right) \quad (3.17)$$

La función *tanh* proporciona un mejor rendimiento de entrenamiento para redes neuronales de múltiples capas que la función sigmoide [21]. La principal ventaja que proporciona la función es que produce una salida centrada en cero, lo que ayuda al proceso de *backpropagation*¹⁰.

Una propiedad de la función *tanh* es que solo puede alcanzar un gradiente de 1, solo cuando el valor de la entrada es 0, es decir, cuando x es cero. Esto hace que la función *tanh* produzca algunas neuronas muertas durante el cálculo, que es una condición donde el peso de activación es de gradiente cero. Esta limitación de la función *tanh* se solventa con la función de activación de la unidad lineal rectificada (ReLU) [21].

Rectified Linear Unit (ReLU)

La función de activación de la unidad lineal rectificada (ReLU) fue propuesta por Nair y Hinton 2010, y desde entonces, ha sido la función de activación más utilizada para aplicaciones de aprendizaje profundo con resultados de vanguardia hasta la fecha. Ofrece el mejor rendimiento y generalización

⁹ **Redes neuronales *feed-forward***: ANN donde las conexiones entre los nodos no forman un ciclo. Cosa que no ocurre en las redes neuronales recurrentes.

¹⁰ ***Backpropagation*** es un método de cálculo del gradiente utilizado en algoritmos de aprendizaje supervisado utilizados para entrenar redes neuronales artificiales y reducir el error de la salida.

en el aprendizaje profundo en comparación con otras funciones de activación como son la Sigmoides y Tanh. La ReLU representa una función casi lineal y, por lo tanto, conserva las propiedades de los modelos lineales [21].

La función de activación ReLU viene dada por:

$$f(x) = \max(0, x) = \begin{cases} x_i, & \text{if } x_i \geq 0 \\ 0, & \text{if } x_i < 0 \end{cases} \quad (3.18)$$

Como vemos, esta función rectifica los valores de las entradas inferiores a cero, forzándolos a cero y eliminando el problema de gradiente de fuga existente en otros tipos función de activación.

La principal ventaja de usar las unidades lineales rectificadas en la red neuronal es que garantiza un cómputo más rápido ya que no calcula exponenciales ni divisiones. Otra propiedad de ReLU es que introduce la escasez en las unidades ocultas, ya que reduce los valores entre cero y máximo. Sin embargo, la ReLU tiene la limitación de que se sobreajusta más fácilmente en comparación con la función sigmoide, aunque la técnica del *dropout* reduce el efecto de la sobreajuste de las ReLU y las redes rectificadas mejoraron el rendimiento de las redes neuronales profundas [21].

Softmax

La función *Softmax* es una de las funciones de activación utilizadas en la computación neuronal. Se utiliza para calcular la distribución de probabilidad de un vector de números reales. La función *Softmax* produce una salida que es un rango de valores entre 0 y 1, con una suma de probabilidades igual a 1. La función *Softmax* se calcula utilizando la relación:

$$f(x_i) = \frac{e(x_i)}{\sum_j e(x_j)} \quad (3.19)$$

La función *Softmax* se usa en modelos de varias clases donde devuelve las probabilidades de cada clase, con la clase objetivo con la mayor probabilidad. La función *Softmax* aparece principalmente en casi todas las capas de salida de las arquitecturas de aprendizaje profundo, donde se usan. Como es el caso de las redes neuronales creadas en este trabajo [21].

iv. Nodo bias

El uso del nodo *bias* suele ser necesario para crear un modelo de aprendizaje exitoso. Si se suma el valor de *bias* se consigue mover la función de activación a izquierda o derecha en el eje, y si se multiplica se puede cambiar la inclinación de dicha función, ayudando a conseguir un mejor ajuste para los datos.

v. Función de coste

La función general de coste representa la suma del error, diferencia entre el valor predicho y el valor real (etiquetado).

$$J(\theta) = \frac{1}{m} \sum_{i=1}^m \text{Cost}(h_{\theta}(x^{(i)}), y^{(i)}) \quad (3.20)$$

El resultado de un problema de clasificación sólo puede tomar valores discretos $\{0,1\}$. Dicho de otro modo, la salida sólo puede pertenecer a un tipo de clase. En nuestro caso, si se pretende clasificar las imágenes histológicas tendremos núcleo (clase 1), estroma (clase 2) y fondo (clase 3). Si la imagen de entrada es un núcleo, las clases en la salida tomarán valor 1 para la clase núcleo y el valor 0 para las otras clases. Esto significa que el valor de la clase, la hipótesis a satisfacer debe tomar valores entre 0 y 1.

$$0 \leq h_{\theta}(x) \leq 1 \quad (3.21)$$

Por tanto, la hipótesis es:

$$h_{\theta}(x) = g(\theta^T x) \quad (3.22)$$

Supongamos que g es una función de activación que tiene valores en el rango $[0,1]$.

El objetivo es optimizar la función de coste, por lo que se debe encontrar $\min J(\theta)$. No obstante, puede que al aplicar la función de activación el resultado sea una función no convexa, lo que significa que puede haber múltiples mínimos locales. Por tanto, no se puede garantizar converger al mínimo global. Se necesita, por tanto, una función convexa, sin mínimos locales, para poder encontrar el mínimo global, es decir, es necesario minimizar $J(\theta)$. Para hacer esto, se toman logaritmos en la función de coste:

$$\text{cost}(h_{\theta}(x), y) = \begin{cases} -\log(h_{\theta}(x)) & \text{if } y = 1 \\ -\log(1 - h_{\theta}(x)) & \text{if } y = 0 \end{cases} \quad (3.23)$$

En caso de que el valor etiquetado y sea igual a 1, la hipótesis puede ser $-\log(h(x))$ o, si está invertido, $-\log(1 - h(x))$.

Tomando como referencia la ecuación (3.23), los valores donde $y = 1$ se registrarán por $-\log(h(x))$, esta función se ilustra en la Figura 3.9. (a). Cabe destacar que, de los valores que puede tomar la función \log , interesa únicamente el rango $[0,1]$, ya que la hipótesis (3.21) sólo puede tomar valores en ese rango.

Además, observando la Figura 3.9. (a), donde $y = 1$, y $h(x)$ se acerca al valor de 1 (eje x), el coste se aproxima al valor 0 ($h(x) - y$ será 0). De lo contrario, si $h(x)$ se aproxima a 0, la función de costo va al infinito, lo que significa que tiene un coste muy grande.

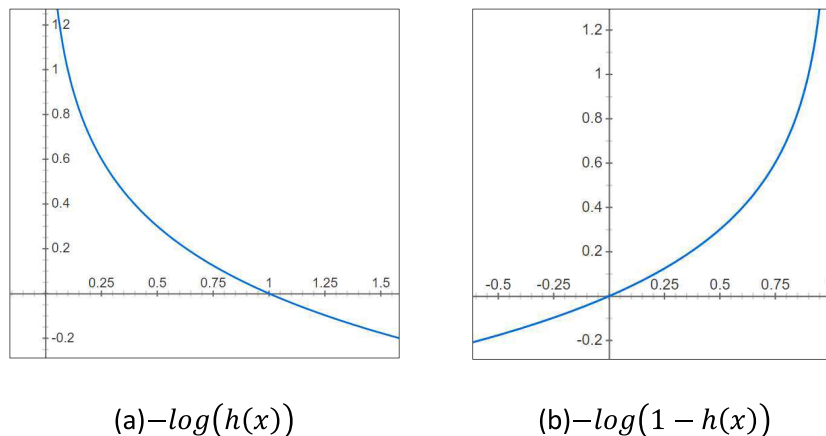


Figura 3.9. Funciones logarítmicas

Por otro lado, cuando $y = 0$, la función de coste es $-\log(1 - h(x))$, que puede verse ilustrada en la Figura 3.9. (b). Puede verse que, si $h(x)$ se acerca al valor 0, el coste se acercaría a 0.

Dado que y (valor etiquetado) siempre es 0 o 1, la función de coste puede reescribirse como

$$\text{Cost}(h_{\theta}(x), y) = -y \log(h_{\theta}(x)) - (1 - y) \log(1 - h_{\theta}(x)) \quad (3.24)$$

Reescribiendo la ecuación de costes completa (3.20) con (3.24) se tiene:

$$J(\theta) = -\frac{1}{m} \sum_{i=1}^m \left[y^{(i)} \log(h_{\theta}(x^{(i)})) + (1 - y^{(i)}) \log(1 - h_{\theta}(x^{(i)})) \right] + \frac{\lambda}{2m} \sum_{j=1}^n \theta_j^2 \quad (3.25)$$

No obstante, esta función de coste, (3.25), es para el caso en el que solo hay un nodo en la capa de salida de la red neuronal. Si se generaliza para múltiples nodos de salida (clasificación multiclase), lo que se obtiene es:

$$J(\theta) = -\frac{1}{m} \sum_{i=1}^m \sum_{k=1}^K \left[y_k^{(i)} \log((h_{\theta}(x^{(i)}))_k) + (1 - y_k^{(i)}) \log(1 - (h_{\theta}(x^{(i)}))_k) \right] + \frac{\lambda}{2m} \sum_{l=1}^{L-1} \sum_{i=1}^{s_{l+1}} (\theta_{j,i}^{(l)})^2 \quad (3.26)$$

vi. Forward Propagation

Este proceso *forward propagation* es el encargado de que se pueda obtener un valor de salida de la red neuronal al introducir una entrada. Este algoritmo se usa para calcular el valor del coste. Hace es el mismo proceso matemático que el descrito anteriormente (Procesos matemáticos de la red neuronal). Efectuando las ecuaciones pertinentes se obtiene el valor de hipótesis (3.14).

Tras obtener el valor $h(x)$ (hipótesis), se emplea la ecuación de la función Coste (3.26) para calcular el coste del conjunto de entradas dado y generar así las predicciones. Los pasos a seguir son los siguientes:

$$x = a^{(1)} \quad (3.27)$$

$$z^{(j+1)} = \theta^{(j)} a^{(j)} \quad (3.28)$$

$$a^{(j+1)} = \sigma(z^{(j+1)}) \quad (3.29)$$

$$h_{\theta}(x) = a^{(L)} = \sigma(z^{(L)}) \quad (3.30)$$

vii. Backpropagation

El objetivo del *backpropagation* es minimizar la función de coste $J(\theta)$, esto se consigue utilizando el conjunto óptimo de valores θ (pesos). El método *backpropagation* calcula la derivada parcial de $J(\theta)$ y, posteriormente, este valor se utiliza en el algoritmo de descenso de gradiente (*Algoritmo 1*) para calcular los valores de θ de la red neuronal que minimizan la función de coste $J(\theta)$.

Algoritmo 1. Descenso de Gradiente

1: **Repetir** {
 2: $\theta_j := \theta_j - \alpha^{11} \frac{\partial}{\partial \theta_j} J(\theta)$ }
 3: **hasta que** $\frac{\partial}{\partial \theta_j} J(\theta) \rightarrow 0$

El algoritmo *backpropagation* tiene 5 pasos:

Algoritmo 2. Backpropagation

1: Establecer $a(1) = x$; para los datos de entrenamiento
 2: Realizar la *forward propagation* y calcular $a(l)$ para las otras capas ($l = 2 \dots L$)
 3: Tomar el valor y (**etiquetado**) y calcular el error como derivada parcial de la última capa
 4: $\delta(L) = h(x) - y$
 5: Calcular los valores de $\delta(l)$ hacia atrás para cada capa mediante el **Algoritmo 1**
 6: Calcular los valores de derivada $\delta(l + 1)$ para cada capa, que representan la derivada
 7: del coste $J(\theta)$ con respecto a $\theta(l)$ para la capa l

Por tanto, *backpropagation*, propaga el error hacia atrás en la red neuronal, para poder analizar, en el camino de regreso cuánto contribuye cada peso en el error general. Los pesos que más contribuyen al error general tendrán valores de derivación mayores, lo que significa que modificarán más su valor (al calcular el descenso de Gradiente) [20].

viii. Hiperparámetros

Batch

El *batch* es un hiperparámetro que define el número de muestras con las que se va a trabajar antes de actualizar los parámetros internos del modelo.

Este hiperparámetro se encarga de ir iterando sobre una o más muestras para hacer predicciones. Al final del *batch*, las predicciones se comparan con las variables de salida esperadas y se calcula un error. A partir de este error, el algoritmo de actualización se usa para mejorar el modelo.

El conjunto de datos de entrenamiento se puede dividir en uno o más *batches*.

