Facultade de Informática
UNIVERSIDADE DA CORUÑA

TRABAJO FIN DE GRADO
GRADO EN INGENIERÍA INFORMÁTICA
MENCIÓN EN COMPUTACIÓN

Aprendizaje profundo con predicción monocular de profundidad estéreo para el diagnóstico de glaucoma

Estudiante: Miguel Alejandro Díaz Freire
Dirección: José Rouco Maseda, Jorge Novo Buján

A Coruña, setembro de 2021.

A un joven llamado Cuervo.

Agradecimientos

Me gustaría agradecer tanto a los directores del proyecto José Rouco y Jorge Novo como a José Morano por su guía y ayuda a lo largo del desarrollo de todo el proyecto, sin la cual no me hubiese sido posible afrontarlo de forma tan positiva ni entender todo de forma tan clara. Por otro lado, agradezco a mi madre, a mi hermano y a mis amigos por su apoyo, confianza y compañía en todo momento. Muchas gracias.

Resumen

El diagnóstico de afecciones oculares se realiza mediante el apoyo de diferentes modalidades de imágenes oftalmológicas. En concreto, en el diagnóstico de glaucoma, grave enfermedad que apenas presenta síntomas en sus etapas iniciales, es útil la utilización de retinografías para obtener indicadores de la presencia de la enfermedad. En general, las retinografías están ampliamente extendidas en el mundo de la oftalmología por ser de obtención no-invasiva. Sin embargo, existen diferentes tipos de retinografías. Descatan las retinografías monoculares, de fácil obtención, y las estereográficas, más complejas de obtener y menos accesibles en el ámbito hospitalario por necesidad de cámaras especializadas, pero que arrojan información sobre la profundidad del fondo ocular muy relevante a la hora de diagnósticar el glaucoma. En este proyecto se plantea la utilización de metodologías de aprendizaje profundo mediante diferentes aproximaciones para la obtención de la información de profundidad de las retinografías estereográficas pero a partir de retinografías monoculares. Por un lado, se plantea la predicción directa del mapa de profundidad del fondo ocular. Por otro lado, se plantea la predicción del par estereográfico a partir de la otra imagen del par, considerada una retinografía monocular. Finalmente, una vez obtenida esta información de profundidad, se plantea el uso del conocimiento adquirido en esta tarea de pre-entrenamiento para entrenar otro modelo cuyo objetivo sea realizar la tarea de segmentación semántica de disco y copa, regiones del fondo del ojo cuyo ratio es indicador de la presencia de glaucoma. Este procedimiento es conocido como *transfer-learning*, del cual se quiere demostrar su validez en tareas contenidas dentro del mismo campo semántico.

Al comparar los resultados obtenidos por los modelos de segmentación utilizando el conocimiento transferido de las otras tareas con modelos de segmentación entrenados sin este conocimiento, se ha podido demostrar que el *transfer-learning* mejora los resultados en cuanto a la segmentación de la copa se refiere. En el caso de la segmentación de disco, los resultados se mantienen parejos a los de los modelos de referencia. Por otro lado, se han encontrado indicios de que dependiendo de la naturaleza del problema a resolver, es posible que, si la tarea de pre-entrenamiento aprende detalles demasiado específicos sobre su propia tarea, el conocimiento transferido empeore los resultados de la segmentación, como es el caso de la tarea de predicción del mapa de profundidad. No obstante, la tarea de predicción del par estereográfico muestra una tendencia a mejorar la segmentación cuanto mejor resuelva su tarea el modelo de pre-entrenamiento. Estas posibilidades plantean nuevas líneas de investigación que sería interesante seguir en un futuro.

Abstract

The diagnosis of ocular conditions is made through the support of different ophthalmological imaging modalities. In particular, in the diagnosis of glaucoma, a serious disease that has nearly no symptoms on its early stages, the use of retinographies is useful in order to obtain indicators of the presence of the disease. In general, retinographies are broadly used on the ophthalmological world because they can be obtained in a non-invasive way. However, there are different types of retinographies, standing out the monocular ones, easily obtained, and the stereographic ones, more complex to obtain and less accesible in the hospital setting due to the need for specialized cameras, but which give information about the depth of the ocular fundus, which is very relevant when diagnosing glaucoma. In this project, the use of deep learning methodologies is proposed through different approaches to obtain the depth information derived from the stereographic retinographies but from monocular retinographies. On one hand, direct prediction of the ocular fundus depth map is proposed. On the other hand, the prediction of the stereographic pair from the other image of the pair is considered. Finally, once the depth information has been obtained, the use of the knowledge acquired during this pre-training tasks is used to train another model whose objective is to perform the task of cup and disc semantic segmentation, regions of the ocular fundus whose ratio is an indicator of the presence of glaucoma. This procedure is known as transfer-learning and the final objetive of the project is to show its validity in the tasks contained in the same semantic field.

When comparing the results obtained by the segmentation models using the transferred knowledge from the other tasks with the segmentation models trained without this knowledge, it has been shown that transfer-learning improved the results of the cup segmentation. In the case of disk segmentation, the results do not improve nor get worse compared with the reference ones. On the other hand, indications have been found that depending on the nature of the problem to be solved, it can be that, if the pre-training task learns too specific details about his own task, the transferred knowledge can make the segmentation results worse, which is the case with the depth map prediction task. Nevertheless, the stereo par prediction task shows a tendency to improve the segmentation the better the pre-training model solves its task. These possibilities raise new lines of research that it would be interesting to pursue in the future.

Palabras clave:

- Oftalmología
- Retinografía estereográfica
- Retinografía monocular
- Glaucoma
- Aprendizaje profundo
- Transfer-learning
- Segmentación semántica

Keywords:

- Ophthalmology
- Stereographic retinography
- Monocular retinography
- Glaucoma
- Deep learning
- Transfer learning
- Semantic segmentation

Índice general

Índice de figuras

Índice de tablas

Capítulo 1

Introdución

En este primer capítulo se explicará a modo introductorio en qué consiste el proyecto y se expondrán los objetivos a conseguir por el mismo.

1.1 Descripción y objetivos del proyecto

1.1.1 Descripción

En el diagnóstico de diferentes afecciones de carácter oftalmológico es común la necesidad de la obtención de imágenes para la visualización del fondo del ojo, cuyas variaciones suelen ser indicativas de este tipo de enfermedades. Existen diferentes modalidades de imagen de las que se puede extraer información relevante, pero podemos destacar el uso de retinografías en el diagnóstico de enfermedades como el glaucoma, cuyo diagnóstico temprano es de suma importancia debido a sus graves consecuencias. El mayor uso de las retinografías respecto a otras modalidades de imagen es debido a su bajo coste y a la sencillez de su obtención, pues se obtienen mediante técnicas no invasivas, es decir, técnicas que no implican penetrar físicamente en el cuerpo del paciente. Sin embargo, dentro de esta modalidad de imagen, podemos diferenciar entre retinografías monoculares y retinografías estereoscópicas. Si bien las segundas, formadas por dos fotografías desfasadas del fondo del ojo, proporcionan más información sobre esta región, sobre todo información relacionada con la profundidad, son menos asequibles a la hora de su obtención pues requieren la utilización de material especializado más caro y menos extendido en el mundo hospitalario. Por su parte, las retinografías monoculares, consistentes en una única fotografía del fondo ocular, son sencillas y baratas de obtener, por lo que suponen la mayoría de imágenes de este tipo de las que se dispone o puede disponer y sobre las que generalmente hay que realizar el diagnóstico.

En este proyecto, en una primera instancia, se propone la obtención de la información de profundidad de las retinografías estereoscópicas a partir de una única retinografía monocular.

Para la tarea propuesta se utilizarán metodologías de *deep learning*, en concreto redes de neuronas convolucionales. Partiendo del diseño e implementación de una arquitectura adecuada, se plantearán dos caminos diferenciados:

- Predecir directamente el mapa de profundidad a partir de una única retinografía.
- Predecir la retinografía desfasada a partir de su par.

Por último, como tarea finalista se plantea la segmentación semántica de disco y copa, zonas del fondo ocular presentes en las retinografías y cuyas características morfológicas ayudan en el diagnóstico de glaucoma. Se propone utilizar *transfer-learning* para volcar la información obtenida en las tareas de predicción del mapa de profundidad y de predicción del par estereográfico a través de los filtros aprendidos por las redes para entrenar una segunda red convolucional que realice la tarea de segmentación de disco y copa y estudiar los resultados comparándolos al entrenamiento de esta misma red partiendo de una inicialización aleatoria de los pesos. Finalmente, compararemos y estudiaremos la validez de los resultados de los dos pre-entrenamientos: el de predicción del mapa de profundidad y el de predicción del par estereoscópico en la tarea de segmentación.

1.1.2 Objetivos

Los objetivos que pretende alcanzar el proyecto empiezan por la obtención de información de profundidad del fondo ocular a partir de una única imagen retinal mediante el estudio, diseño e implementación de una arquitectura de aprendizaje profundo adecuada para el problema.

Por otro lado, se quiere lograr el uso de técnicas de transfer learning para aplicar los modelos conseguidos en la etapa anterior a tareas de diagnóstico basado en la segmentación de la anatomía del fondo ocular. El objetivo final del proyecto es, mediante los resultados obtenidos, realizar un estudio sobre el efecto en los resultados de la segmentación de utilizar un pre-entrenamiento en una tarea diferente pero contenida dentro del mismo dominio de aplicación, esto es, el campo del análisis de imágenes oftalmológicas. Se pretenden elaborar conclusiones sobre la validez y adecuación del procedimiento realizado y de las metodologías y técnicas utilizadas en el mismo.

De forma más concreta, se quiere conseguir un estudio exhaustivo de la arquitectura U-Net, [1], para su posterior adaptación e implementación en la resolución de las tareas planteadas. Esta arquitectura será la que sentará la base de los proyectos y con la que se pretende llegar a los resultados mencionados. Se utilizará esta misma arquitectura en ambas tareas pues lo que intentaremos demostrar es que el pre-entrenamiento planteado en una tarea dentro del mismo campo semántico mejora los resultados obtenidos frente a una red entrenada de cero,

por lo que es conveniente utilizar la misma arquitectura en todos los experimentos, haciendo las menores variaciones posibles para poder comparar los resultados.

Capítulo 2

Contexto y estado del arte

Para entender mejor el propósito y el desarrollo del proyecto es necesario comprender el dominio en el que se desenvuelve, siendo este el del análisis de imágenes oftalmológicas.

Por un lado, será necesario arrojar luz sobre determinados conceptos propios de la oftalmología como son la anatomía ocular, las modalidades de imagen existentes en este campo y, más específicamente, las que se utilizarán en el proyecto. Además, convendrá conocer las afecciones oculares y, en concreto, el glaucoma, pues es su diagnóstico a través de la tarea de segmentación la motivación final del proyecto. Por último, mencionaremos otros detalles que puedan ser de importancia a la hora de entender mejor la base teórica del proyecto del lado de la medicina.

También será necesario comprender de forma más extensiva los conceptos relacionados con las técnicas que se utilizarán en los experimentos para llevar a cabo las tareas planteadas. Se explicará en detalle lo que es el aprendizaje automático y en concreto el aprendizaje profundo y cómo éste se puede aplicar a la resolución de nuestras tareas. Además, se estudiará el actual estado del arte en la resolución de los problemas planteados y, en general, se presentarán trabajos relacionados que puedan ayudar al desarrollo del proyecto, destacando las diferencias con el planteamiento propuesto.

2.1 El ojo

La evolución de los seres vivos a lo largo de millones de años ha llevado a la aparición de sistemas complejos que, a priori, mejoran sus capacidades de supervivencia en los entornos dinámicos que los rodean. Poder visualizar el mundo exterior supuso claramente para estos individuos una ventaja frente a aquellos que no tenían esta capacidad, por lo que la aparición fortuita del sistema visual fue rápidamente perpetuada por la acción de la selección natural y, hoy en día, podemos ver que la gran mayoría de los seres vivos pertenecientes al reino animal constan de un sistema visual funcional.

El ser humano consta de un sistema visual complejo formado por diferentes estructuras, siendo uno de sus componentes principales **el ojo**.

2.1.1 Función y anatomía del ojo humano

A grandes rasgos, el **ojo humano** se encarga de detectar la luz que incide sobre él y de transformarla en señales eléctricas que serán procesadas en el cerebro. En cuanto a su estructura, podemos diferenciar las siguientes partes de manera poco exhaustiva pero académica para la comprensión del proyecto:

- **Córnea**: Primera capa transparente con la función de proteger la órbita ocular.
- **Pupila**: Diafragma u orificio por donde penetra la luz.
- **Iris**: Regulador del diámetro de la pupila.
- **Cristalino**: Lente ubicada a continuación de la pupila y que se ajusta según la distancia a la que se quiere enfocar.
- **Retina**: Tejido sensible a la luz que se encuentra detrás del cristalino. Está formado por conos y bastones, células fotosensibles.

En la figura 2.1 (página 6) podemos apreciar de manera más visual su disposición.

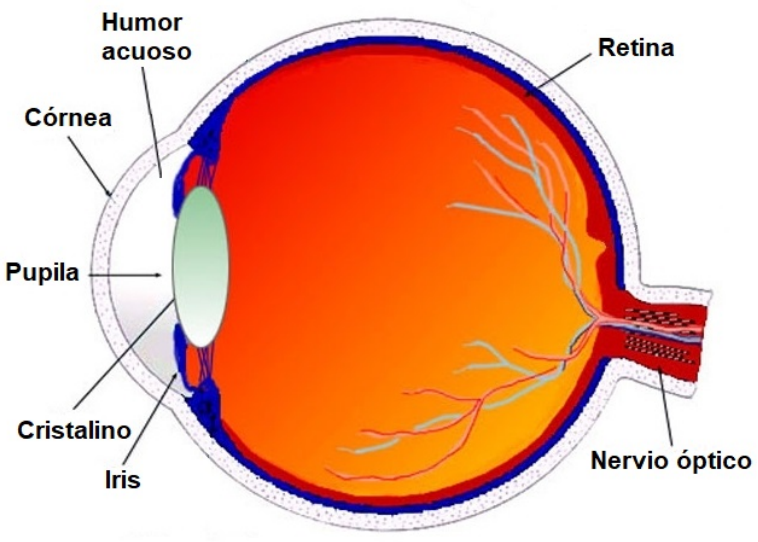

Figura 2.1: Anatomía del ojo humano

El funcionamiento básico del ojo humano consiste en que la luz entra en el globo ocular a través de la pupila y, atravesando el cristalino, incide sobre la retina, donde los conos y los bastones transforman su energía lumínica en impulsos eléctricos que envían al cerebro a través del nervio óptico.

Es relevante destacar que tanto la **cámara anterior**, región comprendida entre la córnea y el iris, como la **cámara posterior**, región comprendida entre el iris y el cristalino, están rellenas de un líquido conocido como **humor acuoso**. La presión en estas regiones, llamada **presión intraocular**, es indicativa de diferentes afecciones relacionadas con el correcto funcionamiento del ojo, véase el glaucoma (sección 2.1.3, página 9).