¹¹ α en esta ecuación representa el *learning rate*, se trata de un hiperparámetro

Cuando todas las muestras de entrenamiento se utilizan para crear un *batch*, el algoritmo de aprendizaje se denomina descenso de gradiente por *batches*. Cuando el *batch* es del tamaño de una muestra, el algoritmo de aprendizaje se denomina descenso de gradiente estocástico. Cuando el tamaño del *batch* es más de una muestra y menor que el tamaño del conjunto de datos de entrenamiento, el algoritmo de aprendizaje se denomina descenso de gradiente de mini – *batches*.

En el caso del descenso por gradiente de mini – *batches*, los tamaños de *batches* populares son de 32, 64 y 128 muestras [22].

Épocas

El número de épocas es un hiperparámetro que define el número de veces que el algoritmo de aprendizaje se ejecutará con el conjunto de datos de entrenamiento.

Una época significa que cada muestra en el conjunto de datos de entrenamiento ha tenido la oportunidad de actualizar los parámetros internos del modelo. Por tanto, una época puede estar compuesta de uno o más *batches*.

El número de épocas suele ser grande, lo que permite que el algoritmo de aprendizaje ejecutarse hasta que el error del modelo se haya minimizado lo suficiente.

En el caso concreto de este proyecto, la red neuronal se ha ejecutado a 30 épocas en la fase de desarrollo de la misma red, y una vez se ha conseguido la arquitectura deseada se ha ejecutado durante 80 épocas.

Es común crear gráficos de líneas que muestren épocas a lo largo del *eje x* como el tiempo y el error o habilidad del modelo en el *eje y*. Estas gráficas suelen conocerse como curvas de aprendizaje. Gracias a ellas se puede diagnosticar si el modelo no ha funcionado, ha aprendido poco o si se ajusta adecuadamente al conjunto de datos de capacitación. En el apartado de resultados se mostrarán las gráficas generadas de las redes neuronales convolucionales propuestas en este trabajo [22].

Learning rate

El *learning rate*, α , es un hiperparámetro que controla cuánto cambia el modelo con respecto al error estimado cada vez que se actualizan los pesos mediante la función de descenso de gradiente (*Algoritmo 1*). Elegir el valor óptimo de *learning rate* es complejo, ya que un valor demasiado pequeño puede hacer que, a largo proceso, el entrenamiento se atascarse (al converger en un mínimo local), mientras que un valor demasiado grande puede hacer que el aprendizaje no sea correcto, consiguiendo un conjunto de pesos que hacen que el resultado no se ajuste al deseado, o haciendo que el proceso de entrenamiento sea inestable.

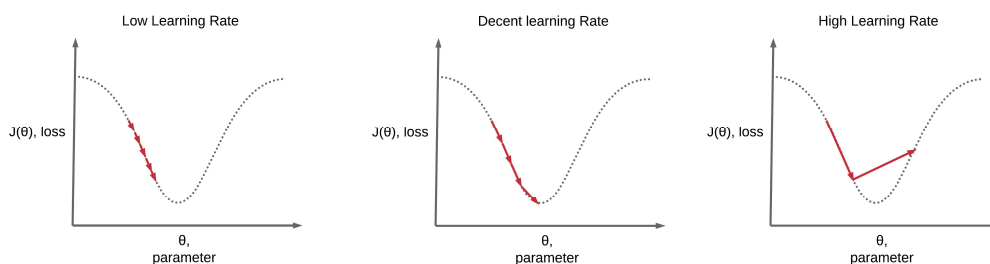


Figura 3.10. Ejemplos de distintos valores de learning rate [23]

El *learning rate* puede ser el hiperparámetro más importante a la hora de configurar la red neuronal. Por lo tanto, es importante investigar los efectos del *learning rate* en el rendimiento del modelo y tener una intuición sobre la dinámica del *learning rate* en el comportamiento del modelo [24].

ix. Overfitting

Es común pensar que podría aumentarse la cantidad de capas ocultas en nuestra red y también aumentar la cantidad de neuronas dentro de ellas con el fin de mejorar la respuesta de la red. No obstante, no es tan sencillo. La principal razón es que no se tiene poder computacional ilimitado ni tiempo infinito para entrenar estas enormes ANN.

Por otro lado, se pretende detener o reducir los efectos del sobreajuste. El sobreajuste es básicamente cuando una red no puede aprender de manera efectiva debido a una serie de razones. Es un concepto importante en la mayoría, si no en todos los algoritmos de aprendizaje automático. Es importante que se tomen todas las precauciones para reducir sus efectos. Si un modelo exhibiera signos de sobreajuste, podría verse una capacidad reducida para identificar características generalizadas no solo para el conjunto de datos de entrenamiento, sino también para los conjuntos de validación y test.

Esta es la razón principal para reducir la complejidad de nuestros ANN. Cuantos menos parámetros se requieran para entrenar, es menos probable que la red se ajuste en exceso y, por supuesto, mejore el rendimiento predictivo del modelo.

x. Dropout

El *dropout* es una forma de regularización que tiene como objetivo ayudar a prevenir el sobreajuste aumentando la precisión en los conjuntos de validación y test, tal vez, a costa de una pérdida de precisión en el entrenamiento. Para cada *mini-batch* en nuestro conjunto de entrenamiento, las capas de *dropout*, con probabilidad p , desconectan aleatoriamente las entradas de la capa anterior a la siguiente capa en la arquitectura de red.

La Figura 3.11 se visualiza este concepto, en este caso se desconecta aleatoriamente con probabilidad $p = 0.5$ las conexiones entre dos capas *fully connected* para un *mini-batch* dado. Nuevamente, observe cómo se corta la mitad de las conexiones para este *mini batch*.

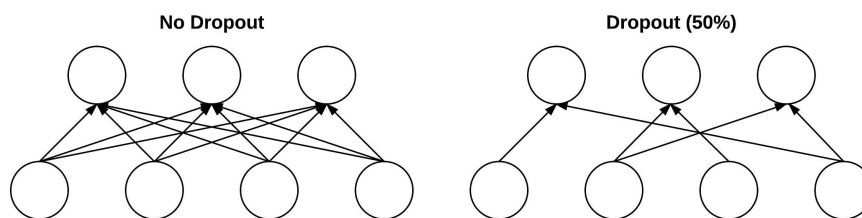


Figura 3.11. Dropout [25]

La razón por la que se utiliza el *dropout* es para reducir el sobreajuste alterando explícitamente la arquitectura de red en el tiempo de entrenamiento. La desconexión aleatoria de conexiones asegura que ningún nodo único en la red sea responsable de "activarse" cuando se le presente un patrón dado. En cambio, el *dropout* asegura que haya múltiples nodos redundantes que se activarán cuando se presenten entradas similares; esto a su vez ayuda a nuestro modelo a generalizar [25].

xi. Redes neuronales convolucionales

Las redes neuronales convolucionales (CNN) son análogas a las ANN tradicionales en lo referente a que están formadas por neuronas cuyos pesos se auto-optimizan a través del aprendizaje. Cada neurona recibirá una entrada y realizará una operación (como un producto escalar seguido de una función no lineal).

La única diferencia notable entre las CNN y las ANN tradicionales es que las CNN se utilizan principalmente en el campo del reconocimiento de patrones dentro de las imágenes. Esto nos permite codificar características específicas de la imagen en la arquitectura, haciendo que la red sea más adecuada para tareas centradas en la imagen, al tiempo que reduce aún más los parámetros necesarios para configurar el modelo.

Una de las mayores limitaciones de las formas tradicionales de ANN es que tienden a luchar con la complejidad computacional requerida para calcular los datos de la imagen. Los conjuntos de datos de evaluación comparativa de aprendizaje automático comunes, como la base de datos MNIST de dígitos escritos a mano, son adecuados para la mayoría de las formas de ANN, debido a su dimensionalidad de imagen relativamente pequeña de solo 28×28 . Con este conjunto de datos, una sola neurona en la primera capa oculta contendrá 784 pesos ($28 \times 28 \times 1$ donde 1 recuerda que MNIST está normalizado a valores en blanco y negro), lo cual es manejable para la mayoría de las formas de ANN.

Si se considera una entrada de imagen en color de un tamaño de 64×64 como es nuestro caso (tamaño que van a tener los parches de las imágenes de histología cuando entran en la red) el número de pesos en una sola neurona de la primera capa aumenta sustancialmente a 12.288.

3.2.5. Arquitectura Red neuronal convolucional

Como se señaló anteriormente, las CNN se centran, principalmente, en que la entrada a la misma será un conjunto de imágenes. Esto hace que la arquitectura se configure de una manera distinta, con el fin de que se adapte mejor a la necesidad de tratar con el tipo específico de datos.

Una de las diferencias clave es que las neuronas que forman las capas dentro de la CNN están formadas por neuronas organizadas en tres dimensiones, la dimensión espacial de la entrada (altura y ancho) y la profundidad. La profundidad no se refiere al número total de capas dentro del ANN, sino a la tercera dimensión de un volumen de activación. A diferencia de las ANNs estándar, las neuronas dentro de cualquier capa dada solo se conectarán a una pequeña región de la capa que la precede.

En la práctica, esto significará que, para nuestro proyecto, el volumen de entrada tendrá una dimensionalidad de $64 \times 64 \times 3$ (altura, ancho y profundidad), lo que conducirá a una capa de salida final compuesta por una dimensionalidad de $1 \times 1 \times n$ (donde n representa el número posible de clases) ya que habríamos condensado la dimensionalidad de entrada completa en un volumen más pequeño de puntajes de clase a través de la dimensión de profundidad.

i. Arquitectura general

Las CNN se componen de tres tipos de capas: capas convolucionales, capas de pooling y capas *fully – connected*. Cuando estas capas se apilan, se forma una arquitectura CNN. En la Figura 3.12 se ilustra una arquitectura CNN para la clasificación de una imagen de entrada como núcleo, estroma, y fondo.

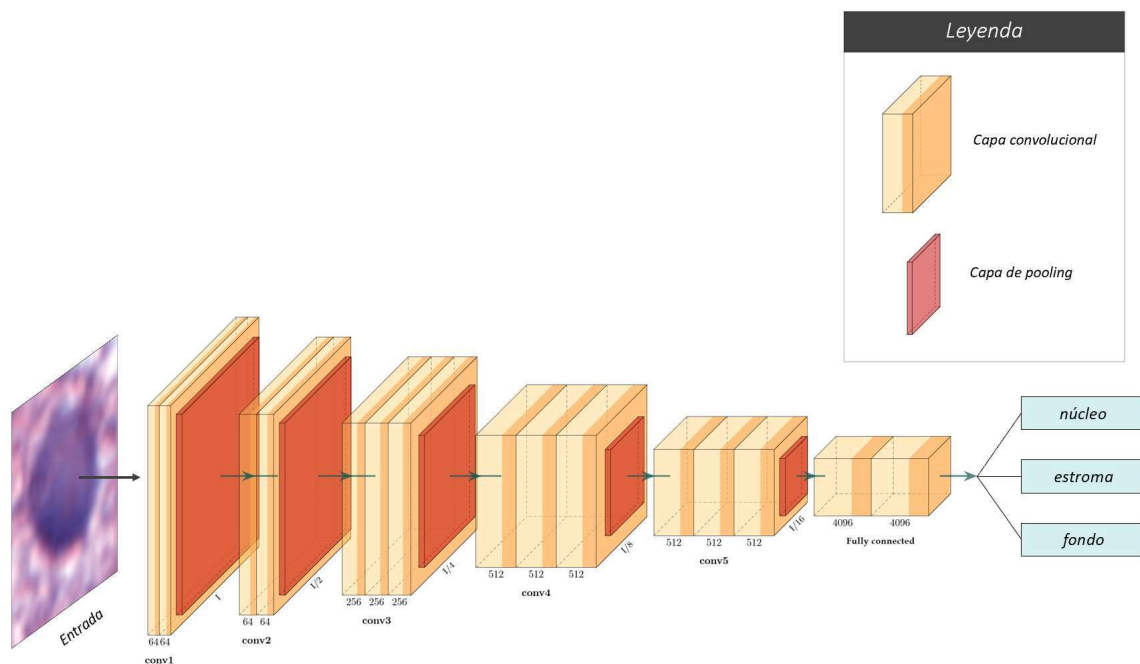


Figura 3.12. Red neuronal convolucional
(Diagrama generado con PlotNeuralNet-master [26])

La funcionalidad básica del ejemplo CNN anterior se puede dividir en cuatro áreas clave.

1. En las CNNs, la capa de entrada contendrá los valores de los píxeles de la imagen.
2. La capa convolucional determinará la salida de las neuronas que están conectadas a regiones locales de la entrada a través del cálculo del producto escalar entre sus pesos y la región conectada al volumen de entrada. La unidad lineal rectificadora (comúnmente acortada a ReLu) tiene como objetivo aplicar una función de activación por elementos, como sigmoide, a la salida de la activación producida por la capa anterior.
3. La capa de *pooling* simplemente realizará un muestreo descendente a lo largo de la dimensionalidad espacial de la entrada dada, reduciendo aún más el número de parámetros dentro de esa activación.
4. Las capas *fully connected* realizarán las mismas funciones que en las ANN estándar e intentarán producir puntuaciones de clase a partir de las activaciones, para ser utilizados para la clasificación. También se sugiere que ReLu se pueda usar entre estas capas, para mejorar el rendimiento.

A través de este sencillo método de transformación, las CNN pueden transformar la entrada original capa por capa utilizando técnicas convolucionales y de muestreo para producir probabilidades de clase con fines de clasificación y regresión [19].

ii. Capa convolucional

Como su nombre indica, la capa convolucional juega un papel vital en el funcionamiento de las CNN. Los parámetros de las capas se centran en el uso de *kernels*.

Estos *kernels* son generalmente pequeños en dimensionalidad espacial, pero se extienden a lo largo de toda la profundidad de la entrada. Cuando los datos llegan a una capa convolucional, la capa hace que cada filtro atraviese la dimensionalidad espacial de la entrada para producir un mapa de activación 2D.

A medida que nos deslizamos a través de la entrada, el producto escalar se calcula para cada valor en ese *kernel*, en la Figura 3.13 vemos como un *kernel* de 3×3 se desplaza a lo largo de la entrada produciendo una salida que tiene unas dimensiones de 2×2 . A partir de esto, la red aprenderá *kernels* que se activarán cuando vean una característica específica en una posición espacial dada de la entrada. Estos se conocen comúnmente como activaciones.

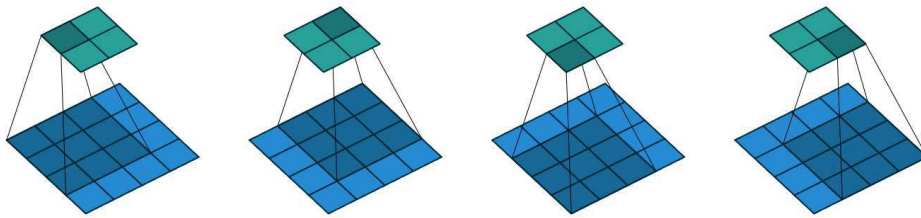


Figura 3.13. Representación de una capa de convolución [27]

Cada *kernel* dará lugar a un mapa de activación, el cual se apilará a lo largo de la dimensión de profundidad para formar el volumen de salida completo de la capa convolucional.

Como se ha mencionado anteriormente, el entrenamiento de ANNs donde las entradas son imágenes conlleva modelos que son demasiado grandes para entrenar. Esto ocurre en las neuronas ANN estándar por estar fully – connected. Por lo que para mitigar esto, cada neurona en una capa convolucional solo estará conectada a una pequeña región del volumen de entrada. La dimensionalidad de esta región se conoce comúnmente como el tamaño de campo receptivo de la neurona. La magnitud de la conectividad a través de la profundidad es casi siempre igual a la profundidad de la entrada.

En nuestro caso, si la entrada a la red es una imagen de tamaño $64 \times 64 \times 3$ (una imagen de color RGB con una dimensionalidad de 64×64) y establecemos el tamaño del campo receptivo como 6×6 , tendríamos un total de 108 pesos en cada neurona dentro de la capa convolucional. ($6 \times 6 \times 3$ donde 3 es la magnitud de la conectividad en la profundidad del volumen). Para poner esto en perspectiva, una neurona estándar vista en otras formas de ANN contendría 12.288 pesos cada una.

Las capas convolucionales también pueden reducir significativamente la complejidad del modelo a través de la optimización de su salida. Estos se optimizan a través de tres hiperparámetros, la profundidad, el *stride* y el ajuste de *zero-padding*.

La **profundidad** del volumen de salida producido por las capas convolucionales se puede establecer manualmente a través de la cantidad de neuronas dentro de la capa en la misma región de la entrada. Esto se puede ver con otras formas de ANN, donde todas las neuronas en la capa oculta están

conectadas directamente a cada neurona de antemano. La reducción de este hiperparámetro puede minimizar significativamente el número total de neuronas de la red, pero también puede reducir significativamente las capacidades de reconocimiento de patrones del modelo.

También es posible definir el **stride**, donde se establece la profundidad alrededor de la dimensionalidad espacial de la entrada para ubicar el campo receptivo. Por ejemplo, si estableciéramos un paso como 1, como en la Figura 3.14, se tiene un campo receptivo muy superpuesto que produciría activaciones extremadamente grandes. Alternativamente, establecer *strides* en un número mayor reducirá la cantidad de superposición y producirá una salida de dimensiones espaciales más bajas.

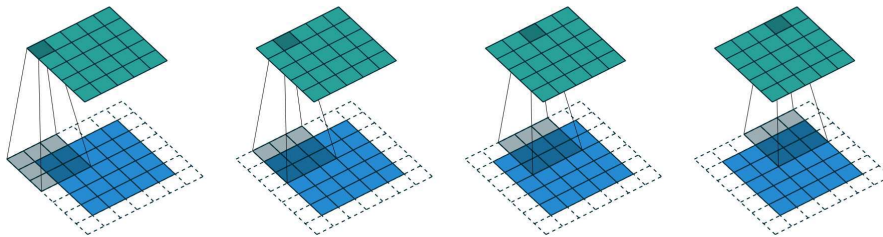


Figura 3.14. Convolución de kernel 3x3 con stride = 1 y zero padding [27]

El *zero – padding* es el proceso simple de rellenar el borde de la entrada, y es un método efectivo para dar un mayor control en cuanto a la dimensionalidad de los volúmenes de salida, ya que nos permite no disminuir la dimensionalidad de la imagen de origen, ya que el *kernel* puede centrarse en todos los píxeles, como se ve en la Figura 3.14.

Es importante entender que, mediante el uso de estas técnicas, es posible modificar la dimensionalidad espacial de la salida de las capas convolucionales, por eso hay que prestar mucha atención cuando se diseña e implementa la red neuronal [19].

iii. Convolución transpuesta (Upsampling)

La capa de deconvolución o convolución transpuesta es el camino contrario a la convolución, es decir, mapear desde un espacio de menos dimensiones a un espacio con mayor número de dimensiones, manteniendo el patrón de conectividad de la convolución representada en la Figura 3.13.

Las convoluciones transpuestas intercambian los pasos hacia adelante y hacia atrás de una convolución. Una forma de ilustrar este concepto es como se relaciona el *kernel* que define la convolución. Cuando se realiza la convolución directa el *forward propagation* y *backpropagation* se calculan a partir de multiplicar el *kernel* por la matriz de convolución C , en concreto, el *forward* se calcula multiplicando por C y el *backpropagation* multiplicando por C^T . Atendiendo a estas operaciones, es posible definir la convolución transpuesta, cuyos *forwards* y *backpropagation* se calculen multiplicando con C^T y $(C^T)^T = C$, respectivamente. Esta idea se ilustra en la Figura 3.15

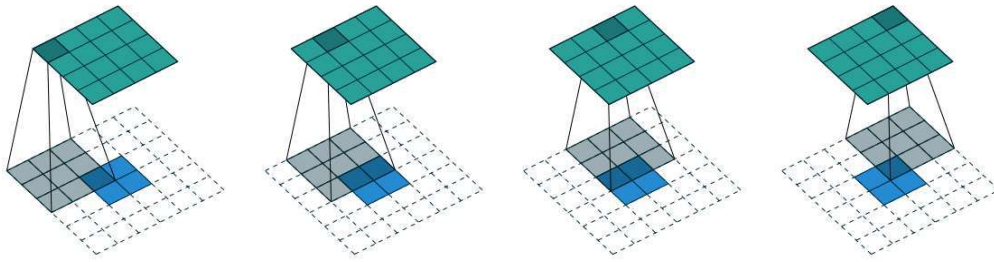


Figura 3.15. Deconvolución con un kernel de 3x3, stride de 1 y obteniendo una salida de 4x4 [27]

iv. Capas de pooling

Las capas de *pooling* tienen como objetivo reducir gradualmente la dimensionalidad de la representación, como ocurre en la Figura 3.13. Eso se hace para reducir, aún más, el número de parámetros y la complejidad computacional del modelo.

La capa de agrupación opera sobre cada mapa de activación en la entrada, y escala su dimensionalidad. En la mayoría de las CNN, estos vienen en forma de capas de agrupación máxima con núcleos de una dimensionalidad de 2×2 aplicados con un *stride* de 2 a lo largo de las dimensiones espaciales de la entrada. Esto reduce el mapa de activación a un 25% del tamaño original, manteniendo el volumen de profundidad a su tamaño estándar.

Debido a la naturaleza destructiva de la capa de *pooling*, solo hay dos métodos generalmente observados de agrupación máxima. Por lo general, la *stride* y los filtros de las capas de agrupación se establecen en 2×2 , lo que permitirá que la capa se extienda a través de la totalidad de la dimensionalidad espacial de la entrada. Además, se puede utilizar la agrupación superpuesta, donde el *stride* se establece en 2 con un tamaño de *kernel* establecido en 3. Debido a la naturaleza destructiva del *pooling*, tener un tamaño de *kernel* superior a 3 generalmente disminuirá en gran medida el rendimiento del modelo.

v. Batch normalization

Durante el entrenamiento de la red, la distribución de parámetros ocultos en una capa puede moverse gradualmente, esto se conoce como cambio covariable interno. De hecho, este es el punto que queremos que cambien los parámetros para aprender patrones sobre los datos subyacentes. Pero a medida que cambia la distribución de los pesos en una capa, las entradas a la siguiente capa pueden desviarse de una distribución ideal. Con el *batch normalization* la norma de *batch* toma la salida de activaciones de la capa anterior y resta la media del *batch* y la divide entre la desviación estándar del *batch*. Esto actúa para volver a centrar la distribución de los valores a con una media de 0 y una desviación estándar de 1 (Figura 3.16).

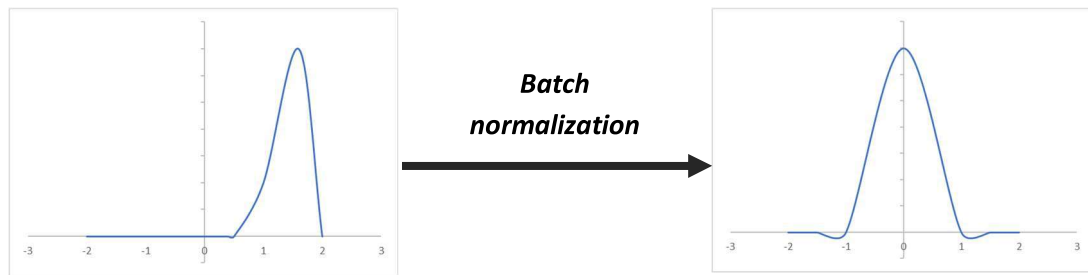


Figura 3.16. Batch normalization

Por lo tanto, si hay valores extremos en la capa anterior, no causarán degradados que desaparezcan en la siguiente capa. El *batch normalization* tiene las siguientes ventajas:

- Permite que las capas aprendan más independientemente de entre sí, ya que los valores grandes en una capa no influirán negativamente en la siguiente capa.
- Permite la selección de una tasa de aprendizaje más alta porque no hay valores extremos en las activaciones normalizadas, lo que permite un aprendizaje más rápido.
- Las salidas de capa se normalizan a la media del *batch* y la desviación estándar, lo que agrega un elemento de ruido (especialmente con *batches* más pequeños) que, a su vez, contribuye a la regularización.

3.2.6. Arquitectura autoencoder (*encoder – decoder*)

Un *autoencoder* de imágenes clásico toma una imagen, la asigna a un espacio vectorial latente a través de un módulo codificador y luego la decodifica a una salida con las mismas dimensiones que la imagen original, a través de un módulo decodificador, Figura 3.17. Luego se entrena utilizando como datos de destino las mismas imágenes que las imágenes de entrada, lo que significa que el codificador automático aprende a reconstruir las entradas originales. Al imponer varias restricciones en el código (la salida del codificador), puede hacer que el codificador automático aprenda representaciones latentes de los datos más o menos interesantes. Con mayor frecuencia, restringirá el código para que sea de baja dimensión y disperso (principalmente ceros), en cuyo caso el codificador actúa como una forma de comprimir los datos de entrada en menos bits de información.