Por otro lado, conviene saber que, a la región formada por la retina y la unión de esta con el nervio óptico se la conoce como **fondo del ojo** o fondo ocular. Es esta región la que es representada en las retinografías (sección 2.2.1, página 10) y en la que se puede ver de forma relativamente clara el aumento de la presión intraocular antes mencionado.

2.1.2 El disco óptico

El **disco óptico**, también conocido como papila óptica o punto ciego, es una región de la retina carente de conos y bastones de forma aproximadamente circular, a través de la cual convergen al exterior del globo ocular, en concreto al nervio óptico, los axones de las neuronas (células ganglionares) de la propia retina.

Dentro del disco óptico encontramos otra región diferenciada del resto que se conoce como **copa** o cúpula. La parte del disco óptico no perteneciente a la cúpula es denominada **anillo neurorretiniano**.

En la figura 2.2 (página 8) podemos ver en más detalle una imagen del fondo ocular. El contorno azul delimita la región del disco óptico. El contorno verde delimita la región de la copa.

Para entender las motivaciones de este proyecto, es necesario saber que el ratio entre el diámetro vertical de la copa y el diámetro vertical total del disco óptico es uno de los indicadores que ayudan a la hora de diagnosticar el glaucoma (sección 2.1.3, página 9), por lo que se justifica la exploración de diferentes técnicas para la obtención de la mayor cantidad de información posible sobre esta región del fondo del ojo. En concreto, uno de los campos más explorados en este aspecto es el de la segmentación del disco ocular, separando copa y anillo neurorretiniano. Así, al automatizar este proceso de segmentación, resultaría trivial obtener el ratio antes mencionado.

Figura 2.2: Imagen del fondo ocular y estructura del disco óptico

2.1.3 Afecciones oculares: el glaucoma

El **glaucoma** es una enfermedad ocular cuya principal causa es el aumento de la presión intraocular debido a una falla en el drenaje del humor acuoso. Es una afección degenerativa que en sus fases iniciales no presenta muchos síntomas, pero que en estados más avanzados conlleva gradualmente a la pérdida de visión, entre otras consecuencias. Esta falta de sintomatología en los estadios iniciales y las graves consecuencias que produce ponen en relieve la importancia de encontrar métodos de diagnóstico temprano más eficaces y accesibles.

Para el diagnóstico de glaucoma, existen dos indicadores relevantes:

- Presiones intraoculares mayores a 21 mmHg.
- Aumento de la profundidad en la copa del disco óptico, con la consecuente variación del ratio de diámetros verticales copa-disco, cuyo valor normal es de 0.3 y que aumenta en presencia de la enfermedad.

En nuestro caso, nos centraremos en el segundo indicador. Por un lado, en la primera etapa del proyecto, obtendremos el mapa de profundidad, mapa que ya puede arrojar información sobre la presencia de glaucoma. En la segunda etapa utilizaremos el aprendizaje obtenido en esta primera tarea para tratar de segmentar disco y copa, de nuevo orientándonos al diagnóstico de esta enfermedad.

En la figura 2.3 (página 11) podemos apreciar que, una vez obtenida la segmentación, es incluso apreciable a simple vista (en algunos casos) la diferencia entre un ojo enfermo de uno que no lo está. La imagen de la izquierda se corresponde a una imagen del fondo ocular de un ojo afectado por la enfermedad del glaucoma en la cual se han anotado las segmentaciones de disco, en color azul, y copa, en color verde. La imagen de la derecha es una imagen de las mismas características pero de un ojo sano. Puede apreciarse que la copa ocupa mayor proporción del disco en el caso de la imagen de la izquierda.

2.2 Modalidades de imagen

La información relativa al fondo ocular en la que se basa el diagnóstico de determinadas afecciones oculares como el glaucoma se encuentra generalmente representada en forma de imágenes.

Dentro de los tipos de imágenes de los que podemos sacar información sobre esta región, podemos destacar dos para la comprensión del proyecto, pues serán los que utilizaremos en nuestra experimentación: la **retinografía** y la **tomografía de coherencia óptica u OCT**.

2.2.1 Retinografía

Este tipo de imágenes médicas consisten simplemente en fotografías detalladas del fondo del ojo. Su obtención se considera no invasiva pues no es necesaria ninguna intervención en el paciente, aunque sí es necesario en la mayoría de los casos aplicar unas gotas de un agente midriático en el ojo para conseguir una dilatación de la pupila, lo que provoca mayor entrada de luz permitiendo visualizar una mayor extensión de la retina y obtener fotografías más claras.

Una de las principales utilidades de las retinografías es que permiten la visualización del disco óptico.

Existen diversos tipos de retinografías atendiendo tanto a criterios de obtención (invasivas o no invasivas) como de representación de la información en sí. En lo que concierne a este proyecto, conviene diferenciar entre las **retinografías monoculares** o no estereoscópicas y las **retinografías estereoscópicas**.

Retinografía monocular

Las **retinografías monoculares** consisten en una única imagen del propio fondo del ojo, obtenidas generalmente mediante cámaras digitales de alta resolución. Son las retinografías más comunes y fáciles de obtener.

En la figura 2.4 (página 11) se muestra una retinografía monocular convencional. Podemos apreciar cómo la región del disco óptico es claramente visible en la imagen.

Retinografía estereoscópica

Las **retinografías estereoscópicas** están basadas en técnicas de visualización binocular.

Las **imágenes estereoscópicas** permiten representar objetos y ubicaciones tridimensionales en dos dimensiones y sobre un medio plano. Para ello, se basan en el funcionamiento del propio ojo humano, que toma dos imágenes de su campo de visión desde puntos de vista diferentes de forma que el cerebro es capaz de interpolar la profundidad de lo que se está observando a partir de estas dos imágenes desfasadas.

El método de obtención queda detallado en la figura 2.5 (página 13). En la imagen podemos apreciar cómo se procede a la hora de obtener imágenes estereoscópicas. Se parte de dos puntos de vista diferentes y se toman las imágenes de manera que al ser proyectadas frente a los ojos de manera adecuada produzcan efecto tridimensional.

Así, las retinografías estereoscópicas no son más que un par de imágenes desfasadas del mismo fondo ocular de manera que es posible obtener de ellas información dimensional de profundidad.

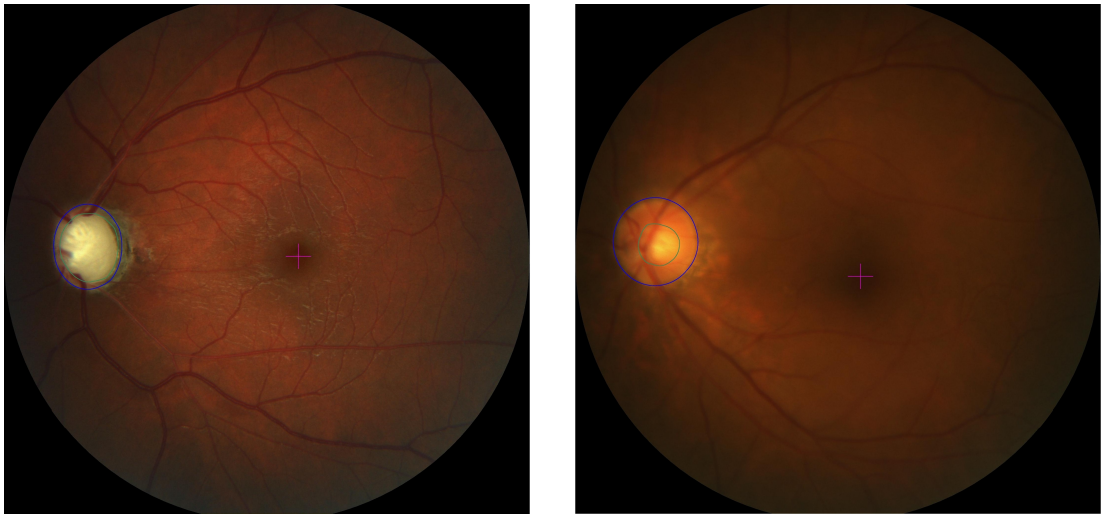

Figura 2.3: Comparativa de imagenes del fondo ocular de un ojo enfermo (izquierda) y un ojo sano (derecha)

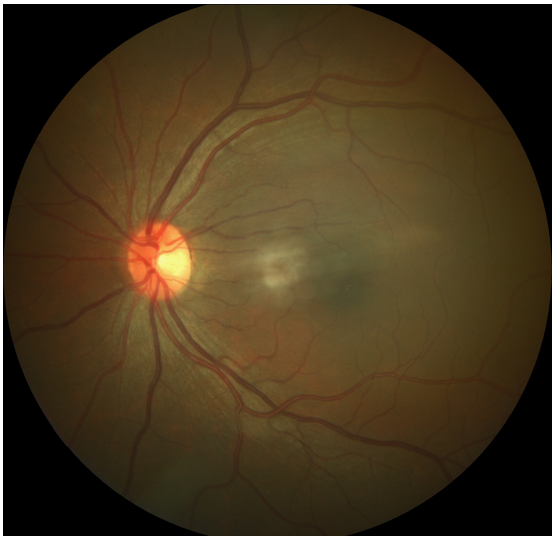

Figura 2.4: Retinografía monocular

Estas retinografías son obtenidas también mediante cámaras digitales de alta resolución, pero que han de tener la capacidad de obtener dos imágenes desplazadas de manera secuencial y que son conocidas como **cámaras de fondo de ojo**.

En la figura 2.6 (página 13) se muestra una retinografía estereoscópica.

El uso de retinografías estereoscópicas está menos extendido debido al mayor coste que supone su obtención a pesar de que la información que proporcionan, relativa sobre todo a los valores de profundidad del fondo ocular, es muy significativa a la hora de diagnosticar diferentes enfermedades oculares, destacando la enfermedad del glaucoma. Por ello, resulta interesante explorar diferentes opciones para la obtención de este tipo de imágenes o de la información que se deriva de ellas mediante otras técnicas que no impliquen la obtención de las retinografías estereoscópicas de forma directa.

2.2.2 Tomografía de coherencia óptica

La **tomografía de coherencia óptica** u OCT es una modalidad de imagen no invasiva que permite ver en detalle la retina y el nervio óptico. Aportan una visión de las capas que conforman la retina en forma de sección transversal, diferente a la visión que aportan las retinografías.

Su funcionamiento se basa en irradiar el fondo ocular con luz infrarroja. Al penetrar el tejido ocular, la luz se dispersa, de forma que se puede comparar con un haz de referencia para reconstruir una imagen del fondo ocular mediante los tiempos de retorno de la luz dirigida al ojo. Este procedimiento se conoce como **interferometría**.

De este procedimiento se obtienen imagenes de gran resolución de esta parte de la anatomía ocular como la que podemos ver en la figura 2.7 (página 13). Estas imágenes son útiles, al igual que las retinografías, en el diagnóstico de enfermedades oculares. Sin embargo, su uso está menos extendido pues los dispositivos necesarios para obtenerlas son más caros y además ocupan gran cantidad de espacio. Por otro lado, es necesario que el paciente mantenga la vista en un punto fijo durante determinado período de tiempo, por lo que para poder obtener una OCT es necesario que el paciente colabore, siendo imposible de realizar a pacientes no colaborativos como, por ejemplo, personas inconscientes.

2.3 Aprendizaje automático

En nuestro día a día, nos enfrentamos a la resolución de problemas de muy diferente índole. Algunos de estos problemas son fácilmente expresables en lenguaje máquina y reductibles a reglas, funciones y variables, por lo que la programación clásica puede darles solución y automatizarla de manera relativamente simple, sin mayor necesidad que traducir el conoci-

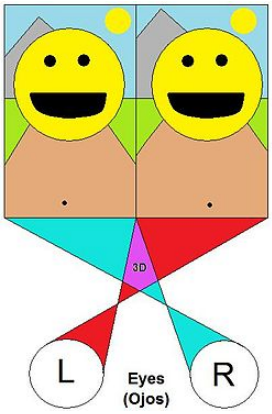

Figura 2.5: Reconstrucción tridimensional de imágenes estereoscópicas

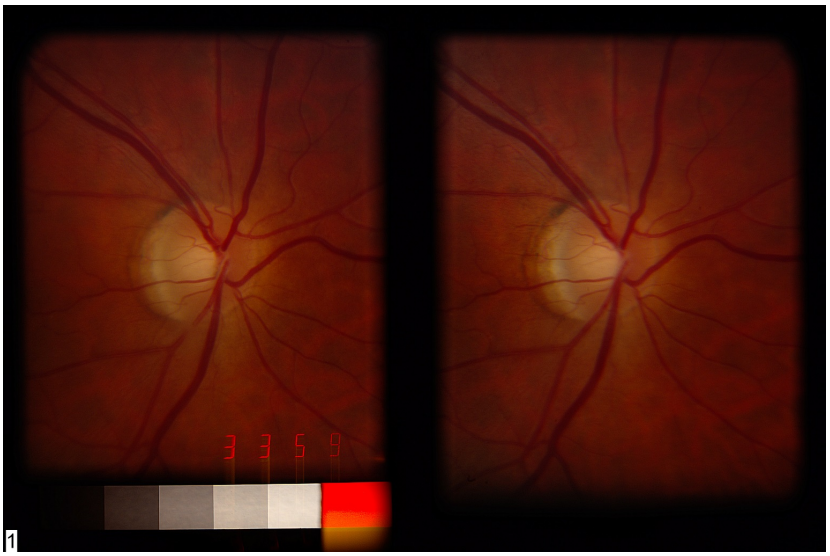

Figura 2.6: Retinografía estereoscópica

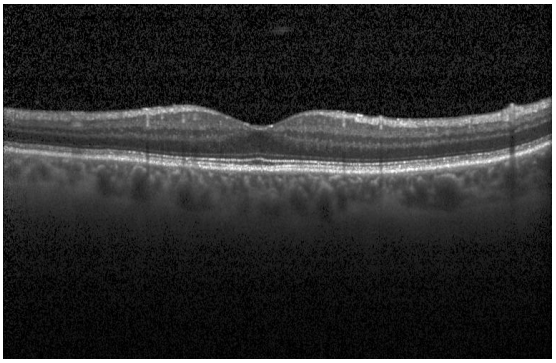

Figura 2.7: Tomografía de coherencia óptica

miento por parte del programador a un lenguaje de programación. Sin embargo, existen otro tipo de problemas que resolvemos de manera más inconsciente o para los cuales el conocimiento necesario para resolverlos no es fácilmente expresable en un lenguaje utilizable por las máquinas. Por ejemplo, no es sencillo explicar qué proceso de razonamiento estamos siguiendo a la hora de reconocer el rostro de un familiar en una fotografía en la que salen muchos más rostros. Para este tipo de problemas, es necesario cambiar el paradigma de programación y plantearnos si es posible que, en lugar de programar explícitamente la resolución, podríamos hacer que la máquina aprendiese a resolver el problema de forma "independiente". Nace de esta inquietud el campo del **aprendizaje automático** o **machine learning**. Este campo de la computación, perteneciente a la rama de la inteligencia artificial, busca encontrar algoritmos que sean capaces de encontrar patrones y comportamientos subyacentes a conjuntos de datos. Podríamos decir que los algoritmos de machine learning aprenden de la experiencia, pues utilizan los datos que se les proporcionan para ir refinando sus resultados, de forma que adquieren capacidades para las que no estaban programados inicialmente.