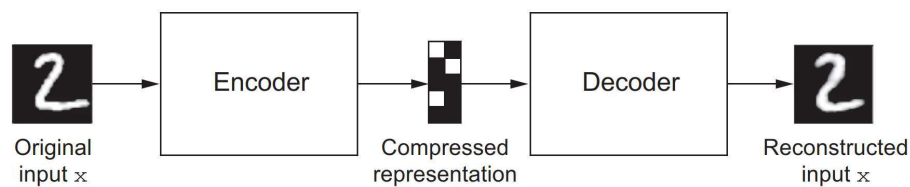


Figura 3.17. Ejemplo autoencoder

Por tanto, el *encoder* aprende la representación compacta de los datos de entrada y el *decoder* descomprime para reconstruir los datos de entrada.

Esto es muy útil en el proyecto que estamos desarrollando ya que podemos asociar una imagen de entrada con su máscara segmentada del mismo tamaño. El tipo de *encoder – decoder* usado es una U-Net.

3.2.7. UNet

La red UNET fue desarrollada por Olaf Ronneberger con el fin de segmentar imágenes biomédicas. Como se ha comentado, la arquitectura contiene dos partes claramente diferenciadas, el *encoder* que se utiliza para capturar el contexto en la imagen. El codificador es solo una pila tradicional de capas convolucionales y de agrupación máxima. La segunda ruta es la ruta de expansión simétrica, es decir, el *decoder*, que se utiliza para permitir la localización precisa mediante convoluciones transpuestas. Por lo tanto, es una red totalmente convolucional de extremo a extremo (FCN), es decir, solo contiene capas convolucionales y no contiene ninguna capa densa.

vi. Skip connections y Bloques residuales

Cuanto mayor es la profundidad de una red, aumenta el riesgo de degradación por lo que la precisión se satura. Dicha degradación no es causada por el sobreajuste, pero la adición de más capas a un modelo adecuadamente profundo conduce a un mayor error de entrenamiento.

La degradación (de la precisión del entrenamiento) indica que no todos los sistemas son igualmente fáciles de optimizar. En la literatura [14] encontramos que una arquitectura menos profunda conseguía mejores resultados en comparación a una más profunda.

Se puede abordar el problema de la degradación mediante la introducción de un marco de aprendizaje residual profundo. En lugar de esperar que pocas capas apiladas encajen directamente en un mapeo subyacente deseado, se obliga a que estas capas se ajusten a un mapeo residual. Formalmente, denotando el mapeo subyacente deseado como $H(x)$, dejamos que las capas no lineales apiladas se ajusten a otro mapeo de $F(x) := H(x) - x$. El mapeo original se vuelve a convertir en $F(x) + x$. Se considera que es más fácil optimizar el mapeo residual que optimizar el mapeo original sin referencia. En el extremo, si un mapeo de identidad fuera óptimo, sería más fácil llevar el residual a cero que ajustar un mapeo de identidad mediante una pila de capas no lineales.

La formulación de $F(x) + x$ puede realizarse mediante redes neuronales de alimentación hacia adelante con "conexiones de acceso directo" (Figura 3.18). Las conexiones de acceso directo son aquellas que omiten una o más capas. En nuestro caso, las conexiones de acceso directo simplemente realizan un mapeo de identidad, y sus salidas se agregan a las salidas de las capas apiladas (Figura 3.18). Las conexiones de acceso directo de identidad no agregan parámetros adicionales ni complejidad computacional.

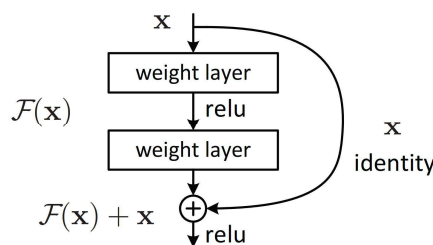


Figura 3.18. Bloque residual [14]

En nuestro caso, las *skip connections* presentes en la UNet funcionan como en la Figura 3.19 donde las capas convolucionales van reduciendo la dimensionalidad de la imagen hasta llegar a la última capa de donde se obtiene una característica y comienza la deconvolución. Las *skip connection* son las

operaciones de adición que se hacen a las distintas capas de la convolución inversa mediante el uso de los bloques convolucionales de la misma dimensión del *downshape* (camino hacia abajo). De esta forma se realimenta el *upshape* (camino hacia arriba) y se puede llegar a obtener un mejor resultado en la predicción.

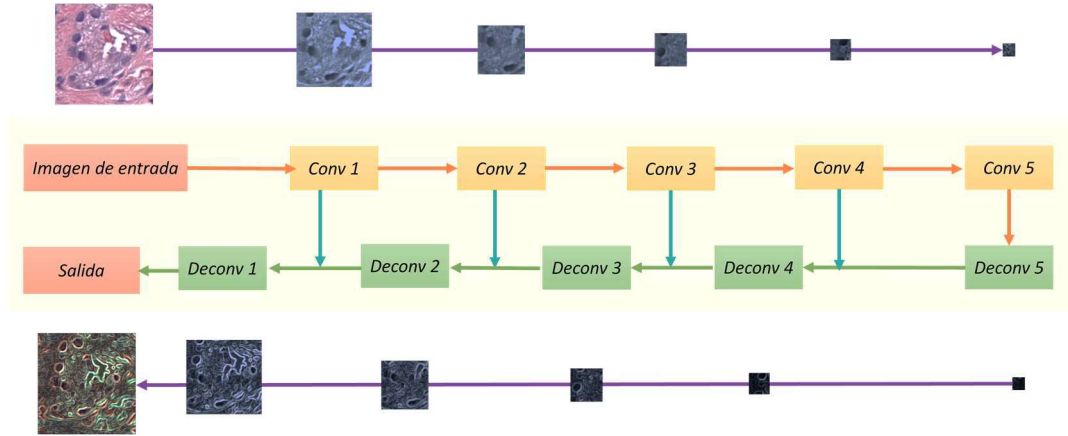


Figura 3.19. Skip connections

3.2.8. Desarrollo de la red UNet

Una vez explicado cómo funcionan las *encoder – decoder*, vamos a exponer la arquitectura que se ha creado para la tarea de segmentación automática de núcleos. Para hacerlo más visual se ha hecho el siguiente diagrama, Figura 3.20.

La red neuronal convolucional UNet consta de 5 bloques convolucionales en el *downsample* compuestos por 64, 128, 256, 512 y 1024 filtros respectivamente, todos ellos de *kernel* 3x3, activación *relu* y *zero padding*, seguido de *batch normalization* y un *MaxPooling* de 2x2. El proceso de entrenamiento se ha llevado a cabo con las 30 imágenes reservadas para training durante 80 épocas y validación con 6 imágenes.

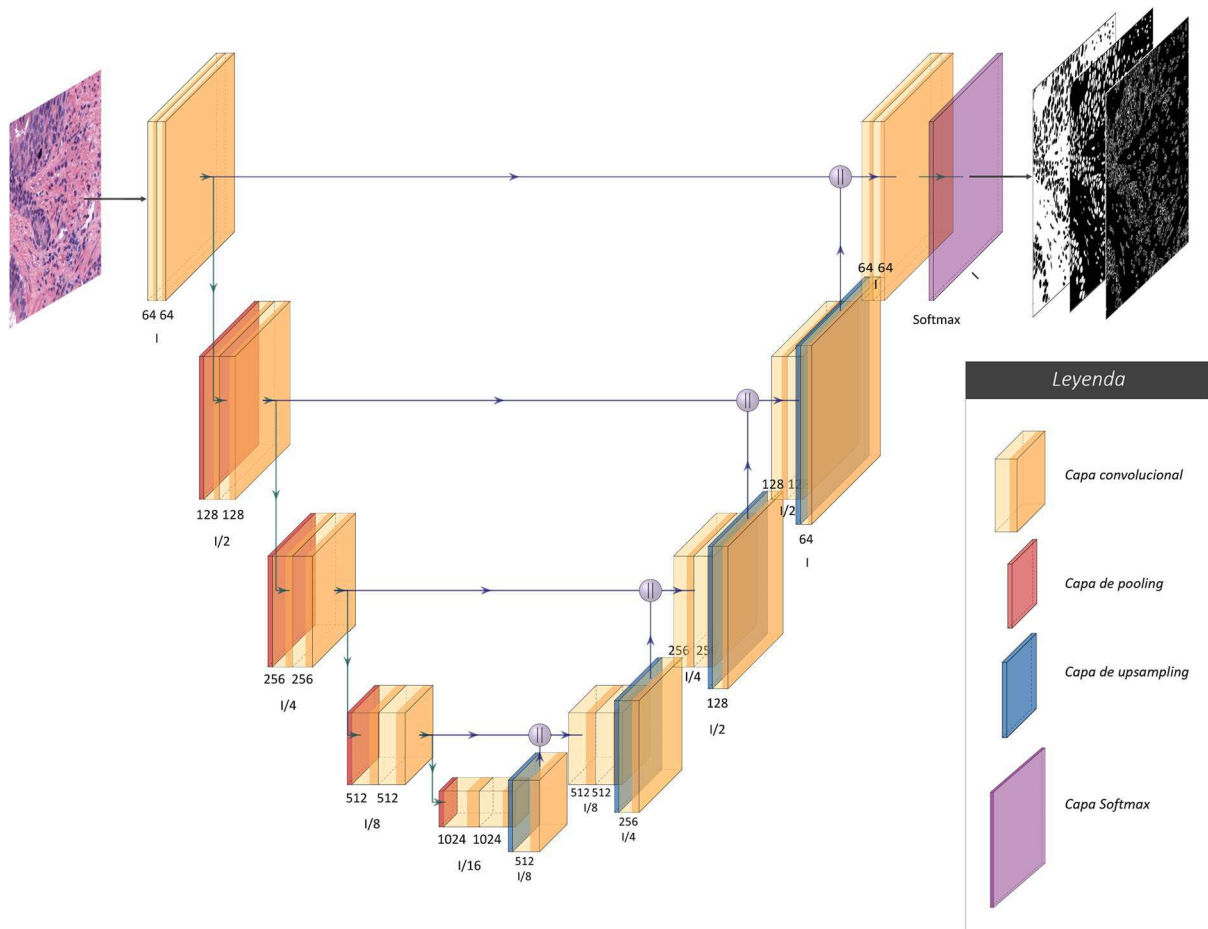


Figura 3.20. UNet del Proyecto
(Diagrama generado con PlotNeuralNet-master [26])

En los bloques convolucionales de 512 y de 1024 se añade una capa de *dropout* de 0.5. Tras esto empieza la parte del *decoder* donde se aumenta la dimensionalidad con un *upsampling* de 2x2 se hace la convolución correspondiente mediante *kernels* de 3x3, *batch normalization* y se concatena la información del mismo nivel de empaquetamiento de la etapa del *encoder*. Una vez concatenados se hacen dos convoluciones más y otro *batch normalization*, y así sucesivamente hasta llegar al último nivel. Finalmente, mediante la función de activación *softmax*, se obtiene el mapa de activación con la predicción de los núcleos predichos.

En la Figura 3.21 se puede observar la misma arquitectura de red, pero con la incorporación de bloques residuales. Estos bloques lo que hacen es transmitir la información de la capa de partida antes de pasar por cada bloque convolucional. Por tanto, se efectúa una operación *MaxPooling* para reducir la dimensionalidad acorde a la reducción que genera el bloque convolucional y se concatena a la salida del bloque activándolo con la función de activación ReLu.

Del mismo modo que la UNet secuencial, el proceso de entrenamiento se ha entrenado durante 80 épocas y ha reservado 6 de las 30 imágenes para la validación.