Dentro de este campo existen diferentes estrategias para llegar al objetivo planteado. En lo que concierne a nuestro proyecto, hablaremos únicamente de dos de ellas: el **aprendizaje supervisado** (sección 2.3.1, página 14) y el **aprendizaje auto-supervisado** (sección 2.3.2, página 15).

2.3.1 Aprendizaje supervisado

El **aprendizaje supervisado** utiliza un conjunto de datos en el que se dispone tanto de las entradas del algoritmo como de las salidas deseadas del mismo, de manera que el trabajo a realizar por estos algoritmos es obtener una relación entre ambas. Se habla de aprendizaje supervisado cuando las salidas deseadas del conjunto de datos de entrenamiento han sido etiquetadas a mano por un experto, de manera que éste actúa de supervisor. Las etiquetas no son más que los valores esperados en la salida dada una determinada entrada y pueden ser valores discretos (problemas de clasificación) o valores continuos (problemas de regresión).

Hoy en día, la investigación en este paradigma de aprendizaje automático es muy relevante, pues cada vez se disponen de mayores cantidades de datos etiquetados. En concreto, en el ámbito de la medicina, podemos destacar diferentes motivos por los que el aprendizaje automático supervisado resulta muy útil. Por un lado, el diagnóstico de muchas enfermedades suele estar asociado al análisis de imágenes médicas por parte de expertos, de manera que tenemos antecedentes de datos de entrada y salida deseada que se han ido almacenando a lo largo de los años. Sin embargo, muchas veces este proceso de diagnóstico responde a la experiencia del profesional médico más que a criterios fácilmente expresables o conocimiento transferible a un programa convencional. Por otro lado, es posible que existan patrones o relaciones en los conjuntos de datos médicos de los que disponemos que aún no hayan sido

encontrados o puestos de manifiesto de forma clara y este tipo de algoritmos de aprendizaje automático pueden ayudar a la hora de identificarlos.

Además, junto a estos motivos, podemos destacar la mejora en los métodos de obtención de datos a lo largo de los años, como por ejemplo la sustancial mejora en las imágenes obtenidas por las cámaras digitales, cada vez de mayor calidad. También resulta relevante la mejora en el hardware de las máquinas encargadas de procesar este tipo de algoritmos, tanto a nivel de memoria como a nivel de capacidad computacional, así como la existencia de lenguajes y librerías cada vez de más alto nivel, que facilitan mucho las tareas de diseño y implementación de los algoritmos en sí. Por todo esto, resulta evidente por qué este tipo de tecnologías son cada vez más utilizadas en un amplio abanico de ámbitos y más específicamente en el ámbito de la medicina, permitiendo explotar esta creciente cantidad de datos etiquetados para la obtención de algoritmos capaces de dar soporte a las tareas de diagnóstico, entre otras aplicaciones.

En la tarea de segmentación se utilizará este tipo de aprendizaje, pues el conjunto de datos que se usará (sección 4.7.1, página 44) consta de salidas etiquetadas a mano por expertos.

2.3.2 Aprendizaje auto-supervisado

El **aprendizaje auto-supervisado** es un término más reciente que nace como respuesta a la aparición de un nuevo tipo de aprendizaje en el que si bien el entrenamiento consiste, al igual que en el aprendizaje supervisado, en aprender a obtener una salida deseada etiquetada a priori a través de la entrada, las etiquetas de esta salida deseada no están definidas manualmente por un agente externo, es decir, no existe un supervisor como tal. Por eso, al ser las etiquetas calculadas de forma automática partiendo directamente de los datos, hablamos de aprendizaje auto-supervisado. De esta manera, es un tipo de aprendizaje aplicable a datos no etiquetados.

En las tareas de pre-entrenamiento de predicción de información de profundidad, será éste el tipo de aprendizaje que utilicemos, pues si bien contamos con las salidas deseadas, no son salidas etiquetadas a mano, las etiquetas consisten directamente en parte de los datos disponibles.

2.3.3 Aprendizaje profundo

El **aprendizaje profundo** o ***deep learning*** es una rama del *machine learning* que aparece debido a la complejidad de encontrar y definir representaciones y características relevantes a partir de los conjuntos de datos para la resolución de determinadas tareas. Si bien en otras ramas del aprendizaje automático estas características son definidas y establecidas a priori, el aprendizaje profundo se centra en desarrollar algoritmos que sean capaces de detectar y aprender por sí mismos las representaciones relevantes, sin necesidad de ser construidos "a

mano" y especificados explícitamente por el programador. De esta manera, este tipo de algoritmos utilizan como entrada los datos crudos y, mediante una etapa de entrenamiento, encuentran los patrones necesarios y sus representaciones para resolver la tarea que se les plantee.

En la figura 2.8 (página 16) podemos ver un diagrama de Venn en el que se muestra la ubicación del aprendizaje profundo dentro del campo de la inteligencia artificial.

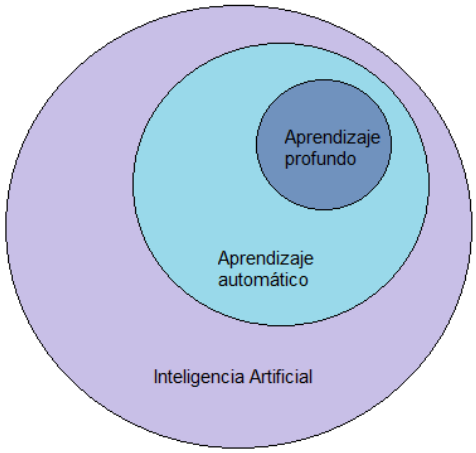

Figura 2.8: Diagrama de Venn del campo de la inteligencia artificial

El nombre de este tipo de aprendizaje automático viene de que los modelos que utiliza están compuestos por sucesivas capas de unidades de procesamiento, generando profundidad en dicho modelo. A nivel teórico, la base de esta jerarquía profunda es que, cuando un patrón es complejo, generalmente puede dividirse en patrones más simples, de manera que en cada capa del modelo vayamos aumentado el nivel de abstracción.

En la figura 2.9 (página 18) podemos ver un ejemplo de modelo de aprendizaje profundo, en el que se puede observar dicha profundidad. El modelo, además de la capa de entrada, consta de otras 5 capas más.

Para comprender mejor los detalles técnicos de las metodologías de aprendizaje profundo utilizadas en el proyecto, en los apéndices A (página 83) y B (página 89) se explican los conceptos relativos a las **redes de neuronas artificiales** y a las **redes neuronales convolucionales**, que serán las que se empleen en la experimentación.

2.3.4 Data augmentation

Uno de los principales problemas dentro del mundo del análisis de las imágenes médicas por medio del aprendizaje profundo es que la aplicación de este tipo de algoritmos requiere enormes cantidades de datos anotados con el fin de aprender correctamente los patrones y características subyacentes a las imágenes. Sin embargo, disponer de tan elevado número de

patrones de ejemplo no siempre es posible, por lo que es necesario recurrir a técnicas que ayuden a subsanar la falta de ejemplares.

Las técnicas de **data augmentation** consisten en aumentar el número de ejemplares de manera artificial modificando ligeramente las imágenes de ejemplo de las que se disponía en un primer lugar. Entre las modificaciones que se realizan a las imágenes podemos señalar las transformaciones geométricas, el volteo (flipping), modificaciones de color, introducción de ruido, rotaciones, etc. En la figura 2.10 (página 18) podemos ver un ejemplo de aplicación de estas operaciones.

Añadiendo las imágenes modificadas al conjunto de entrenamiento podemos aumentarlo de manera significativa sin necesidad de obtener nuevos ejemplares, ayudando no solo a entrenar la red al aumentar el número de patrones de ejemplo, sino también haciendo el aprendizaje más robusto al overfitting al presentar imágenes modificadas con características que podían no estar presentes en el conjunto de entrenamiento inicial.

2.3.5 Transfer learning

Tal y como hemos definido hasta ahora el aprendizaje profundo, es fácil reconocer que el conocimiento que adquieren las redes neuronales artificiales se encuentra en los pesos o valores de los filtros convolucionales que las propias redes van aprendiendo a lo largo del proceso de entrenamiento.

Es importante definir la inicialización de estos valores para los filtros, pues dependerá de ello los resultados que se obtengan durante el proceso de entrenamiento, debido a las características del algoritmo que se utiliza para optimizar sus valores, generalmente basado en **descenso del gradiente**. En otras palabras, la minimización que se obtiene finalmente de la función de coste depende en gran medida de los valores iniciales que se le dan a los pesos de la red neuronal artificial.

Generalmente, la primera aproximación que se lleva a cabo a la hora de inicializar estos pesos es darles un valor aleatorio cercano a cero. Sin embargo, dentro de un mismo dominio de aplicación existen múltiples tareas que podrían ser resueltas mediante aprendizaje profundo, por lo que es relevante poder transferir el conocimiento aprendido por una red para la resolución de una determinada tarea a otra red para una tarea diferente pero dentro de este mismo dominio, sin partir de una inicialización aleatoria.

Este tipo de metodologías de transferencia de conocimiento se conocen como **transfer-learning** y su validez ha sido ampliamente demostrada como podemos ver en la revisión hecha por *Chuanqui Tan et al.* en 2018, [2] .

En este proyecto lo que haremos será entrenar una red para la tarea de predicción de la

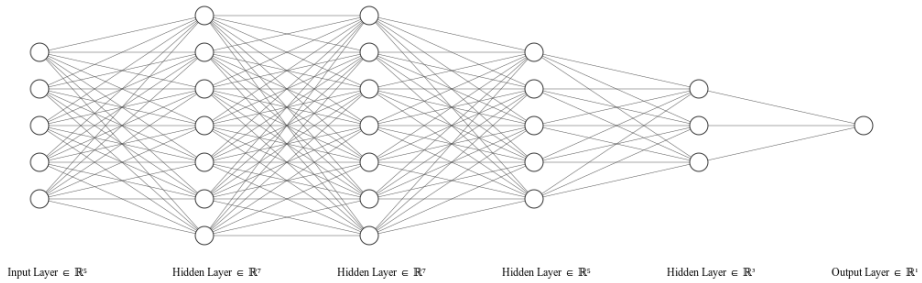

Figura 2.9: Representación de un modelo de aprendizaje profundo

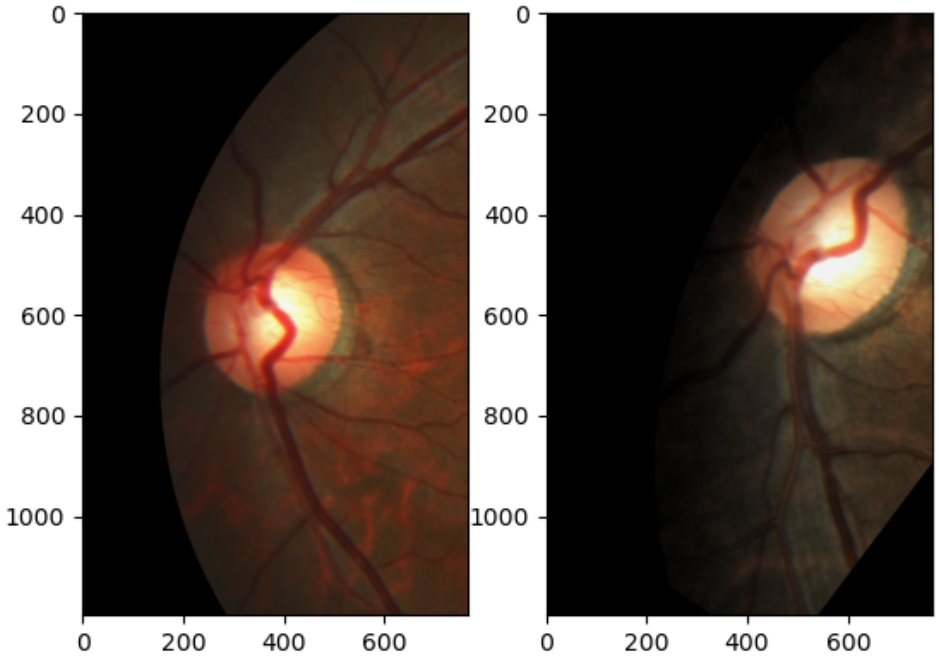

Figura 2.10: Ejemplo de aplicación de operaciones de *data augmentation*

profundidad y, una vez entrenada esta red, utilizaremos los valores de los filtros convolucionales aprendidos por la misma como partida para el entrenamiento de la segunda red en la tarea de segmentación. Nuestro objetivo final es demostrar que esta transferencia del conocimiento ayuda a mejorar los resultados y la eficiencia de la red en la segunda tarea frente a entrenarla de cero partiendo de filtros aleatorios.

2.4 Estado del arte

En 2020 fue publicada una revisión sobre diferentes técnicas utilizadas hasta esa fecha en el problema de la segmentación de disco y copa a partir de imágenes del fondo ocular, [3].

Por un lado, dentro de las diferentes aproximaciones que se presentan, podemos destacar los estudios que utilizan técnicas no relacionadas con el *deep learning*, como el de *Rangaraj*, que utiliza filtros de Gabor y retratos de fase, [4]; el de *S. Maheshwari*, basado entre otras cosas en máquinas de soporte vectorial, [5]; o el de *M. Khunger et al.*, utilizando patrones locales binarios y el algoritmo de Daugman (para la localización del iris), [6]. Por otro lado, en lo concerniente a trabajos relacionados con el proyecto que se plantea, existen estudios que utilizan diferentes arquitecturas de redes neuronales ya existentes como la ResNet, [7] [8], la propia U-Net, [8] [9], o la DenseNet, [10] . Estos estudios se basan en la utilización y modificación (*tuning*) de estas arquitecturas pre-existentes para abordar la tarea de segmentación de disco y copa, pues resultan ser arquitecturas apropiadas para tareas de segmentación semántica, como se demuestra en los artículos mencionados. También existen estudios que plantean arquitecturas nuevas específicas para el problema en sí, [11].

Además de los estudios mencionados en la revisión, cabe mencionar trabajos más recientes que también utilizan técnicas de *deep learning* en su desarrollo. Podemos destacar los realizados por *H.N. Veena et al.*, basado en la utilización de dos redes con arquitecturas específicas, una para segmentar el disco y otra para segementar la copa, [12]; *Y. Jiang et al.*, que, asumiendo que las formas de disco y copa son elípticas, plantea una nueva arquitectura llamada JointRCNN, la cual utiliza también dos redes, una para segmentar el disco y otra la copa, pero integra los resultados dentro de su propia arquitectura de forma que la predicción para el disco se utiliza en la predicción de la copa, pues se sabe a priori que la copa está contenida en el disco, [13]; *D. Natarajan et al.*, estudio en el cual si bien no se utiliza una arquitectura *deep learning* para segmentar disco y copa, lo cual se realiza mediante segmentación de superpíxeles y un filtro *C-Means* difuso, sí que se utiliza para clasificar el resultado de esta segmentación como enfermo o sano para la enfermedad de glaucoma, [14]; *M.K. Khan et al.*, que usa para la segmentación una arquitectura *deep learning* basada en estructura *encoder-decoder*, como la arquitectura U-Net (sección 4.2, página 30), pero añadiendo también una gran memoria a corto plazo (LSTM) bidireccional, arquitectura conocida como M-Net, [15]; y *H. Almuba-*

rak et al., que utiliza una red convolucional basada en regiones (RCNN), redes ampliamente utilizadas en problemas de detección de objetos, a través de dos fases en las cuales primero se detecta y recorta la imagen de entrada al rededor del nervio óptico y una vez recortada la imagen ésta se utiliza como entrada para la posterior segmentación de disco y copa, [16].