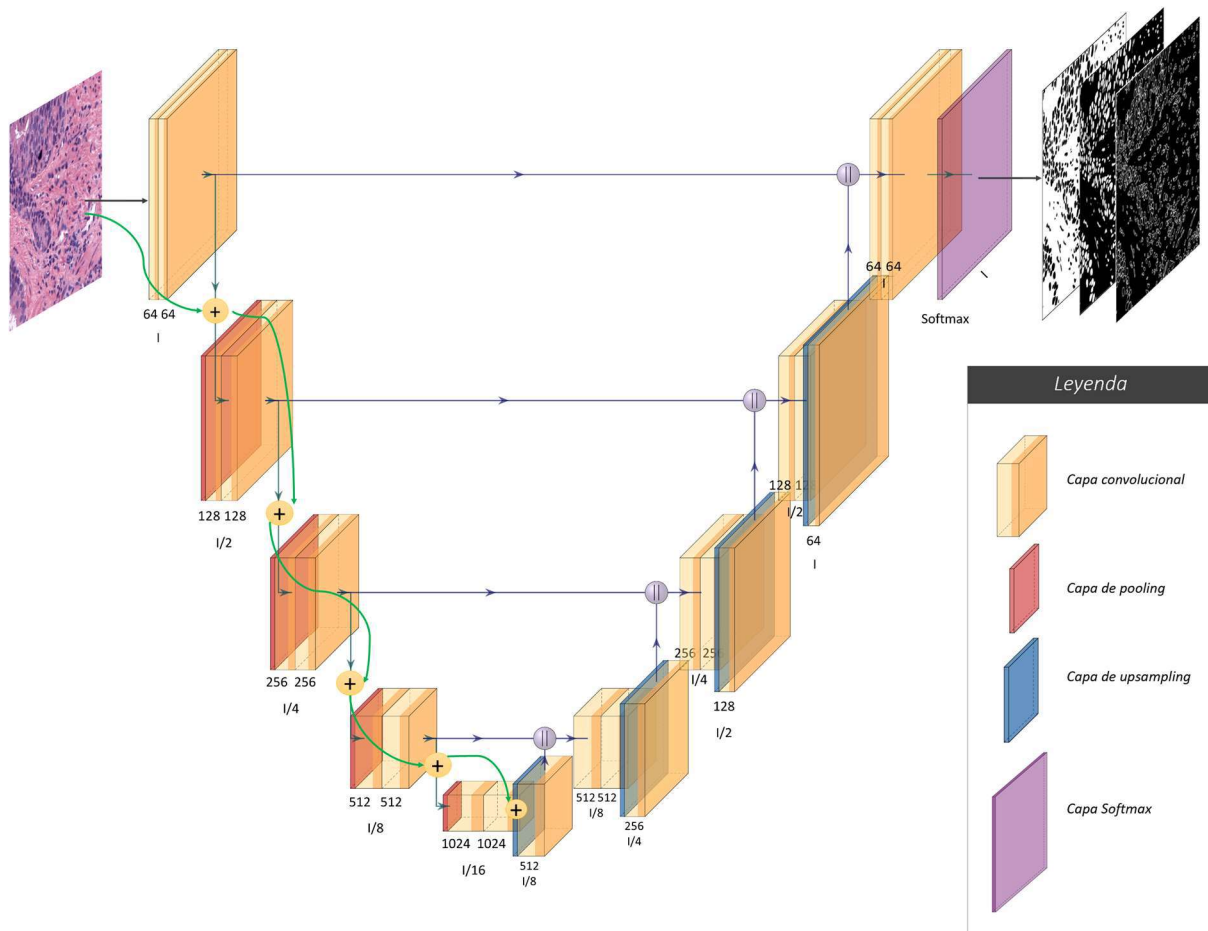


Figura 3.21. UNet con bloques residuales
(Diagrama generado con PlotNeuralNet-master [26])

3.2.9. Postprocesado

Una vez obtenidos los resultados de aplicar ambos modelos de segmentación a las imágenes reservadas para test se obtienen los mapas de probabilidad de pertenencia a clase. En ellos se distinguen tres profundidades, correspondientes a interior de núcleo, borde de núcleo y fondo.

Dichos mapas de probabilidad de clase son la entrada a un algoritmo desarrollado en Matlab encargado de leer dicha imagen y obtener las máscaras de instancia segmentación de núcleos. A dichos mapas de probabilidad, se aplica el algoritmo de inundación de regiones *watershed* para conseguir separar cada uno de los núcleos y generar máscara de instancia de segmentación.

i. Watershed

La transformación *watershed* permite la segmentación de imágenes, dividiendo cada objeto por separado. La forma de hacerlo es la siguiente:

Sea $u(x,y)$ con $(x,y) \in R^2$, una función escalar que describa una imagen I , y el gradiente morfológico de I definido como

$$\delta_D u = (u \oplus D) - (u \ominus D), \quad (3.31)$$

donde $(u \oplus D)$ y $(u \ominus D)$ son respectivamente la dilatación elemental y la erosión de u por el elemento estructurante D .

Por otro lado, el laplaciano morfológico viene dado por

$$\Delta_D u = (u \oplus D) - 2u + (u \ominus D) \quad (3.32)$$

El Laplaciano morfológico nos permite distinguir las zonas de influencia de ínfimos y supremos: las regiones con $\Delta_D u < 0$ se consideran zonas de influencia suprema, mientras que las regiones con $\Delta_D u > 0$ son zonas de influencia ínfima. Por lo que $\Delta_D u = 0$ permite interpretar las ubicaciones de los bordes y representará una propiedad esencial para la construcción de filtros morfológicos. La idea básica es aplicar una dilatación o una erosión a la imagen I , dependiendo de si el píxel se encuentra dentro de la zona de influencia de un mínimo o un máximo.

La cuenca de captación $C(M)$ asociada a un mínimo M es el conjunto de píxeles p de Ω de tal manera que una gota de agua que cae en p fluye hacia abajo a lo largo del relieve, siguiendo un cierto camino descendente, y finalmente llega a M . Las cuencas de captación de la imagen I corresponden entonces a las zonas de influencia de sus mínimos, y la cuenca se definirá por las líneas que separan las cuencas adyacentes.

En la Figura 3.22 vemos se ilustra el funcionamiento del algoritmo *watershed* en las imágenes estudiadas. La forma en que se realiza el watershed es tomando como marcador interno una umbralización del canal que contiene la región interior de núcleo proveniente del mapa de activaciones que sale de la red neuronal. Mientras que el marcador externo es la umbralización del canal que contiene fondo.

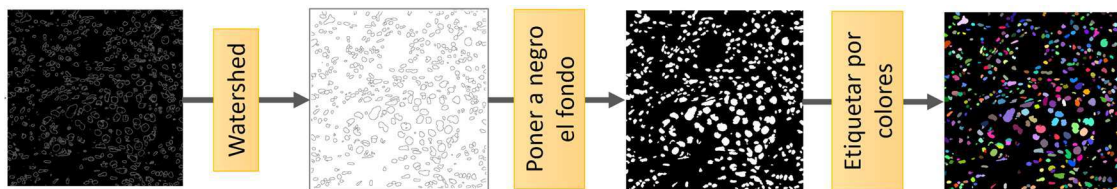


Figura 3.22. Watershed

4. RESULTADOS

En este capítulo se detallarán los resultados obtenidos en las distintas etapas descritas en la Metodología. En primer lugar, se mostrará el resultado de aplicar la red neuronal convolucional a las imágenes de test. Tras eso, se compararán las puntuaciones obtenidas de las métricas AJI para cada una de las redes

4.1. Resultados cualitativos sobre las imágenes de test

La segmentación automática sobre las 16 imágenes de test nos deja como resultado un mapa de predicción a pertenecer a una u otra clase. En la Figura 4.1 podemos ver cómo se ha segmentado automáticamente la imagen *16B0008066 HE 5_060220171540_59655_119188.tif*, la cual pertenece al dataset del Hospital Clínico de Valencia.

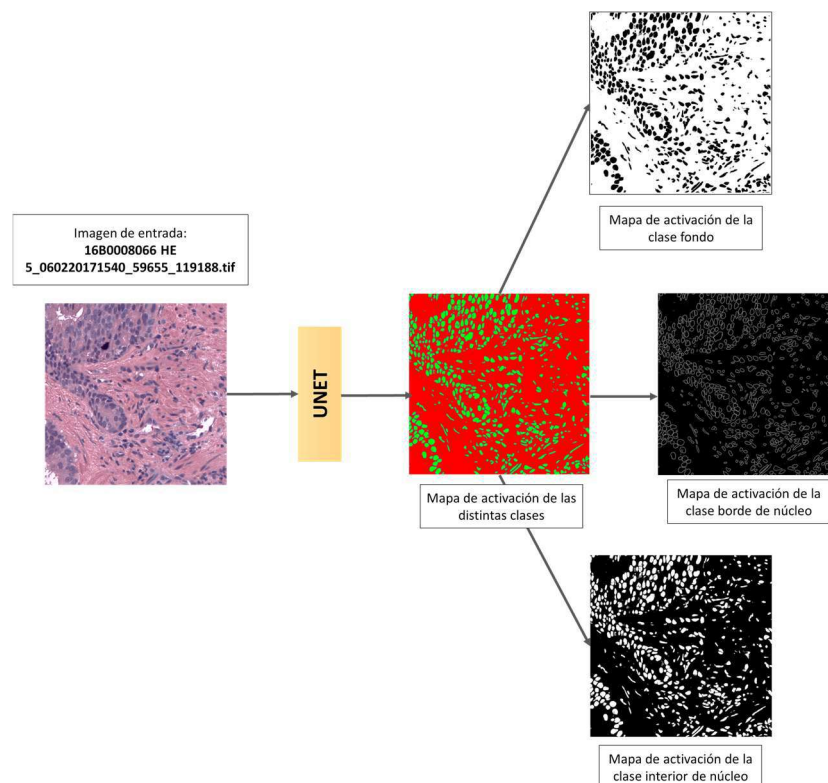


Figura 4.1. Resultado del modelo de segmentación sobre una imagen representativa del conjunto de test

Se obtiene una imagen de 3 canales donde en la primera profundidad del volumen predicho se obtienen valores de probabilidad de pertenencia a la clase fondo de núcleo. Es por eso que vemos prácticamente todo el estroma pintado de blanco, dado que la red ha delimitado muy bien esta clase.

En segundo lugar, nos encontramos con el segundo canal, que contiene la probabilidad de pertenecer a la clase interior de núcleo. En esta ocasión los píxeles que tienden a la unidad son los relativos a los núcleos. Mientras que el fondo se va a mantener a 0, es decir, a negro.

Finalmente, el tercer canal hace referencia a la clase borde de núcleo. Los píxeles con alta probabilidad en este canal serán los encargados de delimitar cada uno de los núcleos contenidos en la imagen histológica.

Una vez obtenidos los mapas de probabilidad de pertenecer a una u otra clase se procede a separar los núcleos uno a uno mediante la transformación *watershed*. Esto permite generar la máscara de instancia que se ilustra en la Figura 4.2.

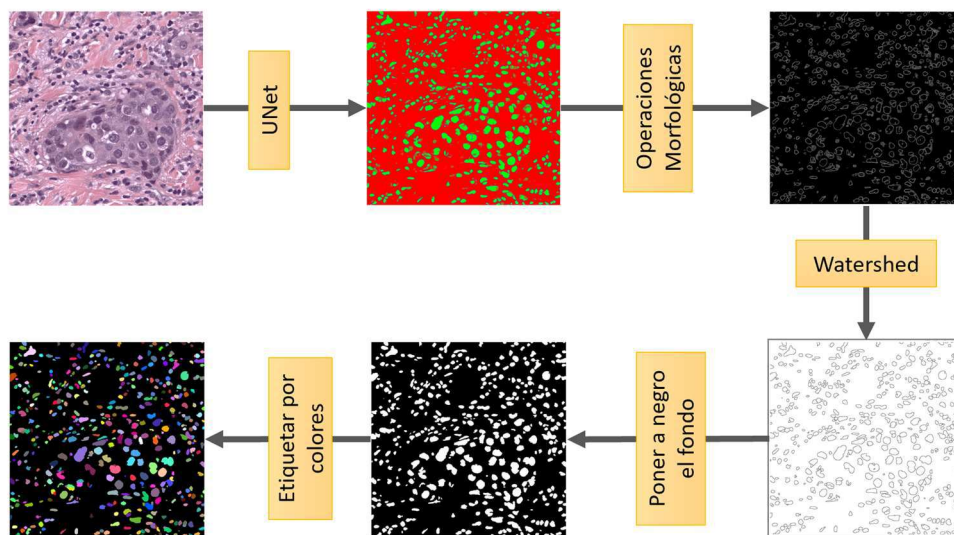
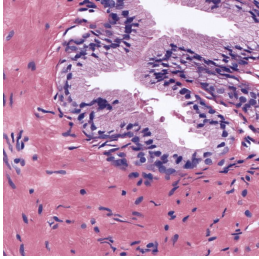
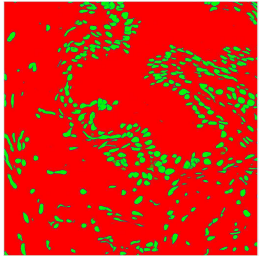
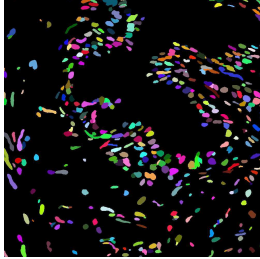
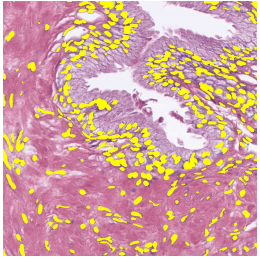
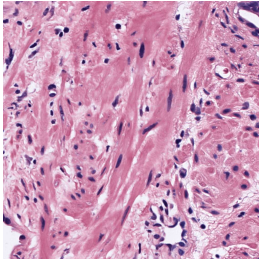
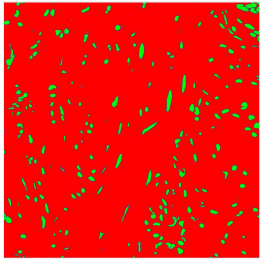
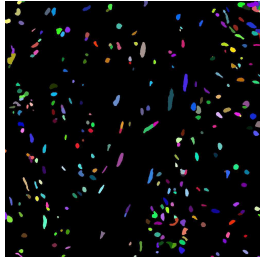
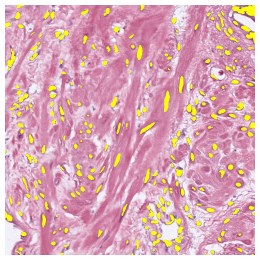
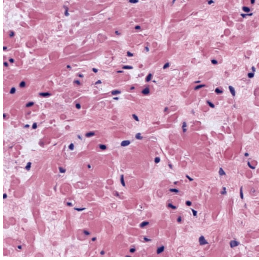
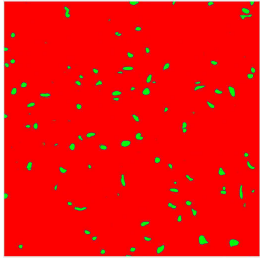

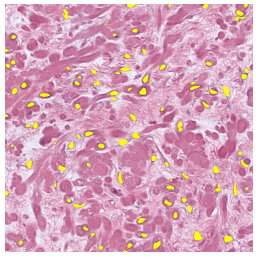
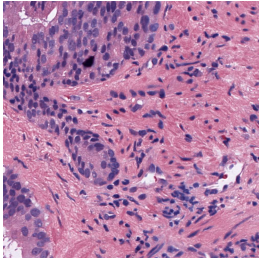
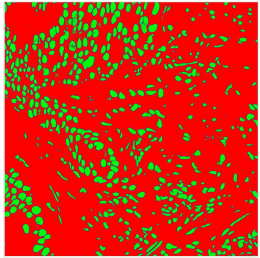
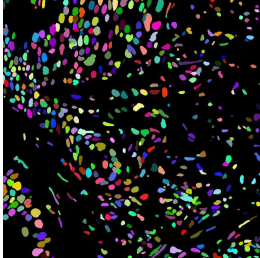
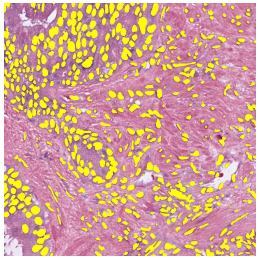
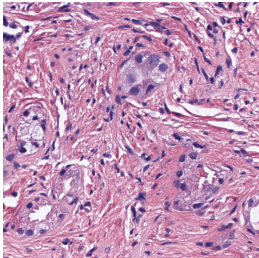
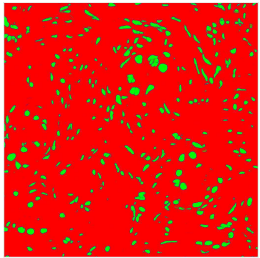
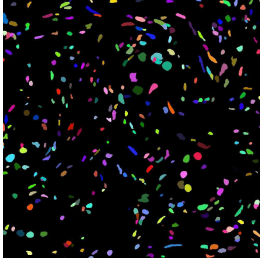
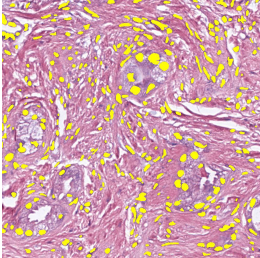
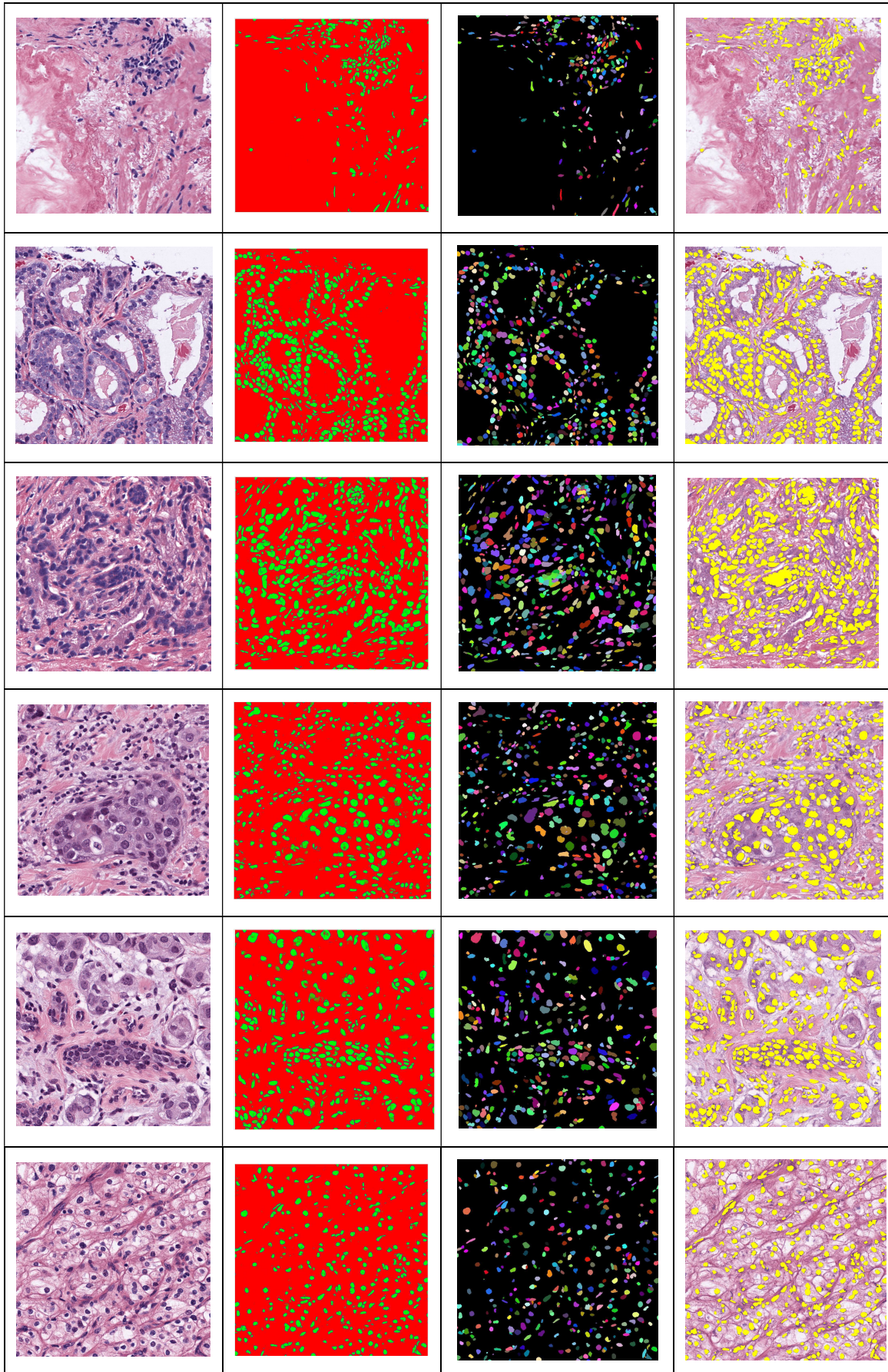


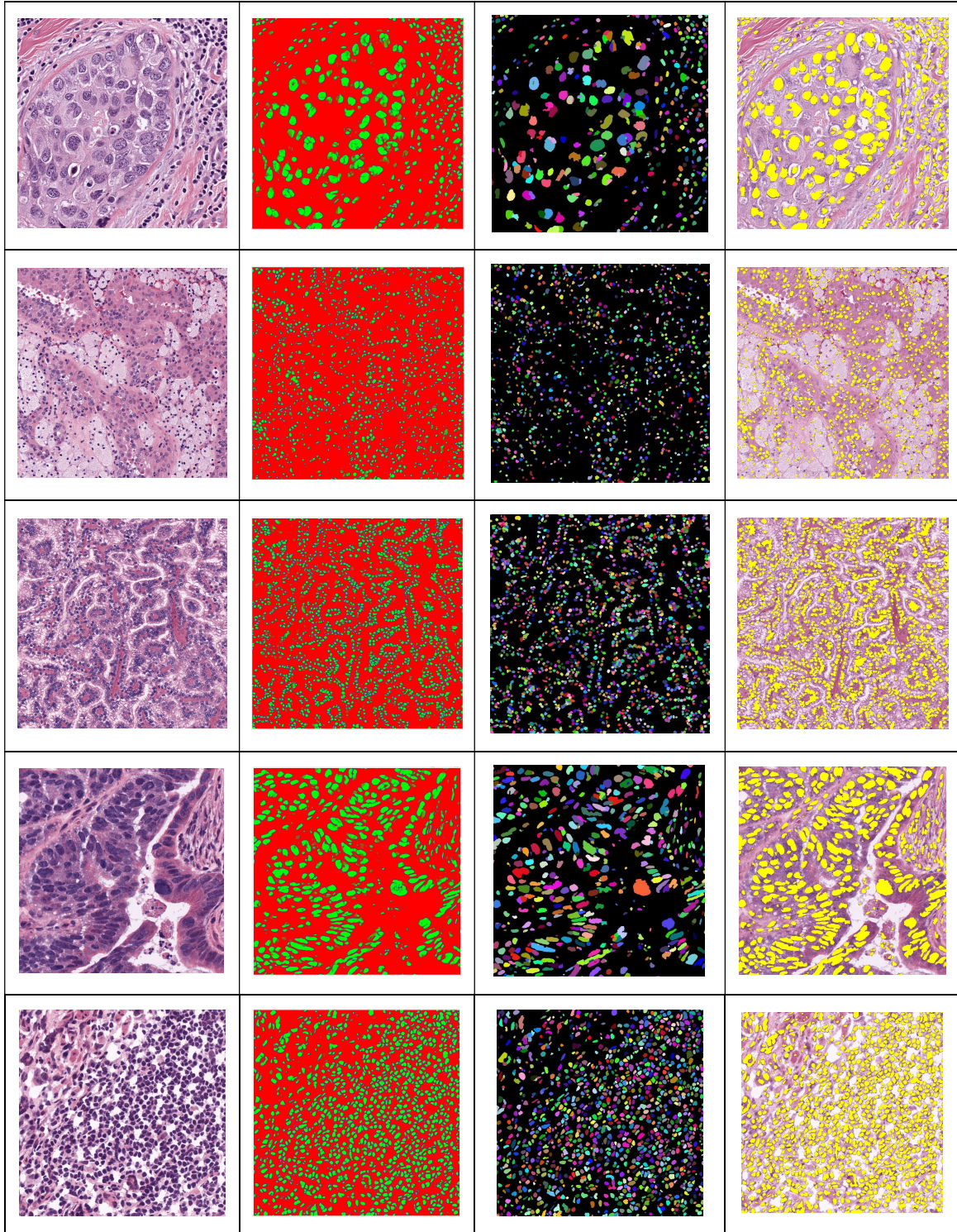
Figura 4.2. Postprocesamiento y obtención de la máscara de instancia

En la Tabla 4.1 se exponen las segmentaciones automáticas realizadas con la red UNet secuencial (sin bloques residuales) sobre las 16 imágenes que componen el dataset de test. En esta tabla podemos ver en, primer lugar, la imagen a segmentar que, tras pasar por la red se convierte en la mapa de probabilidades que contiene en cada canal las probabilidades de pertenecer a las tres posibles clases (borde, interior o fondo de núcleo).

Tabla 4.1. Resultado de la segmentación automática

Imagen original	Mapa predicho	Máscara de instancia	Solapamiento
			
			
			
			
			





4.2. Resultados cuantitativos y comparativa

La evaluación de los modelos de segmentación se llevará a cabo mediante la métrica *Aggergated Jaccard Index* (AJI).