En el presente estudio, si bien también se partirá de una arquitectura existente, U-Net, el elemento diferenciador consiste en la utilización de información complementaria multimodal en el entrenamiento, pues se parte de una red entrenada para una tarea determinada, en este caso la predicción de información de profundidad a partir de una única imagen retinográfica, y después se utiliza el conocimiento obtenido para comenzar el entrenamiento de la red que realizará la tarea de segmentación de copa y disco, pero sin utilizar explícitamente la información de profundidad obtenida como entrada en la segunda red. Existen otros estudios que realizan una aproximación similar, como es el caso del realizado por *A. Hervella et al.*, [17], en el cual se utiliza un pre-entrenamiento de predicción de profundidad a través de angiografías, otra modalidad de imagen del fondo ocular. Sin embargo, no existe ningún estudio que utilice información de profundidad obtenida a través de imágenes estereográficas y OCTs como pre-entrenamiento para la tarea de segmentación.

Capítulo 3

Metodología y planificación

En este capítulo se planteará el método de trabajo y la planificación del proyecto, desde la planificación temporal hasta la planificación y coste de recursos, tanto humanos como materiales.

3.1 Método de trabajo

Se partirá del estudio del dominio de aplicación del proyecto para comprender los detalles del mismo, así como de las metodologías y arquitecturas de aprendizaje profundo más recientes y con mejores resultados utilizadas en problemas similares, haciendo una revisión general del estado del arte. Como siguiente paso, se abordará el diseño de la resolución de las tareas planteadas a través de la arquitectura seleccionada, en este caso la U-Net. Se plantea una metodología de desarrollo iterativa e incremental en la cual poder establecer un proyecto base e irlo modificando sucesivamente en diferentes iteraciones, pues se considera que es el método de trabajo que mejor se ajusta a las necesidades del proyecto por ser éste susceptible a muchas variantes que pueden ser abordadas en diferentes iteraciones y comparadas entre ellas para mejorar los resultados finales. Por último, se realizarán los experimentos también de forma incremental, siguiendo el diseño planteado, y se estudiarán sus resultados para extraer las conclusiones pertinentes, utilizando diferentes representaciones de los resultados para determinar la validez o no validez de la premisa planteada en el proyecto, así como de la arquitectura implementada.

3.2 División de tareas

Para una correcta planificación del proyecto es necesario dividir este en tareas de manera que podamos establecer una línea base de trabajo y hacer una predicción de costes y tiempos de realización. De esta manera, podremos repartir los recursos de manera eficiente y agilizar

la ejecución del proyecto. Consideramos las siguientes tareas como principales del proyecto, ordenadas secuencialmente:

- **Estudio del dominio de aplicación y del estado del arte**: El objetivo de esta tarea es obtener los conocimientos necesarios del dominio de aplicación para poder realizar las tareas planteadas entendiendo sus bases de manera que permitan flexibilidad a la hora de tomar decisiones sobre la forma de abordarlas. Además, se quiere conseguir un conocimiento suficiente del campo del aprendizaje profundo para poder estudiar y elegir la mejor arquitectura posible para los experimentos.

- **Diseño de la experimentación**: Etapa en la que se planteará conceptualmente la experimentación a llevar a cabo de forma concisa para poder comenzar su implementación y la consecuente ejecución de los experimentos a llevar a cabo. Se diseñarán tanto las arquitecturas a utilizar (partiendo de la arquitectura U-Net) como los propios experimentos que se quieren realizar con las variantes necesarias y se plantearán las maneras de estudiar sus resultados, de forma que el proyecto a realizar quede claramente cerrado.

- **Predicción del mapa de profundidad**: En esta tarea se quiere, partiendo de una implementación de la arquitectura U-Net, obtener predicciones de los mapas de profundidad partiendo de retinografías monoculares. Para ello, el objetivo es modificar los aspectos necesarios de la arquitectura, así como preparar las imágenes de entrada y elegir convenientemente los aspectos metodológicos relevantes. Es necesario completar esta tarea para poder realizar la tarea siguiente, pues depende directamente de sus resultados.

- **Segmentación (con y sin pre-entrenamiento)**: El objetivo de esta tarea es obtener la segmentación de disco y copa a partir de retinografías monoculares. Para ello, se realizarán dos experimentos: un entrenamiento desde cero y un entrenamiento utilizando los resultados de la tarea anterior (los valores aprendidos para los filtros convolucionales) como pre-entrenamiento.

- **Análisis de resultados y elaboración de conclusiones**: Una vez realizada la experimentación, se propone analizar los resultados obtenidos y arrojar conclusiones sobre la validez de la metodología empleada, así como comparar los resultados de los dos experimentos de segmentación, comprobando si existe mejora con el pre-entrenamiento.

- **Elaboración de la memoria**: Una vez obtenidos los resultados, el objetivo es escribir la memoria del proyecto. Se organizará y redactará de forma ordenada todo lo relacionado con la misma.

- **Predicción de par estereográfico**: El objetivo de esta tarea es obtener el par estereográfico de una retinografía monocular. Partiremos de la misma arquitectura que para su tarea análoga, modificando los detalles pertinentes para esta tarea, pues las características de las salidas son diferentes. Es necesario, de nuevo, completar esta tarea para realizar la siguiente, pues depende de sus resultados directamente.

- **Segmentación (con y sin pre-entrenamiento)**: El objetivo es únicamente entrenar la misma red de segmentación que en la tarea de segmentación previa pero utilizando los pesos obtenidos en la red de la tarea anterior, para comparar los resultados con los obtenidos utilizando pesos al azar y utilizando los pesos obtenidos de la predicción del mapa de profundidad.

- **Análisis de resultados y elaboración de conclusiones**: Idéntica en ejecución a la otra tarea de análisis de resultados.

- **Compleción de la memoria con los nuevos resultados**: Se añadirán a la memoria los nuevos resultados y conclusiones obtenidos, así como los aspectos metodológicos de este último experimento.

Las tareas se plantean de manera que se pueda tener un primer experimento base, consistente en el entrenamiento de la red de segmentación con los pesos obtenidos de la red de predicción del mapa de profundidad y poder extraer conclusiones de este primer experimento. Una vez completada esta primera fase, se pasaría a realizar el mismo experimento pero utilizando la red de predicción del par estereográfico. Se abordará de esta manera porque el entrenamiento de las redes consume tiempo y recursos computacionales de forma exigente y es conveniente tener una primera base sobre la que apoyar el proyecto antes de continuar con la experimentación.

Esta estructura iterativa queda clarificada en el esquema de la figura 3.1 (página 23).

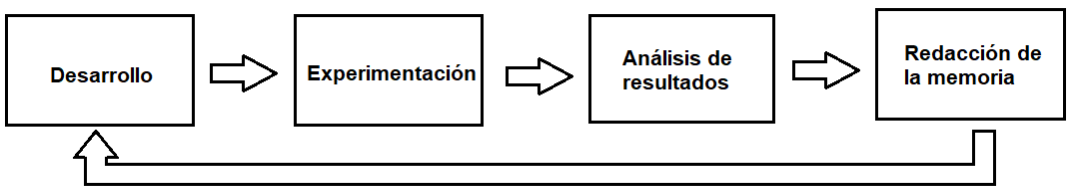

Figura 3.1: Esquema de planificación del proyecto

Por último, debido a los resultados obtenidos, se plantea finalmente una última etapa en la que se repetirán los experimentos realizados, en caso de tener tiempo suficiente, pero con

modificaciones arquitectónicas que se explicarán en detalle más adelante (capítulo 4, página 4).

3.3 Planificación temporal

La planificación temporal inicial de las tareas descritas en la sección 3.2 (página 21) se detalla en el diagrama de Gantt de la figura 3.2 (página 24).

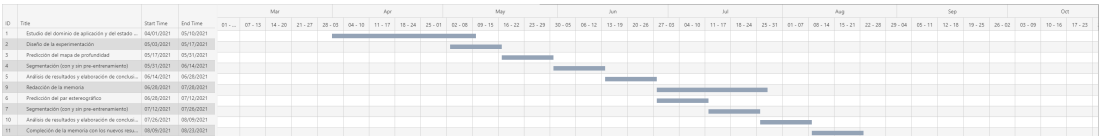

Figura 3.2: Diagrama de Gantt de la planificación temporal del proyecto

Se puede observar que el proyecto comienza su desarrollo el día 1 de abril de 2021 y termina el 23 de agosto de 2021.

Estimando que se trabaja en el desarrollo del proyecto entre 2 a 4 horas diarias (exceptuando los domingos) durante todas las semanas que abarca la planificación, tenemos un total de 124 días de trabajo que se traducen en aproximadamente 372 horas de trabajo totales por parte del alumno.

Para el trabajo por parte de los directores del proyecto se tomará como referencia una parte proporcional para guiar al alumno y para la revisión del proyecto de un 10% de la duración total del mismo, obteniendo un total de aproximadamente 40 horas.

Esta planificación inicial fue alterada a lo largo del desarrollo del proyecto debido a diferentes desviaciones que serán mencionadas en el apartado de estimación de costes (sección 3.5, página 26).

3.4 Planificación de recursos

Para la realización de este proyecto hemos dispuesto de recursos tanto materiales como humanos. En este apartado describiremos estos recursos para una posterior estimación de su coste.

3.4.1 Recursos materiales

- Ordenador portátil MSI GT72VR 7RE Dominator Pro
  - Procesador: Intel Core i7-7700HQ CPU @ 2.80GHz; 4 núcleos, 8 subprocesos.

  - Memoria RAM: 16 GB
  - Sistema operativo: Windows 10 Pro
  - Tarjeta gráfica: NVIDIA GeForce GTX 1070
  - Almacenamiento: 920 GB HHD, 237 GB SSD

- Ordenador portátil MSI 262XBE Gaming GE60

  - Procesador: Intel Core i7-4710HQ CPU @ 2.50GHz; 4 núcleos, 8 subprocesos.
  - Memoria RAM: 8 GB
  - Sistema operativo: Ubuntu 18.04
  - Tarjeta gráfica: NVIDIA GeForce GTX 870M
  - Almacenamiento: 315 GB HHD, 254 GB SSD

- Servidor de cómputo GPU facilitado por el grupo VARPA (CITIC)

  - Procesador: Intel Xeon CPU E5-2650 v4 @ 2.20GHz; 48 cores.
  - Memoria RAM: 128 GB
  - Tarjeta gráfica: NVIDIA Tesla K80 (24GB)

Se utilizará el ordenador portátil *MSI GT72VR 7RE Dominator Pro* para realizar la ejecución de las redes predicción de información de profundidad, así como para la programación y redacción de la memoria. Por su parte, el ordenador portátil *MSI 262XBE Gaming GE60* se usará para almacenar los datos y realizar las conexiones con el servidor del grupo VARPA desde Ubuntu, así como para ciertas tareas de programación. Por último, el servidor VARPA se utilizará para ejecutar las redes de segmentación, siendo necesaria su alta capacidad computacional para una ejecución más eficiente de las mismas.

3.4.2 Recursos software

- Sublime Text 3
- Pytorch 1.8.0
- Numpy 1.9.2
- Scikit-image 0.17.2
- Matplotlib 3.3.2

El editor de texto Sublime Text 3 se utilizó para las labores de programación del proyecto.

La librería PyTorch (versión 1.8.0) fue utilizada para la implementación del código relacionado con las redes de neuronas artificiales.

Las librerías Numpy, Scikit-image y Matplotlib se utilizaron para gestionar el tratamiento de los datos (imagenes de entrada, salida, etc) y para la representación de los mismos.

3.4.3 Recursos humanos

Los recursos humanos de los que se dispuso durante la realización del proyecto están formados por los directores del proyecto y el alumno.
Las tareas del alumno son todas las relacionadas con la realización del proyecto, es decir, las descritas en la sección 3.2 (página 21).

Los directores del proyecto se encargarán de la supervisión y guía del alumno a lo largo de la consecución de las tareas asignadas al mismo.

3.4.4 Conjuntos de datos

Consideraremos los conjuntos de datos utilizados a lo largo de la realización del proyecto un recurso a tener en cuenta. Concretamente, los conjuntos de datos utilizados son:

- **ISPIRE-stereo**: Compuesto por 30 retinografías estereográficas acompañadas de su mapa de profundidad basado en información obtenida de OCTs.
- **REFUGE**: Formado por 1200 retinografías monoculares acompañadas por su correspondiente anotación de la segmentación de disco y copa.

3.5 Estimación de costes

Consideramos que los recursos materiales no suponen ningún coste, pues son recursos privados (en el caso de los ordenadores portátiles) y disponibles antes de la elaboración del proyecto.
Por otro lado, tanto los recursos software como los conjuntos de datos son todos recursos disponibles de forma gratuita y de código abierto, por lo que tampoco suponen un coste para la realización del proyecto. En definitiva, únicamente tendremos en cuenta la estimación de costes derivada de los recursos humanos, la cual podemos ver en la tabla 3.1 (página 27).

Una vez terminado el proyecto, ha habido diferentes desviaciones que alteraron esta estimación inicial, de forma que finalmente los costes reales del mismo, que pueden observarse

	Horas x hombre	Salario	Coste
Director de proyecto	40	30€/h	2400 €
Alumno	372	20€/h	7440 €
		Total:	9840 €

Tabla 3.1: Tabla de estimación de costes de los recursos humanos

en la figura 3.2 (página 28), variaron respecto a los estimados. Las desviaciones más notorias que han afectado realmente a las horas finalmente computadas a la hora de realizar el proyecto son las siguientes:

- La tarea de diseño finalmente duró más tiempo del esperado por la necesidad de familiarizarse de forma más profunda con los conocimientos necesarios del dominio de aplicación, nunca utilizados antes por el alumno. Además, durante esta primera tarea hubo que dedicar tiempo a la realización del anteproyecto, que no fue contemplado en la planificación inicial. Se alargó aproximadamente una semana más, añadiendo 18 horas.

- La tarea de predicción del mapa de profundidad duró también más de lo esperado por ser la primera tarea en la que se realizó trabajo de implementación por lo que el alumno no estaba aún familiarizado con la librería pytorch y fue necesario un estudio exhaustivo del código de implementación de la arquitectura U-Net así como de otros elementos relevantes para el desarrollo del proyecto. Se alargó durante dos semanas más de las previstas, añadiendo 36 horas a la planificación inicial.

- Las tareas posteriores de implementación y ejecución de la experimentación, es decir, las tareas de la primera segmentación, predicción del par estereográfico y la segunda segmentación, fueron más cortas de lo esperado debido a que al ya haber adquirido el conocimiento necesario para las modificaciones arquitectónicas y la ejecución de las redes en la tarea de prediccion de profundidad, se agilizó bastante el proceso. Se acortó en total entre las tres tareas una semana, es decir, 18 horas.

- Las tareas de realización de la memoria (tanto la inicial como su compleción) y además la posterior revisión de la misma, fueron más largas de lo esperado, añadiendo unas dos semanas más a las horas totales, es decir 36 horas.

En total pues, se añadieron en total 72 horas más a la elaboración del proyecto. En consecuencia, también hubo necesidad de más trabajo por parte de los directores del proyecto.

	Horas x hombre	Salario	Coste
Director de proyecto	45	30€/h	2700 €
Alumno	444	20€/h	8880 €
		Total:	11580 €

Tabla 3.2: Tabla de costes reales de los recursos humanos

Capítulo 4

Materiales y métodos

En este capítulo se expondrán los detalles metodológicos del proyecto. Explicaremos la arquitectura seleccionada para la experimentación y sus características. Para cada subsistema presente en el sistema general planteado en el proyecto, descrito a continuación, explicaremos diferentes detalles arquitectónicos así como otras decisiones metodológicas de los mismos, incluidas las imágenes que utilizaremos para el entrenamiento.

4.1 Descripción general del sistema

El sistema que queremos construir tiene como objetivo final la obtención de la segmentación de disco y copa en una retinografía para el diagnóstico de glaucoma. Para la consecución de esta tarea se plantea el diseño de un sistema que conste de las siguientes partes:

- Subsistema de predicción de información de profundidad a partir de retinografía monocular.
- Subsistema de segmentación semántica de disco y copa a partir de retinografía monocular.

A su vez, para la construcción del primer subsistema, se plantean dos alternativas:

- Predicción del mapa de profundidad a partir de retinografía monocular.
- Predicción del par estereográfico a partir de retinografía monocular.

En la figura 4.1 (página 30) podemos ver un esquema general más detallado de la estructura de los sistemas presentes en el desarrollo del proyecto.

Cada uno de los bloques elementales presentes en el sistema estará constituido por una red convolucional con arquitectura U-Net (sección 4.2, página 30). Además, como se puede

ver en el esquema, el subsistema de predicción de información de profundidad a partir de retinografía monocular se utilizará como pre-entrenamiento para detectar patrones relevantes en la retinografía. Haciendo uso de *transfer-learning*, este conocimiento adquirido en la primera tarea servirá como punto de partida para el refinamiento de la segmentación semántica de disco y copa.

Este proceso de *transfer-learning* es sobre el cuál quiere obtenerse información, comparando el resultado del subsistema de segmentación semántica de disco y copa sin uso del conocimiento transferido frente al resultado del mismo subsistema pero con uso del conocimiento proviniente del otro subsistema.

De forma práctica, el *transfer-learning* no consistirá más que en usar los pesos aprendidos por alguna de las dos redes del primer subsistema en la red del segundo subsistema y ver si se obtiene alguna mejora frente a un uso de pesos aleatorio.

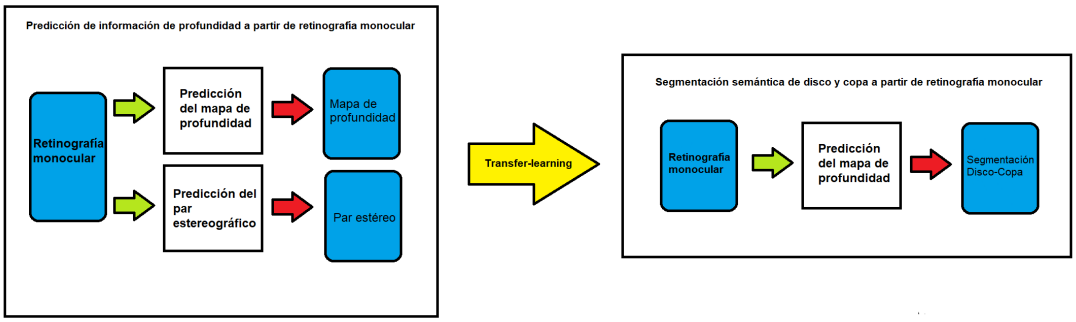

Figura 4.1: Esquema general del proyecto

4.2 Arquitectura U-Net

La arquitectura U-Net fue presentada por primera vez en *Olaf Ronneberger et al.*, [1]. Su objetivo principal era resolver el problema de la necesidad de disponer de enormes cantidades de patrones anotados para entrenar una red convolucional. De esta forma, plantea un tipo de arquitectura que, ayudada por técnicas de *data augmentation*, permite obtener muy buenos resultados con pocos datos etiquetados.

Esta arquitectura se basa principalmente en una estructura encoder-decoder. Dicha estructura está formada por dos caminos de datos, el *contracting path* y el *expansive path*, y tiene como finalidad extraer diferentes patrones y características de alto nivel de las imágenes de entradas en el encoder para después localizarlos espacialmente y contextualizarlos en el decoder.

En su presentación obtuvo muy buenos resultados en el *ISBI Cell Tracking Challenge 2015* y desde entonces ha sido ampliamente utilizada en tareas de segmentación semántica, sobre

todo en el ámbito de análisis de imágenes biomédicas.

Además, es una arquitectura muy flexible que, realizando pequeños ajustes, puede ser utilizada tanto para problemas de segmentación como de regresión o detección de objetos, por lo que resulta idónea para ser utilizada en un amplio abanico de problemas y suele ser el estándar de facto y utilizarse como línea base en proyectos de este tipo. Existen numerosos estudios que utilizan esta arquitectura en tareas similares a la propuesta o dentro del mismo dominio de aplicación, es decir, el análisis de imágenes del fondo ocular, [18, 19, 20, 21], por lo que será nuestra elección a la hora de llevar a cabo el diseño y desarrollo de nuestro proyecto, planteando posteriormente las modificaciones pertinentes para adaptarla a cada etapa.

4.2.1 Características

La arquitectura, en forma de U, se divide en dos caminos diferenciados: el *contracting path* o camino de contracción y el *expansive path* o camino de expansión, cada uno con unas características diferenciadas. Como característica general, podemos decir que es una **red completamente convolucional** (*fully convolutional network*), pues es una red que únicamente realiza operaciones de convolución, alternadas con operaciones de submuestreo (*downsampling*) y sobremuestreo (*upsampling*). Además, es una red neuronal ***end-to-end***, es decir, toma como entrada los datos crudos de imagen y produce como resultado la salida deseada para la tarea a realizar, sin necesidad de pre-procesados o post-procesados.

Podemos apreciar su estructura con más claridad en la figura 4.2 (página 33).

Por otro lado, cada uno de los bloques, ya sean del camino de contracción o del camino de expansión, utiliza un número prefijado de canales de características, coincidiendo con el doble del bloque anterior. El número de canales de características que obtenemos de la imagen de entrada en el primer bloque es conocido como número de **canales base o *base channels*** y será relevante a la hora de encontrar de manera más eficiente una buena representación de la salida deseada.

Camino de contracción

En cada capa del **contracting path** se aplican a la imagen de entrada filtros de convolución de 3×3 sin relleno (*unpadded*), seguidos por unidades lineales rectificadas (sección 4.2.2 , página 32) y, por último, *max-pooling* 2×2. Cada una de estas capas submuestrea su entrada a la mitad del tamaño y duplica el número de canales de características (*feature channels*) debido al número de filtros convolucionales que se aplican.

Camino de expansión

En cada capa del **expansive path** se sobremuestrea el mapa de características que proviene de la capa anterior, obteniendo un mapa de características del doble de tamaño, seguido de una convolución traspuesta 2×2 (*upconvolution*) que reduce el número de canales de características a la mitad. Después se concatena el nuevo mapa de características con el correspondiente al mismo nivel de la arquitectura en el contracting path. Por último, se aplican dos convoluciones convencionales 3×3 seguidas cada una por la aplicación de una ReLU. La concatenación se realiza con el objetivo de subsanar la pérdida de píxeles al realizar convoluciones sin relleno (*unpadded*).

Por último, se aplica una capa con una convolución 1x1 para mapear los vectores de características al número total de clases que queremos segmentar.

4.2.2 Función rectificadora

Como hemos explicado con anterioridad, es necesario que las redes convolucionales consten de funciones de activación que permitan a la red aprender patrones no separables linealmente. Estas funciones son generalmente aplicadas después de los bloques convolucionales.

La **función rectificadora** consiste en una función de activación que toma el valor de la entrada en caso de que este sea positivo y cero en el contrario.

Matemáticamente, viene expresada por la fórmula 4.1.

$$ReLU(x) = max(0, x) \tag{4.1}$$

En la figura 4.3 (página 33) podemos ver de forma más gráfica su aplicación.

Los bloques o unidades computacionales que utilizan esta función de activación se conocen como **unidades lineales rectificadas** o **ReLUs**, por sus siglas en inglés.

Es la función de activación que se utiliza por defecto en las redes de neuronas convolucionales, pues soluciona los problemas que dan otro tipo de funciones de activación como son la función de activación sigmoide o la función de activación de Tanh. Estos problemas están relacionados con la desaparición del gradiente debido a la saturación de estas funciones, pues a partir de ciertos umbrales dejan de ser sensibles a los aumentos o disminuciones en la entrada.

En la arquitectura U-Net, las ReLUs son las unidades que se utilizan después de los bloques

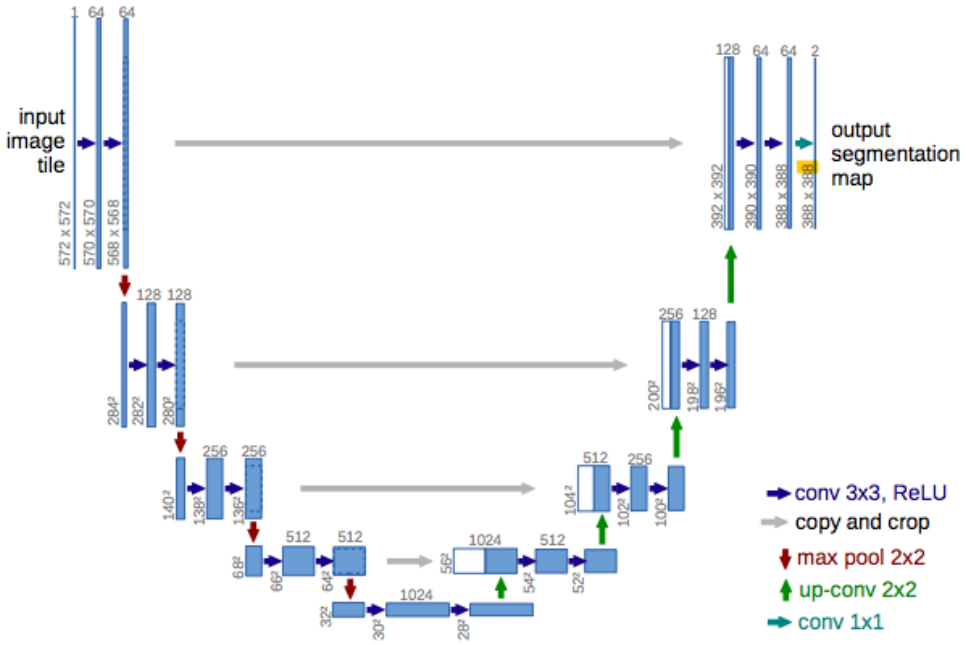

Figura 4.2: Arquitectura U-Net

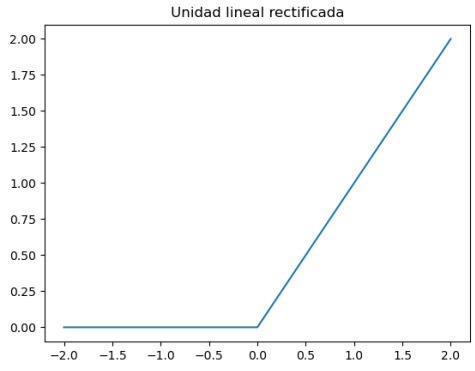

Figura 4.3: Gráfica de la función de activación rectificadora

convolucionales tanto del camino de contracción como del camino de expansión, por lo que será la que se utilice por defecto en el desarrollo del proyecto.

4.3 Imágenes de entrada y salida

Las redes de neuronas artificiales que utilizaremos para la resolución de las tareas planteadas toman retinografías como entrada, tanto para el entrenamiento como para su explotación a posteriori. Sin embargo, las características de los datos de *ground-truth* para el entrenamiento difieren entre tareas, pues en el caso de las tareas de predicción de profundidad necesitaremos o bien el mapa de profundidad o bien el par estereoscópico y, por otro lado, en la tarea de segmentación necesitaremos imágenes anotadas con dicha segmentación de disco y copa.

Para la realización del proyecto, debido a esta necesidad de contar con imágenes de diferente índole, se utilizarán dos conjuntos de datos diferentes:

- INSPIRE-stereo, [22].
- REFUGE, [23].

4.4 Optimizadores

Como hemos explicado en la sección **??** (página **??**), las redes neuronales utilizan algoritmos basados en el algoritmo de descenso del gradiente para modificar los pesos de la red y llegar a la solución óptima. Este tipo de algoritmos se conocen como **optimizadores** y existen múltiples variantes que es preciso conocer para poder elegir aquella que mejor se adecúe a las necesidades de la tarea. En este apartado explicaremos las opciones elegidas a la hora de desarrollar nuestro proyecto.

4.4.1 Descenso del gradiente estocástico

El **descenso del gradiente estocástico** es el algoritmo de descenso del gradiente por defecto, en el cual únicamente tomamos como dirección de actualización de parámetros la contraria a la dada por el gradiente, es decir, vamos en contra de la pendiente.
En la fórmula 4.2 podemos ver dicha actualización, donde w_i es el peso en esta iteración, w_{i-1} es el peso en la iteración anterior, α es la tasa de aprendizaje y δx es el gradiente para el vector de parámetros, x.

$$w_i = w_{i-1} - \alpha * \delta x \tag{4.2}$$

Se conoce como **estocástico** porque realiza la operación de actualización de los parámetros después de calcular la salida para una entrada aleatoria, de manera que la red no se estanque al aprender siempre a través de los mismos ejemplos.

Esta variante por defecto no suele utilizarse, pues tiene muchas limitaciones, por lo que es conveniente encontrar modificaciones en la misma que ayuden a optimizar la función de coste de manera más eficiente.

4.4.2 Adam

Se conoce como **Adam** a una variación del descenso del gradiente que utiliza dos conceptos clave: el **momento** y la **tasa de aprendizaje adaptada a cada parámetro**.

Sin entrar en detalles matématicos, pues son complejos y se escapan del alcance del proyecto, el **momento** lo que trata de conseguir es aprovecharse del concepto físico de la energía potencial de manera que las actualizaciones de los parámetros se realicen en una dirección que tenga en cuenta tanto el gradiente como las direcciones seguidas hasta ese momento en una ventana de intervalos de tiempo previos, pues es más probable que siguiendo el mismo camino se llegue antes a un punto mínimo.