4.2.1. *Aggergated Jaccard Index*

La métrica que se ha utilizado para evaluar el rendimiento del método de segmentación nuclear sobre una imagen o un conjunto de datos de imágenes es el índice Jaccard agregado (AJI). Esta métrica calcula un numerador de cardinalidad de intersección agregada y un denominador de cardinalidad de unión agregada para la máscara de instancia etiquetada a mano y la máscara de instancia segmentada mediante la red neuronal convolucional.

Para el *ground truth* de núcleos, G_i , en una imagen, se le asocia un núcleo segmentado S_j , y se añaden las contribuciones al índice Jaccard agregado mediante el recuento de píxeles de $G_i \cap S_j$ al numerador AJI y el de $G_i \cup S_j$ al denominador. De este modo se añaden píxeles del *ground truth* de núcleos que no se encuentran en la intersección de núcleos segmentados automáticamente al denominador (detección de falsos negativos). También agregamos los recuentos de píxeles de todos los núcleos segmentados no estudiados al denominador (detección de falsos positivos).

Debido a que esta métrica agrega los recuentos de píxeles de falsos positivos y falsos negativos al denominador, además de los píxeles de no superposición entre el *ground truth* y los núcleos detectados (detección verdadera), penaliza estos cuatro tipos de errores:

- 1) Pérdida en la detección de objetos del *ground truth*
- 2) Falsa detección de objetos fantasmas
- 3) Segmentación insuficiente de los objetos detectados correctamente
- 4) Segmentación excesiva de los objetos detectados correctamente.

Las métricas de segmentación que solo se calculan sobre verdaderos positivos, y los píxeles en falsos positivos y falsos los negativos ignoran por completo la calidad de la segmentación. Cuando un núcleo sub-segmentado corresponde a múltiples núcleos de verdad fundamentales, la métrica propuesta tiene el potencial de contar varios píxeles falsamente detectados varias veces en el denominador. Esto es necesario, porque de lo contrario es posible obtener un índice Jaccard medio bajo al sesgar un sistema de segmentación hacia una ligera sub-segmentación. Por lo tanto, el índice Jaccard agregado (AJI), en general, tiene un valor más bajo que el puntaje F1 y el índice Jaccard medio.

El criterio de detección y la métrica de evaluación se describen en detalle en el *Algoritmo 3*. En caso de que ningún núcleo segmentado se cruce con un núcleo de verdad fundamental, la cardinalidad del píxel de intersección es cero y la cardinalidad del píxel de unión es la misma que $|G_i|$ en el paso 4 del algoritmo. AJI oscilará entre 0 para el peor de los casos (sin intersección entre la verdad fundamental y los objetos segmentados) y 1 para el mejor de los casos (detección y segmentación perfectas).

Algoritmo 3. Computación del Aggregated Jaccard Index (AJI)

Input: Conjunto de imágenes que combinan núcleos anotados G_i indexados por i , y un conjunto de núcleos segmentados S_k indexados por k .

Output: Aggregated Jaccard Index A .

```

1: Inicializar recuentos de píxeles de unión y correcta:  $C \leftarrow 0$ ;  $U \leftarrow 0$ 
2: for cada núcleo del ground truth  $G_i$  do
3:    $j \leftarrow \operatorname{argmax}_k (|G_i \cap S_k| / |G_i \cup S_k|)$ 
4:   Actualiza el recuento de píxeles:  $C \leftarrow C + |G_i \cap S_j|$ ;  $U \leftarrow U + |G_i \cup S_j|$ 
5:   Marcar la  $S_j$  usada
6: end for
7: for Cada núcleo segmentado  $S_j$  do
8:   If  $S_k$  no está anotada
9:      $U \leftarrow U + |S_k|$ 
10: end for
11:  $A \leftarrow C/U$ 
    
```

4.2.2. Comparativa de las dos redes propuestas

En la Tabla 4.2 podemos ver el resultado AJI obtenido para cada una de las imágenes segmentadas automáticamente.

Tabla 4.2. Resultados de la métrica AJI

Imagen	UNet	UNet + ResNet
16B 0024181 HE B_060220171401_58753_38961.tif	0.52710	0.52040
16B 0028813 HE B_060220171257_12240_111982.tif	0.56700	0.56600
16B 0028817 HE B_060220171416_42714_92070.tif	0.77890	0.77830
16B0008066 HE 5_060220171540_59655_119188.tif	0.60560	0.61540
16B0022615 HE 3_060220171438_31004_17485.tif	0.59490	0.59430
16B0022781 HE 4_060220171312_12859_117325.tif	0.53360	0.53420
16B0028914 HE 3_060220171505_65064_118316.tif	0.64410	0.64190
17B0016589G4_051220171417_34706_23525.tif	0.62900	0.62250
TCGA-A7-A13E-01Z-00-DX1.tif	0.58240	0.56930
TCGA-A7-A13F-01Z-00-DX1.tif	0.59580	0.60570
TCGA-B0-5711-01Z-00-DX1.tif	0.61650	0.61840
TCGA-E2-A14V-01Z-00-DX1.tif	0.57390	0.56530
TCGA-HE-7128-01Z-00-DX1.tif	0.59890	0.58820
TCGA-HE-7129-01Z-00-DX1.tif	0.54790	0.54270
TCGA-NH-A8F7-01A-01-TS1.tif	0.53690	0.54070
TCGA-RD-A8N9-01A-01-TS1.tif	0.69630	0.69740
Media	0.60180	0.60004

Se puede apreciar que la red UNet sin bloques residuales ha dado un mejor resultado en 10 de las 16 imágenes. Con el fin de facilitar esta lectura se ha sombreado el mejor resultado en cada caso.

Por tanto, la media de todas las métricas AJI de la red UNet es 0.60180. En el siguiente apartado se va a hacer una comparativa con los resultados que obtuvieron los participantes de MoNuSeg. No obstante, cabe destacar que el dataset usado en el challenge fue distinto y no incorporaba ninguna imagen cancerígena. La presencia de estas imágenes en nuestro dataset lo hace mucho más representativo y escalable a cualquier muestra que se estudie en un futuro.

4.2.3. Comparativa de resultados con MoNuSeg

Vemos, en la Tabla 3.1 los resultados de los participantes que se presentaron al challenge de segmentación nuclear MoNuSeg [15].

Tabla 4.3. Resultados challenge MoNuSeg

Rank	Team	AJI	Affiliation	Country
1	CUHK&IMSIGHT	0.6907	The Chinese University of Hong Kong; Imsight Technology	China
2	BUPT.J.LI	0.6868	Beijing University of Post and Telecommunication	China
3	pku.hzq	0.6852	Peking University	China
4	Yunzhi	0.6788	University of Oklahoma	USA
5	Navid Alemi	0.6779	University of Warwick	UK
6	xuhuaren	0.6642	Shanghai Jiao Tong University, Shanghai, China; National University of Defense Technology, Changsha, China	China
7	aetherAI	0.6632	aetherAI	Taiwan
8	Shuang Yang	0.6624	Zhejiang University	China
9	Bio-totem&SYSUCC	0.6619	Bio-totem Pte Ltd, The University of Hong Kong, Sun Yat-sen University Cancer Center	China
10	Amirreza Mahbod	0.6574	Medical University of Vienna, TissueGnostics GmbH, Loughborough University, KTH Royal Institute of Technology	Austria
11	CMU-UIUC	0.6557	Carnegie Mellon University; University of Illinois at Urbana-Champaign; University of Science, VNU-HCM	USA
12	Graham&Vu	0.6532	University of Warwick, Sejong University	UK
13	Unblockabulls	0.6514	American Express, India	India
14	Tencent AI Lab	0.6459	University of California, Berkeley; The Hong Kong University of Science and Technology	USA

15	DeepMD	0.6333	Science and Research Branch, Islamic Azad University, Tehran	Iran
16	Canon Medical Research Europe	0.6325	Canon Medical Research Europe	UK
17	Johannes Stegmaier	0.6231	Institute of Imaging and Computer Vision, RWTH Aachen University	Germany
18	Yanping	0.623	University of Science and Technology of China	China
19	Philipp Gruening	0.6211	University of Luebeck, Institute for Neuro- and Bioinformatics	Germany
20	Agilent Labs	0.618	Agilent Labs, Agilent Tehcnologies Israel Ltd	Israel
21	Konica Minolta Laboratory Europe	0.6112	Konica Minolta Laboratory Europe, Munich, Germany	Germany
22	OnePiece	0.6061	Shenzhen University, Shenzhen, China	China
23	Junma	0.5934	Department of Mathematics, Nanjing University of Science and Technology	China
24	Biosciences R&D, TCS Research	0.5783	Biosciences R&D, TCS Research, TATA Consultancy Services Ltd., Pune, India	India
25	Azam Khan	0.5749	Korea University	South Korea
26	CVBLab	0.5739	Universitat Politècnica de València	Spain
27	Linmin Pei	0.5623	Vision Lab, Electrical & Computer Engineering, Old Dominion University	USA
28	DB-KR-JU	0.4545	Jadavpur University	India
29	VISILAB	0.4441	Visilab Research Group, University of Castilla - La Mancha, Ciudad Real, Spain	Spain
30	Sabarinathan	0.4437	Cognizant Technology Solutions, India	India
31	Silvers	0.278	Xiamen University, Xiamen, China	China
32	TJ	0.1301	Tongji University	China

Teniendo en cuenta estos datos, la red propuesta quedaría en la posición 23, mejorando por tanto la red presentada por el propio CVBLab, que se encuentra en la posición 26. Como se ha comentado en el apartado anterior, los resultados del challenge MonuSeg se obtienen de entrenar y testear con imágenes sanas, donde los núcleos se presentan con formas mucho más regulares, lo que hace el aprendizaje de estos más sencillo para la red que se proponga.

5. CONCLUSIONES Y LÍNEAS FUTURAS

5.1. Conclusiones

En este trabajo se ha desarrollado una introducción al cáncer, haciendo hincapié en la cantidad de personas que lo padecen y el número de muertes que produce. Seguidamente se han detallado todos los procedimientos que se deben llevar a cabo para poder obtener una imagen histológica un patólogo puede analizar a microscopio. Se presenta entonces el problema a tratar, los núcleos de las imágenes histológicas. Estos tienen gran diversidad de formas y en muchas ocasiones se solapan unos con otros. A lo largo del trabajo se han propuesto dos redes neuronales capaces de abordar el problema de la segmentación nuclear. Los resultados de la red UNet son capaces de segmentar los núcleos, obteniendo un AJI de 0,6018. Aunque los resultados no son ideales y quedan por debajo de los mejores resultados del challenge MoNuSeg, se puede afirmar que sí son capaces de ofrecer un resultado prometedor. Cabe destacar que en el concurso MoNuSeg no se trabajó con pacientes con cáncer, por lo que las imágenes son mucho más regulares y puede ser indicativo de obtener mejores resultados.

El primer paso ha sido conocer las imágenes desde un punto de vista histológico para poder determinar qué formas y aspectos puede tener el tejido, así como saber si los núcleos eran susceptibles a cambio al padecer cáncer. Estos datos son una información necesaria para la conformación de la base de datos. También en esta línea se ha realizado un análisis del estado del arte con el fin de conocer las técnicas que han sido empleadas hasta el momento y qué calidad de resultados se han obtenido. Esto nos permite, por un lado, abordar el problema mediante una aproximación adecuada y, por otro lado, proponer nuevas herramientas y evaluar qué pueden aportar a la resolución del problema principal.

En segundo lugar, se ha creado una base de datos que es combinación de el dataset del challenge MoNuSeg y otras imágenes propias cedidas por el Hospital Clínico Universitario de Valencia. Esta base de datos contiene imágenes histológicas de 1000 x 1000 píxeles, a las que se ha normalizado el color mediante SNMF. En la base también encontramos las máscaras con 3 intensidades distintas que denotan interior de núcleo, borde de núcleo y fondo y por último los mapas de núcleos para conformar el *ground truth*.

Con esta base se han entrenado dos redes neuronales donde se han destinado 30 imágenes al entrenamiento, con el 20% a la validación, y otras 16 imágenes al test. La primera red neuronal es una *encoder – decoder* tipo UNet con *skip connections* que ha sido capaz de sacar un resultado de métrica AJI de 0,6018, superando a la otra red tipo UNet pero con bloques residuales. Se ha implementado un

algoritmo de postprocesado basado en el método watershed utilizando el mapa de predicción que sale de la red neuronal convolucional.