En la fórmula 4.3 podemos ver una actualización de pesos con momento. La variable μ se corresponde con un hiperparámetro adicional. La variable v se inicializa a 0 y es análoga al concepto de velocidad.

$$\begin{aligned} v_i &= v_{i-1} * \mu - \alpha * \delta x \\ w_i &= w_{i-1} + v_i \end{aligned} \tag{4.3}$$

Por otro lado, a diferencia de en el descenso del gradiente estocástico, en este algoritmo optimizador no utilizamos la misma tasa de aprendizaje para todos los parámetros, si no que utilizamos una diferente para cada uno, de manera que el proceso es más individualizado y es más fácil llegar a los mínimos globales.

Matemáticamente, utiliza la fórmula representada en 4.4 para la actualización de parámetros.

$$
\begin{aligned}
m &= \beta_1 * m + (1 - \beta_1) * \delta x \\
v &= \beta_2 * v + (1 - \beta_2) * (\delta x)^2 \\
w_i &= -\frac{\alpha * m}{\sqrt{v} + \epsilon}
\end{aligned}
\tag{4.4}
$$

Es el algoritmo que se utiliza actualmente por defecto pues demostró tener mejores resultados en la gran mayoría de los casos, [24]. Por este motivo, será el algoritmo optimizador que utilizaremos a la hora de implementar la arquitectura.

Como valores de β_1 y β_2 utilizaremos 0.9 y 0.999 en todas las redes utilizadas en el proyecto.

4.4.3 Operaciones de data augmentation

Como se ha explicado en el apartado 2.3.4 (página 16) es posible utilizar operaciones de *data augmentation* para aumentar artificialmente el tamaño del conjunto de datos a utilizar y además hacer más robusta la red que estamos entrenando a diferentes cambios tanto morfológicos como de color de las imágenes de entrada, de forma que las predicciones futuras sean mejores.

Utilizaremos las mismas operaciones de data augmentation en todas las redes que se entrenarán a lo largo de la experimentación, siendo estas las siguientes:

- **Variación aleatoria del color en escala HSV (*random hsv*)**: Se modifica de manera aleatoria el color de la imagen de entrada en escala HSV con valores bajos de forma que éste siga siendo coherente con las posibles imágenes de entrada, pero produciendo modificaciones que podrán simular modificaciones en la luminosidad o contraste de las retinografías.
- **Transformación afín aleatoria de la imagen (*random affine*)**: Consiste en aplicar transformaciones a la imagen (rotación, zoom, etc) de forma aleatoria que preservan ciertas características geométricas de la misma, como que los puntos que estaban sobre una línea lo siguen estando o que las razones de distancia entre diferentes puntos sigue siendo igual.
- **Volteo horizontal aleatorio (*random horizontal flip*)**: La imagen es volteada horizontalmente de forma aleatoria, es decir, a veces sí y a veces no.
- **Volteo vertical aleatorio (*random vertical flip*)**: Igual que el horizontal pero en el eje vertical.

Estas operaciones son aplicadas tanto a la imagen de entrada como a los *ground-truths*, pues la salida deseada también queremos que contenga la información de estas modificaciones para que la red no aprenda una relación errónea entre entrada y salida.

4.5 Subsistema de predicción del mapa de profundidad

La primera aproximación que se abordará en el proyecto para la predicción de información de profundidad a través de una retinografía monocular se hará a través de la predicción del mapa de profundidad.

En la figura 4.4 (página 37) podemos ver un esquema general del procesamiento que queremos que realice la red convolucional que utilizaremos.

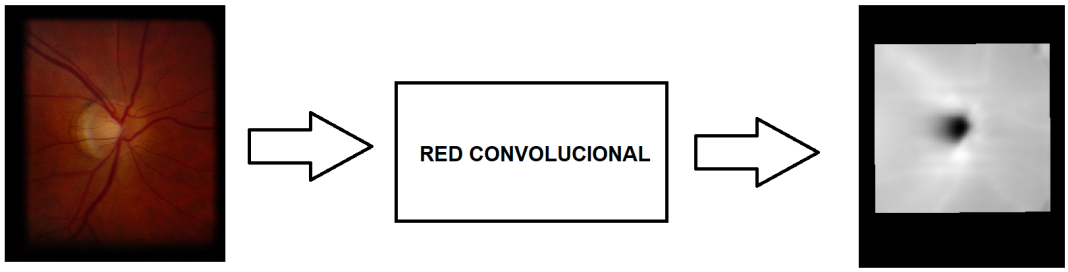

Figura 4.4: Esquema de aprendizaje del mapa de profundidad

Para cumplir con este cometido, la red utilizará la arquitectura U-Net (sección 4.2, página 30) y será entrenada utilizando una metodología de aprendizaje auto-supervisado. Para ello, utilizaremos las imágenes del conjunto de datos *INSPIRE-stereo* (sección 4.5.1, página 37) en el que tenemos tanto las retinografías de entrada como los mapas de profundidad deseados.

4.5.1 Imágenes de entrada: INSPIRE-stereo

El conjunto de datos *INSPIRE-stereo* consta de 30 retinografías estereoscópicas a color acompañadas de un *ground-truth* del mapa de profundidad para cada par de imágenes. De esta manera, con este conjunto de datos podremos abordar tanto la tarea de predicción del propio mapa como la tarea de predicción de la imagen desfasada dado su par.

Cabe destacar que el mapa de profundidad utilizado como *ground-truth* se obtiene a partir de imágenes OCT alineadas con las retinografías, de forma que de estas imágenes OCT es de donde deriva realmente la información detallada de profundidad.

En la figura 4.5 (página 41) podemos ver un ejemplo de cada una de las imágenes de las que dispone el dataset. Las dos imágenes de la izquierda son el par estereoscópico de la retinografía y la imagen de la derecha es el mapa de profundidad.

No se realizará ningún pre-procesado de las imágenes a priori. No obstante, a partir del *ground-truth* del mapa de profundidad, el cual solo comprende una región determinada de la extensión de la retinografía completa, se calculará una máscara que se tendrá en cuenta a la hora de entrenar la red, pues será ésta la única región de interés. El tamaño de las imágenes es de 768×1019. El formato de las retinografías es tif, constando cada archivo tif con dos páginas, una para cada par de la retinografía estereoscópica. Por motivos de simplificación a la hora de tratar las imágenes, se convirtieron estas imágenes tif a dos imágenes png cada una. Los *ground-truths* también se encuentran en formato png.

A la hora de entrenar la red convolucional para predicción del mapa de profundidad, utilizaremos 25 imágenes como conjunto de entrenamiento y 5 imágenes como conjunto de validación.

4.5.2 Función de coste: L2

La **función de coste L2**, también conocida como **MSE**, consiste en calcular la norma euclidea o distancia entre los vectores que se están comparando, en este caso la imagen de salida y la salida de la propia red. De esta manera, cuanto más alejados estén los vectores entre sí, mayor será el valor de la función de coste.

Se calcula con expresión de la fórmula 4.5.

$$L2\,(p,q) = \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2} \tag{4.5}$$

Es una función de coste bastante genérica que aplica a muchos casos y suele utilizarse por defecto siempre y cuando no conozcamos una función de coste mejor que para el problema específico que intentamos resolver. Cabe destacar que es bastante sensible a valores atípicos (*outliers*) y en caso de ser notables en el dataset es conveniente usar la función de coste L1. En el resto de casos L2 obtiene mejores resultados.

Será la función de coste que utilizaremos para entrenar la red neuronal que predice el mapa de profundidad a partir de una única imagen retinal. En el estudio realizado por *Sharath M. Shankaranarayana et al.*, [25], se demuestra que su utilización es positiva a la hora de predecir mapas de profundidad, lo que apoya la decisión tomada.

4.5.3 Detalles de entrenamiento específicos

Si bien la red que utilizaremos es una red convolucional con al arquitectura U-Net estándar, existen ciertos aspectos que modificaremos para adecuarla a la tarea específica que estamos resolviendo.

Básicamente, el proceso consiste en conseguir una imagen en escala de grises que prediga la profundidad en cada punto de la imagen a partir de una imagen a color. Para ello utilizaremos los siguientes valores:

- Canales de entrada: 3
- Canales de salida: 1
- Canales base de la red (neuronas): 32 y 64
- Learning rate: 10^{-3}

Para determinar el número de épocas de entrenamiento se utiliza un *scheduler* mediante el cual si el valor obtenido por la función de coste no mejora en 200 iteraciones se detiene el entrenamiento.

La red será entrenada con inicialización de pesos aleatoria, utilizará la función de coste L2 y no usará ninguna función de activación a mayores de las ReLU propias de la arquitectura U-Net.

4.5.4 Salida de la red

La salida de la red de predicción del mapa de profundidad será una imagen en escala de grises que defina el propio mapa, es decir, obtendremos una imagen con un único canal con valores entre 0 y 255 que definirán el valor de gris de cada uno de los píxeles.

Este resultado, obtenido directamente de la red, es comparable con el *ground-truth*, pues es una imagen de las mismas características.

4.5.5 Métodos de evaluación

Métodos cualitativos

La evaluación cualitativa en esta tarea consistirá en visualizar los resultados obtenidos por la red para poder determinar si estos son correctos o no. Será necesario visualizar también la imagen de entrada de la que proviene una salida determinada de forma que podamos observar si la red está realizando bien la relación entre las estructuras presentes en la retinografía y la profundidad del mapa de salida.

Será necesario comprobar si las zonas más profundas, es decir, con valores más cercanos a negro en la imagen de salida, se corresponden con regiones de la retinografía que realmente tienen más profundidad. Además, en las retinografías tomadas a pacientes con glaucoma, hemos de poder visualizar una diferencia de profundidad frente a las retinografías de pacientes sanos.

También se utilizará una representación tridimensional del mapa de profundidad obtenido, de forma que podemos ver hasta qué punto la red está aprendiendo detalles más complejos o únicamente está realizando una aproximación genérica para todos los casos. Esta representación resulta muy útil para este problema en concreto pues la profundidad es realmente una característica tridimensional, por lo que poder visualizar los datos con esta perspectiva es lo idóneo para comprobar su validez.

Métodos cuantitativos

La evaluación cuantitativa para esta tarea consistirá en el cálculo de la **métrica L2** o MSE, calculada de la misma forma que la función de coste L2 (sección 4.5.2, página 38) utilizada en el entrenamiento de la red.

4.6 Subsistema de predicción del par estereográfico

La segunda aproximación que se abordará en el proyecto para la predicción de información de profundidad a través de una retinografía monocular se hará a través de la predicción del par estereográfico.

En la figura 4.6 (página 41) podemos ver un esquema general del procesamiento que queremos que realice la red convolucional que utilizaremos.

Para cumplir con este cometido, al igual que en la tarea de predicción del mapa de profundidad, de la red utilizará la arquitectura U-Net (sección 4.2, página 30) y será entrenada utilizando una metodología de aprendizaje auto-supervisado. Para ello, utilizaremos las imágenes del conjunto de datos *INSPIRE-stereo* (sección 4.5.1, página 37) en el que tenemos tanto las retinografías de entrada como los mapas de profundidad deseados.

4.6.1 Imágenes de entrada

Las imágenes de entrada que utilizaremos en este subsistema serán las mismas que se utilizan en el subsistema de la tarea de predicción del mapa de profundidad (sección 4.5.1, página 37), ya que queremos obtener el mismo tipo de información a través de los mismos datos de entrada.

Sin embargo, como queremos que esta información se encuentre en el par estereoscópico en lugar de en el propio mapa de profundidad, utilizaremos este par como *ground-truth* para

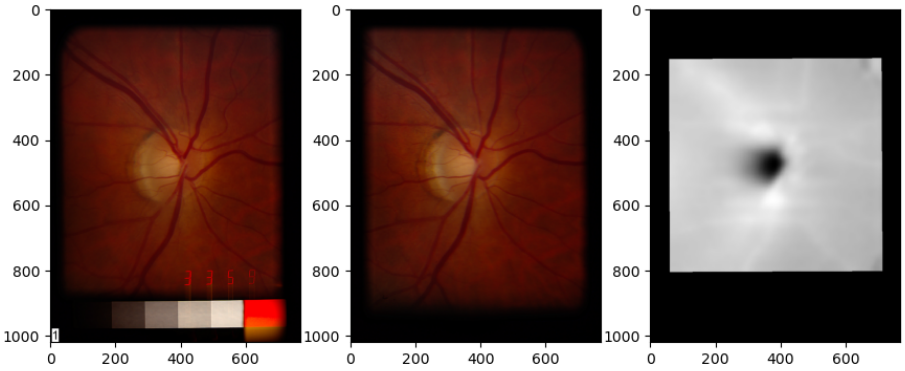

Figura 4.5: Imágenes del dataset INSPIRE-stereo

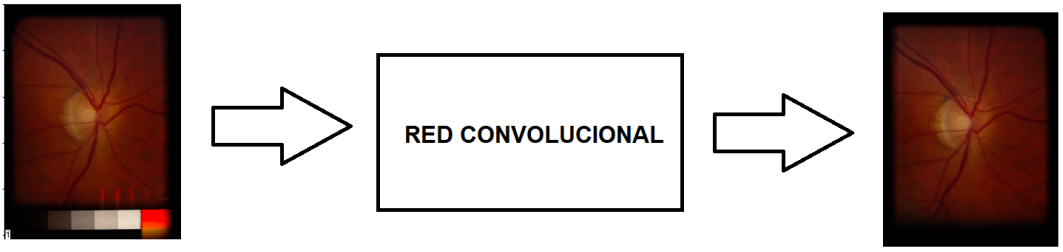

Figura 4.6: Esquema de aprendizaje del par estereográfico

el entrenamiento y como objetivo de salida de la red.

4.6.2 Función de coste: SSIM

La **similitud estructural** o **SSIM**, es una medida de la similitud entre dos imágenes. Se diferencia de otras técnicas, como la MSE, en que tiene en cuenta la relación estructural entre los elementos de la imagen, es decir, considera que la posición de los elementos determina sus relaciones de interdependencia y utiliza esta información para calcular su valor, considerando que los píxeles vecinos están más altamente relacionados que los lejanos. Para conseguir esta información estructural, se calcula utilizando diversas ventanas de un tamaño determinado dentro de la imagen.

Utiliza la fórmula detallada en 4.6 para calcular su valor.

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + C_1) + (2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \tag{4.6}$$

donde μ_x y μ_y son las medias locales de x y de y (vectores a comparar) respectivamente, σ_x y σ_y son las varianzas de x y de y y C_1 y C_2 son dos variables a las que nosotros mismos daremos valor para estabilizar la división.

También es necesario especificar el tamaño de la localidad de la imagen en la que calcularemos estos parámetros. Además, dependiendo de la técnica de integración espacial utilizada, puede ser necesario indicar el valor de otros parámetros, como por ejemplo σ en el caso de utilizar un filtro Gaussiano.

Como realmente el SSIM es un índice de similaridad, hemos de utilizar su valor negativo, pues nosotros buscamos que cuanto más similares sean dos imágenes, menor sea el valor de la función de coste. De esta manera, nuestra función de coste será realmente la que se indica en la fórmula 4.7.

$$SSIMLoss(x,y) = -SSIM(x,y) \tag{4.7}$$

Es adecuada para problemas de traslación, como es el caso de nuestro pre-entrenamiento en el que queremos predecir la imagen desfasada dado su par. Esto se debe a que es una métrica

de evaluación de calidad que no es punto a punto, es decir, no solo mide las diferencias de píxeles, sino también la estructura de la imagen, y ha arrojado buenos resultados en este tipo de casos, [17].

4.6.3 Detalles de entrenamiento específicos

En esta tarea es necesario adecuar los diferentes aspectos variables de la red al problema de predicción del par estereoscópico, que consideramos un problema de traslación. La red parte de una imagen a color y obtiene como resultado otra imagen a color de las mismas características pero desfasada.

Los valores que utilizaremos serán los siguientes:

- Canales de entrada: 3
- Canales de salida: 3
- Canales base de la red (neuronas): 32 y 64
- Learning rate: 10^{-4}

Para determinar el número de épocas de entrenamiento utilizaremos el mismo *scheduler* que en el subsistema de predicción del mapa de profundidad (sección 4.5.3, 39)

La red será entrenada con inicialización de pesos aleatoria, utilizará la función de coste SSIM y no usará ninguna función de activación a mayores de las ReLU propias de la arquitectura U-Net.

Como parámetros de la función de coste SSIM, utilizaremos un tamaño de ventana estándar de 11x11. Además, utilizaremos unos valores de $C1 = 0.01^2$ y $C2 = 0.03^2$, valores que se utilizan por defecto en el cálculo de la similaridad estructural.

4.6.4 Salida de la red

La salida de la red en esta tarea se corresponde con la predicción del par estereográfico de la entrada que se le proporcione, por lo que tendremos imágenes con tres canales de color, con valores entre 0 y 255 en cada canal.

La red aprenderá estas representaciones de forma directa por lo que los resultados obtenidos podrán compararse directamente con los *ground-truth* a la hora de comprobar su validez.

4.6.5 Métodos de evaluación

Métodos cualitativos

La evaluación cualitativa para esta tarea consistirá en visualizar los resultados obtenidos por la red de manera que al observarlos juntos a las imágenes de entrada podamos ver si estos

resultados son correctos o no. Simplemente hemos de ver que la imagen de salida obtenida mantiene las mismas características y elementos estructurales que la imagen de entrada, dispuestos de la misma manera, pero desplazados en el eje horizontal.

Métodos cuantitativos

Al igual que en la otra tarea de predicción de información de profundidad, utilizaremos la métrica L2 (sección 4.5.2, página 38) para comparar los resultados obtenidos con los deseados.

4.7 Subsistema de segmentación de disco y copa

La tarea finalista será llevada a cabo por un subsistema que, a partir de una retinografía monocular, clasifique cada uno de los tres píxeles de entrada en una de las tres clases posibles: fondo, disco o copa.

En la figura 4.7 (página 44) podemos ver un esquema general del procesamiento que queremos que realice la red convolucional que utilizaremos.

Figura 4.7: Esquema de aprendizaje de segmentación disco-copa

Para la realización de esta tarea, la red utilizará también la arquitectura U-Net (sección 4.2, página 30) y será entrenada utilizando una metodología de aprendizaje puramente supervisado, utilizando un esquema de clasificación para cada píxel. Para ello, utilizaremos las imágenes del conjunto de datos *REFUGE* (sección 4.7.1, página 44) en el que tenemos tanto las retinografías de entrada como las imagenes etiquetadas con la segmentación de disco y copa.

4.7.1 Imágenes de entrada: REFUGE

El conjunto de datos *REFUGE* consta de 1200 imágenes dividas en 3 grupos: entrenamiento, validación y test, en una proporción 1:1:1, es decir, 400 imágenes en cada grupo. Para el entrenamiento de nuestra red utilizaremos solo 800 imágenes, las pertenecientes a los grupos

de entrenamiento y validación. El conjunto de datos está formado por retinografías monoculares a color de las que se dispone de un *ground-truth* con las etiquetas de las regiones de disco y copa. Además, en el caso de las imágenes del cojunto de entrenamiento (no es así para los conjuntos de validación y de test) se dispone de un indicador de si el paciente al que se le sacó la retinografía sufre la enfermedad del glaucoma o no. Con estas imágenes entrenaremos la red en la tarea de segmentación de disco y copa.

A la hora de evaluar el correcto funcionamiento de la red utilizaremso las 400 imágenes restantes del conjunto de test.

Para que las características de las imágenes sean coherentes con las del primer conjunto de datos, lo cual es necesario pues queremos entrenar la red en la tarea de segmentación con los pesos aprendidos en la primera tarea de predicción de información de profundidad, éstas han sido sometidas a un pre-procesamiento consistente simplemente en recortar las imágenes tomando como referencia el centroide del *ground-truth*, que es de hecho el centroide del disco óptico. Así, utilizando como medida estándar la de 4 medidas de disco óptico de alto y 3 de ancho (cifras obtenidas a mano de manera aproximada) obtuvimos imágenes centradas en el disco óptico con una proporción aproximada a las del otro dataset. Por último, se reescalaron manteniendo el ratio original para igualar el ancho de las imágenes del primer dataset, quedando finalmente de un tamaño de 768×1195. Las imágenes están en formato png, mientras que los grount-truth están en formato bmp.

En la figura 4.8 (página 46) se pueden observar las imágenes que utiliza el dataset (después del procesamiento). La imagen de la izquierda se corresponde con la retinografía y la imagen de la derecha es el *ground-truth* de la segmentación de disco y copa, siendo el disco la región en gris y la copa la región en negro.

4.7.2 Función de coste: Cross-Entropy

De forma general, la **entropía cruzada** se utiliza para calcular la diferencia entre dos distribuciones de probabilidad y, al ser utilizada como función de coste, su valor aumentará cuanto más difieran dichas distribuciones.

La fórmula para calcular la función de coste Cross-Entropy es la presentada en la ecuación 4.8.

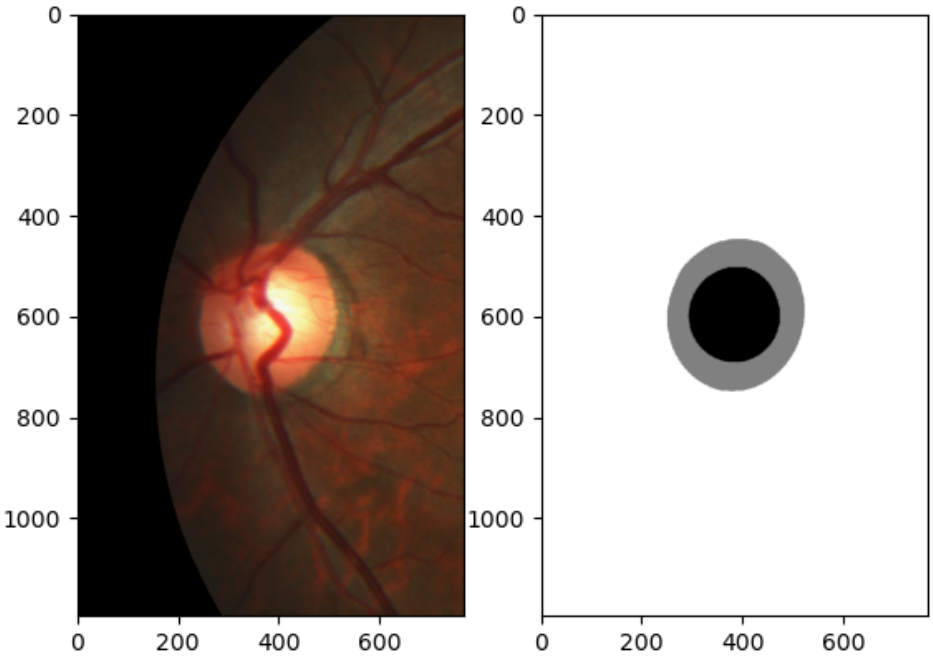

Figura 4.8: Imágenes del dataset REFUGE

$$CrossEntropyLoss(y, p) = -\sum_{i=1}^{C} y^{(i)} \log p^{(i)} \tag{4.8}$$

siendo C el total de clases, y el valor deseado para esa clase (0 o 1) y p el valor de predicción dado por la red para esa clase.

Resulta adecuada para problemas de clasificación, pues podemos decir que una imagen pertenece a una clase, C, con probabilidad 1 en el patrón anotado y después aplicar la función de coste *Cross-Entropy* para compararlo con la probabilidad resultante que el modelo clasificador da para dicha clase.

En nuestro caso, utilizaremos esta función de coste para el problema de segmentación de copa y disco, pues lo consideraremos a grandes rasgos como un problema de clasificación en el que cada píxel pertenece a la región del fondo, de la copa o del disco y el objetivo de la red será clasificar cada píxel individualmente, dándoles un valor de probabilidad para cada región. De esta manera, podemos aplicar convenientemente la función de coste *Cross-Entropy* comparando las etiquetas de los *ground-truths* con las probabilidades arrojadas por la red. Aplicaremos la fórmula 4.8 a cada píxel de la imagen para obtener posteriormente un valor medio, que será la pérdida o *loss* total.

4.7.3 Función de activación: Softmax

La **función softmax** o **función exponencial normalizada** es una función utilizada para comprimir un vector k-dimensional con valores reales en un rango arbitrario a un vector k-dimensional pero con valores en el rango [0-1]. Podemos ver su expresión matemática en la fórmula 4.9.

$$softmax(x)_i = \frac{exp(x_i)}{\sum_j exp(x_j)} \quad i = 1, \dots, K \tag{4.9}$$

La salida de la función softmax representa una distribución de probabilidad sobre K posibles salidas. Así, al aplicarlo a la salida de una red neuronal de clasificación lo que obtenemos es la posibilidad de que la entrada pertenezca a una clase u otra. Realmente no son probabilidades en el sentido estricto de la palabra, pero pueden interpretarse como tal y utilizarlas como indicador de qué clase es más probable frente a las demás, si bien el valor no puede tomarse de forma absoluta.

En nuestro proyecto y en concreto en la tarea de segmentación, utilizaremos esta función de activación en la salida de la última capa de la arquitectura, de manera que mapeemos los valores de la salida a probabilidades, obteniendo los mapas de probabilidad de la clasificación para cada píxel (sección 4.7.5, página 4.7.5). La utilización de esta función de activación es de uso estándar para redes neuronales de clasificación. Como se ha planteado la tarea de segmentación como un problema de clasificación de cada píxel de la imagen individualmente en una de las tres clases posibles (fondo-disco-copa), resulta una función de activación adecuada.

4.7.4 Detalles de entrenamiento específicos

Esta última tarea que realiza la red de segmentación es de carácter diferente a las dos redes planteadas para el otro subsistema, en el sentido de que el problema a abordar ahora es un problema de clasificación que utilizará aprendizaje puramente supervisado.

Los valores que utilizaremos serán los siguientes:

- Canales de entrada: 3
- Canales de salida: 3
- Canales base de la red (neuronas): 32 y 64
- Learning rate: 10^{-4}

Para determinar el número de épocas de entrenamiento utilizaremos el mismo *scheduler* que en el subsistema de predicción del mapa de profundidad (sección 4.5.3, 39)

La red será entrenada con dos inicializaciones de pesos diferentes:

- Inicialización aleatoria.
- Inicialización con los pesos obtenidos en el subsistema de predicción de información de profundidad (*transfer-learning*).

Además, utilizará la función de coste Cross-Entropy y aplicará una función de activación Softmax a la salida para producir los mapas de probabilidad de la clasificación (sección 4.7.5, página 4.7.5).

4.7.5 Salida de la red

La salida de la red para la tarea de segmentación de disco y copa estará formada por tres mapas de probabilidad, uno para cada región de la imagen: fondo, disco y copa. Consideraremos fondo a todo píxel de la imagen de entrada que no esté dentro del disco.

Estos mapas tal como los aprende y genera la red, no son probabilidades como tal, sino una especie de valores de puntuación, que serán más altos cuanta más probabilidad haya de que ese píxel pertenezca a la región representada en ese mapa. Por ello, es preciso aplicar la función de activación Softmax después de obtener la salida, pasando los valores al rango 0-1 y pudiéndolos usar como distribuciones de probabilidad.

Como estos mapas de probabilidad no son fácilmente interpretables de manera visual, lo que haremos posteriormente será procesarlos para obtener imágenes con las mismas características que el *ground-truth*, es decir, imágenes en las que los píxeles de la región del fondo tienen color blanco (valor 255), los píxeles de la región del disco tienen color gris (valor 128) y los píxeles de la copa tienen color negro (valor 0). Para obtener estas imágenes, lo que hacemos es comparar las probabilidades de cada píxel en los tres mapas y darle un valor u otro a ese píxel en la imagen de salida dependiendo de cual de los tres mapas tiene una probabilidad mayor.

4.7.6 Métodos de evaluación

Métodos cualitativos

De nuevo, los métodos cualitativos de evaluación de la red de segmentación pasarán por la visualización de los resultados obtenidos por la misma.

Es posible ver en las retinografías de entrada de manera aproximada el disco y la copa si en estas hay un contraste suficiente, por lo que al observar el resultado obtenido en forma de imagen etiquetada, podemos comprobar si el resultado es bueno o no si estas etiquetas respetan la forma y el tamaño del disco y la copa de la retinografía original. Además, en imágenes de pacientes afectados por la enfermedad de glaucoma tenemos que ser capaces de observar como el ratio vertical disco-copa es menor que en imágenes de pacientes sanos. Esto quiere decir que veremos como la copa ocupa mayor área del disco en unas imágenes que en otras, fenómeno observable a simple vista.

Métodos cuantitativos

A la hora de evaluar cuantitativamente los resultados de la red de segmentación, utilizaremos dos métodos: la métrica ***intersection over union*** o **IOU** y la representación mediante **curvas ROC**. Para aplicar estos métodos de evaluación será necesario transformar el problema de clasificación multi-clase en problemas de clasificacion binaria. En el caso del cálculo de

intersección over union, utilizaremos los problemas de clasificación de copa contra el resto y disco contra el resto. Sin embargo, en el caso de las curvas ROC, debido a que la red de segmentación realmente un clasificador multi-clase, no obtenemos probabilidades para la región del disco como tal, sino para el anillo neurorretiniano, por lo que consideraremos los problemas de clasificación binaria de copa contra el resto y anillo contra el resto.

La métrica ***intersection over union*** consiste en calcular la intersección de la salida de la red con el *ground-truth* correspondiente a la imagen de entrada procesada. Con estos valores se calcula la relación de la interesección sobre la unión, valor que es representativo de lo bien que se aproxima la salida de la red al *ground-truth*, pues indica qué proporción del area segmentada está realmente en la segmentación deseada.

Su expresión matemática queda reflejada en la fórmula 4.10, donde x es la imagen de salida de la red e y es el *ground-truth* correspondiente a su imagen de entrada.

$$IOU(x, y) = \frac{\cap(x, y)}{\cup(x, y)} \tag{4.10}$$

Los valores de la métrica van entre 0 y 1, de forma que podemos decir que:

- Los valores entre 0 y 0.5 son una mala aproximación.
- Los valores entre 0.5 y 0.9 son una aproximación aceptable.
- Los valores mayores a 0.9 son una buena aproximación.