Como se ha comentado, los resultados obtenidos aún deben mejorarse para poder equipararlos a la segmentación hecha a mano. Esto puede deberse a diversos factores, entre ellos el parcheo de las imágenes, es fácil comprobar que en el mapa de predicción aparecen errores en forma de líneas que parecen coincidir con los bordes del parche, si estas líneas caen sobre un núcleo al hacer el *watershed* se divide en dos cuencas y segmenta incorrectamente el núcleo. Por otro lado, también puede deberse a la forma en la que se ha implementado el *watershed*, el umbral de decisión se ha ido ajustando a mano mediante el mapa de predicción, pero no existe un método que evalúe cada mapa por separado con el fin de hacer el *watershed* óptimo para esa imagen en concreto. Esto también puede estar relacionado con el resultado de la UNet con bloques residuales, como se ha comentado en la literatura encontramos referencias que indican que el uso de bloques residuales mejora la segmentación. No obstante, en nuestro caso no ha ocurrido. Esto podría deberse a la profundidad de la red, quizá si generamos una red más profunda esta segmente mejor hasta que llegue a la saturación por degeneración, y entonces sí sea útil el uso de bloques residuales. Pero como se comenta, es posible que si implementamos un algoritmo que estudie individualmente de los mapas de predicción y encuentre el mejor umbral para comenzar a hacer el *watershed*, obtengamos mejores resultados en la métrica AJI.

Por lo tanto, se ha determinado que, en este caso, la implementación de bloques residuales no aumenta el resultado y es conveniente no emplearlos ya que aumentaría, aún más, el tiempo de cómputo.

Como conclusión global del trabajo se puede afirmar que se ha logrado alcanzar el objetivo principal planteado: crear un modelo de segmentación de núcleos basado en aprendizaje profundo.

5.2. Líneas futuras

En base a las conclusiones extraídas de todo el proceso del trabajo se van a proponer algunas mejoras que partan de los puntos débiles de la red neuronal para mejorarlo y conseguir resultados de mayor calidad.

Por un lado, sería interesante implementar data augmentation, con el fin de introducir mayor cantidad de ejemplos de núcleos a la red. Así como modificar el parcheo para evitar la aparición de las aberraciones en el mapa de salida.

En cuanto a la etapa de la base de datos, sería conveniente conseguir más muestras de todos los órganos, para poder generar una red capaz de identificarlos todos con la misma precisión. Con esto se aumenta también el tamaño del training y la red aprende todos los tipos de núcleo.

Otro enfoque que se podría seguir es la implementación de una red neuronal generadora adversaria (GAN), este tipo de redes se compone de una parte generadora y otra discriminatoria. De modo que podría hacerse una red que a partir de una matriz de ruido aleatorio generase una imagen histológica con núcleos y otra red antagónica entrenada para poder discriminar esta imagen como realista o no. De este modo, sin necesidad de segmentar a mano nuevas imágenes, podríamos obtener imágenes segmentadas que sirviesen para entrenar la red propuesta en este proyecto.

Por último, si se implementara la red neuronal en un sistema de ayuda al diagnóstico completo, podría hacerse telepatología, con este sistema las imágenes donde la distribución y formas nucleares sean atípicas se envíen directamente al patólogo para que las estudie. Reduciendo así el esfuerzo de analizar todas las imágenes que no tienen ninguna anormalidad. Podría así implementarse en los quirófanos un sistema de histología por congelación, como el que se ha descrito en la sección 1.2.1, y un instrumental que digitalizase la imagen, se analizara con la red neuronal convolucional y se enviara al patólogo al instante. De este modo, las operaciones en las que se debe decidir si extirpar o no una zona posiblemente afectada por cáncer no tendría que esperar a que el patólogo haga todo el procedimiento y observe la muestra. No obstante, para este caso tendría que reentrenarse la red con imágenes histológicas por congelación.

BIBLIOGRAFÍA

- [1] "OMS | Cáncer," *WHO*, 2017.
- [2] R. W. Ruddon, *Characteristics of Human Cancer*, no. October. 2007.
- [3] S. H. Hassanpour and M. Dehghani, "Review of cancer from perspective of molecular," *J. Cancer Res. Pract.*, vol. 4, no. 4, pp. 127–129, 2017.
- [4] M. Roser and H. Ritchie, "Cancer," *Our World Data*, Jul. 2015.
- [5] G. Rolls, "An Introduction to Specimen Preparation." 29-Nov-2011.
- [6] M. Meg, P. Molist, and M. A. Pombal, "Técnicas histológicas de Tinción," 2018.
- [7] W. Avenue, E. Eu, and E. A. Pacific, "Hematoxylin and Eosin," *Encycl. Cancer*, pp. 1645–1645, 2011.
- [8] "La Inteligencia Artificial (AI) frente a la Máquina de Aprendizaje y Profundo de Aprendizaje | Skymind." [Online]. Available: <https://skymind.ai/wiki/ai-vs-machine-learning-vs-deep-learning>. [Accessed: 04-Sep-2019].
- [9] "Computing Machinery and Intelligence | Alan Turing | 1st Edition." [Online]. Available: <https://www.manhattanrarebooks.com/pages/books/227/alan-turing/computing-machinery-and-intelligence/?soldItem=true>. [Accessed: 26-Jun-2019].
- [10] F. Mahmood *et al.*, "Deep Adversarial Training for Multi-Organ Nuclei Segmentation in Histopathology Images," *IEEE Trans. Med. Imaging*, pp. 1–1, 2019.
- [11] A. Janowczyk and A. Madabhushi, "Deep learning for digital pathology image analysis: A comprehensive tutorial with selected use cases," *J. Pathol. Inform.*, vol. 7, no. 1, 2016.
- [12] J. Long, E. Shelhamer, and T. Darrell, "Fully Convolutional Networks for Semantic Segmentation."
- [13] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2015, vol. 9351, pp. 234–241.
- [14] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition."
- [15] "MoNuSeg - Home." [Online]. Available: <https://monuseg.grand-challenge.org/Home/>. [Accessed: 24-Aug-2019].
- [16] "MATLAB - El lenguaje del cálculo técnico - MATLAB & Simulink." [Online]. Available: <https://es.mathworks.com/products/matlab.html>. [Accessed: 24-Aug-2019].
- [17] "Histology images of Cells & epithelia by PathPedia.com: Pathology e-Atlas." [Online]. Available: https://www.pathpedia.com/Education/eAtlas/Histology/cells_and_epithelia/Images.aspx. [Accessed: 03-Sep-2019].
- [18] J. Xu *et al.*, "Sparse Non-negative Matrix Factorization (SNMF) based color unmixing for breast histopathological image analysis," *Comput. Med. Imaging Graph.*, vol. 46, pp. 20–29, 2015.

- [19] K. O’Shea and R. Nash, “An Introduction to Convolutional Neural Networks,” no. November, 2015.
- [20] “Everything you need to know about Neural Networks and Backpropagation — Machine Learning Easy and Fun.” [Online]. Available: <https://towardsdatascience.com/everything-you-need-to-know-about-neural-networks-and-backpropagation-machine-learning-made-easy-e5285bc2be3a>. [Accessed: 03-Sep-2019].
- [21] C. E. Nwankpa, W. Ijomah, A. Gachagan, and S. Marshall, “Activation Functions : Comparison of Trends in Practice and Research for Deep Learning,” pp. 1–20.
- [22] “What is the Difference Between a Batch and an Epoch in a Neural Network?” [Online]. Available: <https://machinelearningmastery.com/difference-between-a-batch-and-an-epoch/>. [Accessed: 04-Sep-2019].
- [23] “Learning Rate Scheduling - Deep Learning Wizard.” [Online]. Available: https://www.deeplearningwizard.com/deep_learning/boosting_models_pytorch/lr_scheduling/. [Accessed: 04-Sep-2019].
- [24] “Understand the Impact of Learning Rate on Neural Network Performance.” [Online]. Available: <https://machinelearningmastery.com/understand-the-dynamics-of-learning-rate-on-deep-learning-neural-networks/>. [Accessed: 04-Sep-2019].
- [25] A. Rosebrock, *Deep Learning for Computer Vision with Python*. 2017.
- [26] H. Iqbal, “PlotNeuralNet.” 2018.
- [27] V. Dumoulin and F. Visin, “A guide to convolution arithmetic for deep learning,” pp. 1–31, 2016.

Parte II

Presupuesto

ÍNDICE DEL PRESUPUESTO

1.	PRESUPUESTOS PARCIALES	5
1.1.	COSTE DE PERSONAL	5
1.2.	COSTE DE LOS RECURSOS	5
1.2.1.	<i>Coste de las herramientas hardware y software</i>	6
1.2.2.	<i>Coste de los materiales</i>	6
2.	PRESUPUESTO TOTAL	7

1. Presupuestos parciales

En este capítulo se va a hacer una valoración económica del trabajo realizado. Se van a ir detallando cada una de las partes de las que consta el presupuesto y se va a realizar un breve estudio de la viabilidad económica en función a los resultados que se han obtenido, con el fin de saber si es eficiente realizar este tipo de método de segmentación nuclear.

Primeramente, se desglosarán los precios de modo que se analicen por separado la mano de obra y la maquinaria y los materiales empleados.

1.1. Coste de personal

Dentro de la mano de obra se recogen los recursos humanos que han sido requeridos para la realización del trabajo en función del tiempo que ha sido necesario dedicar. Concretamente, las contribuciones al trabajo han venido de la mano de: Dra. Valery Naranjo Ornedo (tutora del trabajo), Adrián Colomer Granero (cotutor del trabajo) y Adrián Galiana (alumno y autor del proyecto).

Tabla 1.1. Cuadro de precios unitarios de la mano de obra

Denominación	Uds.	Cantidad	Precio Unitario (€)	Total (€)
Catedrática	h.	15	42,00	630,00
Doctor	h.	45	20,50	922,50
Alumno GIB	h.	720	12,50	9.000,00
			Total	10.552,50

1.2. Coste de los recursos

En lo referente a los recursos materiales necesarios en el trabajo se ha hecho la siguiente distinción: maquinaria y materiales.

La maquinaria se refiere a aquellos materiales que han sido necesarios para el desarrollo del trabajo explícitamente, a las herramientas que han sido necesarias en todo el proceso. Se distingue entre hardware y software.

1.2.1. Coste de las herramientas hardware y software

En el caso de este proyecto, las herramientas software utilizadas han sido: MATLAB, concretamente *Image Processing Toolbox* y *Statistics and Machine learning Toolbox*, y para la creación de la presente memoria se ha empleado Microsoft Office, un procesador de textos con la licencia educativa.

Tabla 1.2. Cuadro de precios unitarios de maquinaria

Denominación	Uds.	Cantidad	Precio Unitario (€)	Periodo de amortización (años)	Intervalo amortizado (meses)	Total (€)
Office Hogar y Estudiantes 2019	u	1	149,00	1	11	149,00
Licencia MATLAB	u	1	800,00	1	11	800,00
Equipo especializado equipado con NVIDIA TITAN V	u	1	4.000,00	5	11	4.000,00
DELL XPS 15 9530	u	1	1.300,00	5	11	1.300,00
Total						6.249,00

1.2.2. Coste de los materiales

Siguiendo con los materiales utilizados en el trabajo, en este caso son las imágenes, que provienen de pruebas de biopsia realizadas en el Hospital Clínico Universitario de Valencia. El precio de las imágenes se divide en el precio relativo a la adquisición de la muestra del paciente y en el coste asociado a cada corte histológico que se digitaliza. Es importante destacar que este coste no se ha asumido explícitamente en el presente trabajo.

Tabla 1.3. Cuadro de precios unitarios de los materiales

Denominación	Uds.	Cantidad	Precio Unitario (€)	Total (€)
Biopsia	u	16	600,00	9.600,00
Muestras	u	16	10,00	160,00
Total				9.760,00

2. Presupuesto total

Finalmente, se va a calcular el presupuesto total del proyecto. Para ello, a los cuadros de presupuestos parciales anteriores (Tabla 1.1, Tabla 1.2, Tabla 1.3) hay que sumar el porcentaje de gastos generales (13%), el beneficio industrial (6%) y el IVA (21%). Tras esto, se obtiene el presupuesto total que supone la realización del presente trabajo.

Tabla 2.1. Presupuesto total

CAPÍTULO	IMPORTE (€)
Coste de la mano de obra	10.552,50
Coste de la maquinaria	6.249,00
Coste de los materiales	9.760,00
PRESUPUESTO DE EJECUCIÓN DE MATERIAL	26.561,50
13 % de gastos generales	3.453,00
6% de beneficio industrial	1.593,69
PRESUPUESTO DE EJECUCIÓN POR CONTRATA	31.608,19
21 % de IVA	6.637,72
PRESUPUESTO TOTAL	38.245,91