En la figura 4.9 (página 50) podemos ver un ejemplo de aplicación de esta métrica.

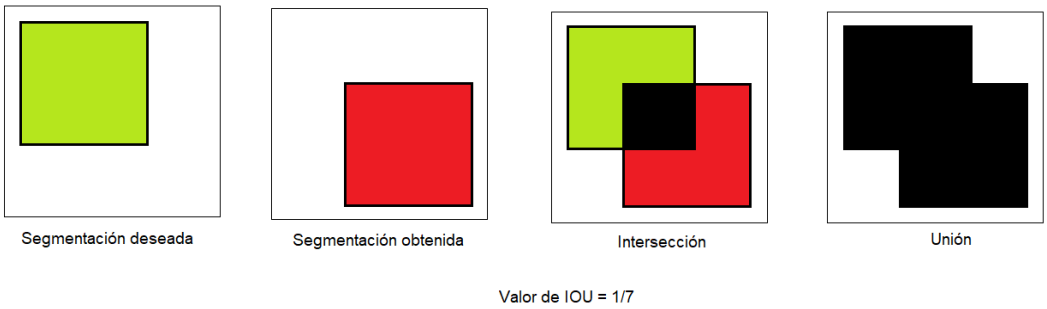

Figura 4.9: Ejemplo intersection over union

La representación mediante **curvas ROC** consiste en representar el ratio de falsos positivos (FPR), es decir, el ratio de píxeles clasificados de la clase objetivo pero que no lo son, frente al ratio de verdaderos positivos (TPR), es decir, el ratio de píxeles clasificados como de la clase objetivo que realmente sí son de esa clase.

El área total máxima bajo una curva ROC es de 1 y este valor se corresponde con la segmentación objetivo, es decir, si obtenemos un área bajo la curva (AUC) de 1, significa que la segmentación ha sido la deseada. El valor del AUC va decreciendo cuanto peor sea la segmentación.

En la figura 4.10 (página 51) podemos ver un ejemplo de curvas ROC.

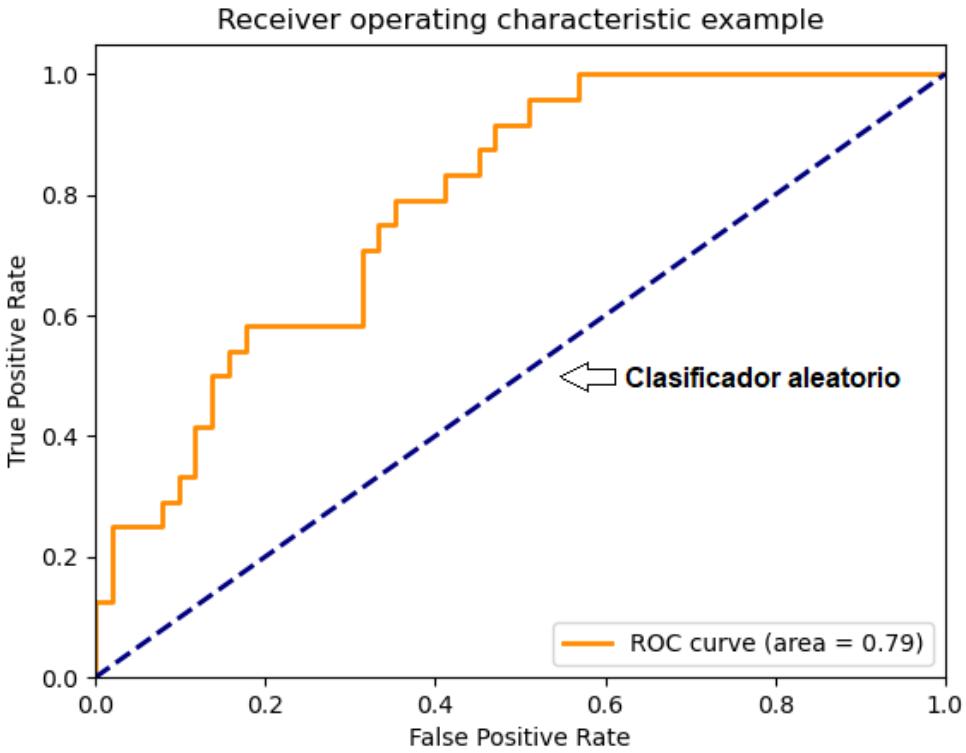

Figura 4.10: Ejemplo de curva ROC

Capítulo 5

Predicción de información de profundidad a partir de retinografía monocular

En este capítulo hablaremos sobre la primera tarea a realizar en este proyecto, la predicción de información de profundidad a partir de una retinografía monocular. En concreto, para cada una de las aproximaciones hablaremos sobre la experimentación realizada, presentaremos los resultados obtenidos y los discutiremos brevemente en vista de la conclusión final.

5.1 Predicción de mapa de profundidad

5.1.1 Experimentación

El experimento consiste en la ejecución de la red convolucional que, tomando como entrada una retinografía monocular, predice el mapa de profundidad correspondiente a dicha retinografía (4.5, página 37).

Esta ejecución se llevará a cabo utilizando dos arquitecturas:

- Arquitectura U-Net con 32 canales base.
- Arquitectura U-Net con 64 canales base.

Además, al ser una red que no consume mucho tiempo a la hora de ejecutarse, se probarán diferentes learning rates: 10^{-2}, 10^{-3} y 10^{-4}.

De estas ejecuciones se obtendrán las curvas de aprendizaje, que representan la mejora del loss frente al tiempo (iteraciones), así como las imágenes de salida para el conjunto de

		LR	
		10^{-2}	10^{-3}
BC	32	0.0142	0.0115
	64	-	0.0213

Tabla 5.1: Métrica L2 para los resultados de las redes convolucionales de predicción del mapa de profundidad

validación cada 625 iteraciones de entrenamiento. Las curvas de aprendizaje están calculadas en escala logarítmica.

Con los resultados del conjunto de validación se calcularán las métricas cuantitativas pertinentes, en este caso la métrica L2. Esta métrica será cálculada únicamente en la región de interés delimitada por la máscara.

Finalmente, se mostrarán los resultados obtenidos para el mejor modelo (conjunto de pesos) que se obtuvo durante el entrenamiento y se discutirán los valores de las métricas obtenidas para cada experimento.

5.1.2 Resultados parciales

En la figura 5.1 (página 55) podemos visualizar la salida de la mejor red obtenida en el etrenamiento para el conjunto de validación, imágenes centrales, representadas frente la imagen de entrada, imágenes de la izquierda, y el *ground-truth*, imágenes de la derecha. Por orden, son las salidas pertenecientes a la red de 32 canales con *learning rates* 10^{-3} y 10^{-2}.

La figura 5.2 (página 60) representa una imagen de las mismas características que la anterior pero para la red de 64 canales y un *learning rate* de 10^{-3}.

En las figuras 5.3 (página 61) y 5.4 (página 62) podemos ver una representación tridimensional de estos mismos resultados.

En la figura 5.5 (página 63) podemos ver las curvas de aprendizaje obtenidas en las diferentes ejecuciones de la red.

Por último, en la tabla 5.1 (página 54) quedan registrados los resultados de la métrica L2 obtenidos para cada red.

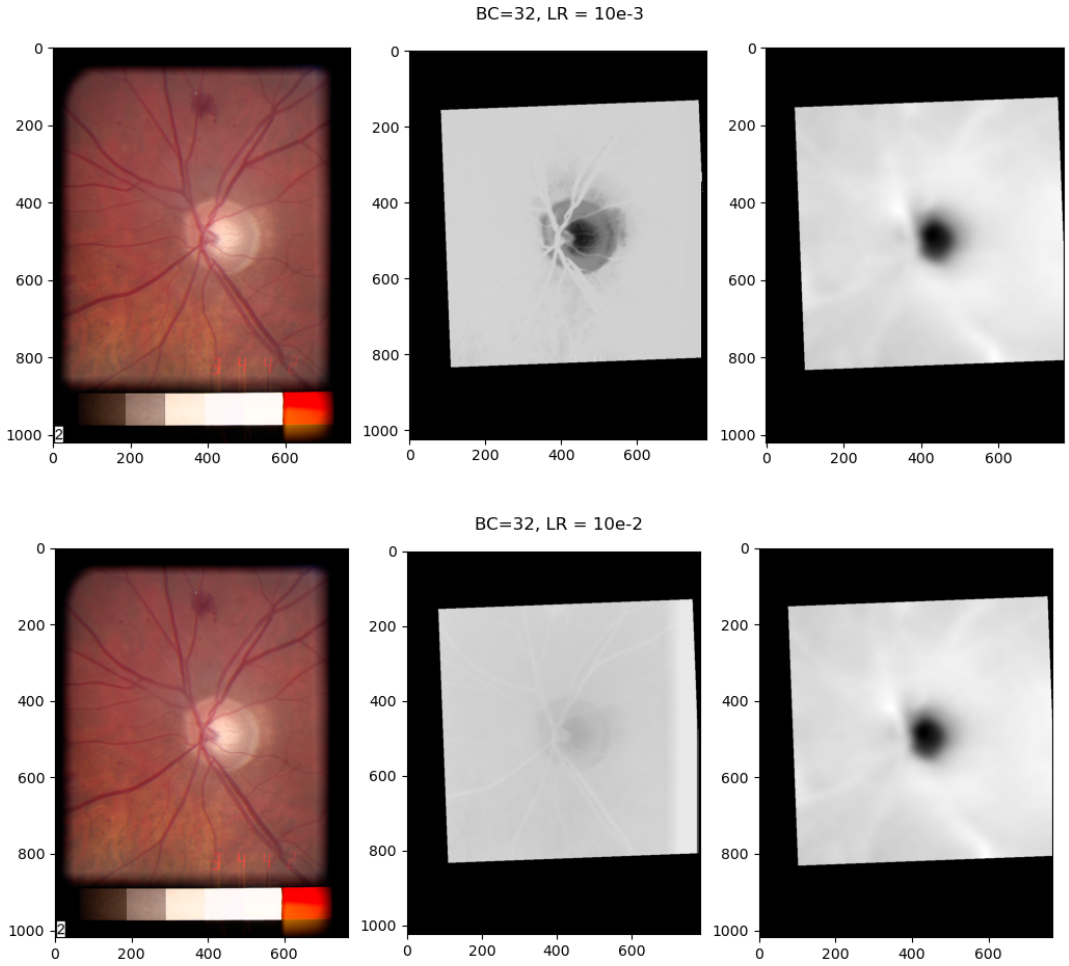

Figura 5.1: Resultados en el problema de predicción del mapa de profundidad de la red convolucional de 32 canales

5.1.3 Discusión de los resultados

Podemos ver que los resultados en esta tarea no son del todo satisfactorios. De forma cualitativa, se puede observar que las características de las imágenes obtenidas y de los ground-truths son sustancialmente diferentes. Principalmente, podemos ver como realmente lo que parece estar haciendo la red es mapear las zonas con cambios de color y contraste a zonas de diferente profundidad. Si bien esto puede ser un primer acercamiento a la resolución del problema, da como resultado soluciones insuficientes, pues se aprecia que la red no está aprendiendo a representar realmente información subyacente de profundidad

La red ha de aprender una representación de carácter muy diferente a la entrada, pues la salida deseada es obtenida a partir de una imagen OCT que tiene una naturaleza muy diferente a la de la imagen de entrada. Por ello, la red tiende a adoptar una solución intermedia entre mantener las características de la imagen de entrada y dar un resultado genérico a todas las imágenes.

Aún así, podemos ver como la red que mejores resultados obtiene, tanto a nivel cualitativo como a nivel de métrica L2 es la red de 32 canales base con *learning rate* de 10^{-3}. Por un lado, en la representación cualitativa frente al *ground-truth* (figura 5.1, página 55) podemos observar que, aunque siguen presentes elementos estructurales que no deberían aparecer, como son los vasos sanguíneos o el propio disco, la intensidad de gris es mayor en las zonas internas del disco, manteniendo características de intensidad de gris parecidas a las de la imagen deseada. Esto queda más latente en la representación tridimensional (figura 5.3, página 61), donde podemos ver como realmente se está dando como resultado un pico de profundidad en el medio del disco, similar al del ground-truth, aunque también observamos que la región que rodea este pico es más homogénea de lo que debería, fruto del compromiso antes mencionado entre preservación de la imagen original y generalización del resultado.

Por su parte, la red de 32 canales con *learning rate* 10^{-2} no parece estar aprendiendo ningún detalle relevante de profundidad, como podemos ver en la representación tridimensional. Además, los niveles de intensidad de gris difieren bastante de los de la salida deseada.

Por último, la red de 64 canales con *learning rate* 10^{-3}, si bien da un resultado similar a la de 32 canales base con el mismo *learning rate*, se queda un poco más del lado de preservación de características de la imagen original. El entrenamiento de esta red es más corto, como podemos ver en la figura 5.5 (página 63), lo que justifica este resultado. Así pues, los resultados obtenidos predicen algo peor la información de profundidad.

Los valores de la métrica L2 calculada para todas las redes planteadas (tabla 5.1, página 54) también permiten observar un mejor resultado para la red de 32 canales base y *learning rate* 10^{-3}, obteniendo peor valor para la red de 64 canales base y el mismo *learning rate*. Sin

embargo, se consideran peores los resultados de la red de 32 canales y *learning rate* 10^{-2} a nivel cualitativo, pues la generalización elimina casi todo detalle de la imagen y no predice información de profundidad.

5.2 Predicción del par estereoscópico

5.2.1 Experimentación

Este segundo experimento destinado a la obtención de información de profundidad de una retinografía monocular consiste en la ejecución de la red convolcional que predice el par estereográfico tomando como entrada dicha retinografía (figura 4.6, página 40).

La ejecución, como en el caso de la experimentación para la predicción del mapa de profundidad, será llevada a cabo en arquitectura U-Net tanto de 32 canales base como de 64 canales base. Utilizaremos siempre un *learning rate* de 10^{-4}.

De la ejecución de las redes se obtendrán las curvas de aprendizaje, así como las imágenes de salida para el conjunto de validación cada 625 iteraciones de entrenamiento.

Con los resultados obtenidos utilizando el mejor modelo se calculará la métrica L2 para medir cuantitativamente la corrección de los resultados. Además, se mostrarán frente a los *ground-truths* para analizarlos de manera cualitativa.

Para terminar, se discutirán los resultados en base tanto a las métricas obtenidas como a las caracerísticas observadas en la evaluación cualitativa.

5.2.2 Resultados parciales

La figura 5.6 (página 64) muestra el resultado obtenido por las redes convolucional de 32 y 64 canales base utilizando los mejores pesos obtenidos en entrenamiento en la imagen del centro, frente a la imagen de entrada en la imagen de la izquierda y el *ground-truth* en la imagen de la derecha.

En la figura 5.7 (página 65) podemos ver las curvas de aprendizaje obtenidas en las diferentes ejecuciones de la red.

BC	**32**	0.0034
	64	0.5178

Tabla 5.2: Métrica L2 para los resultados de las redes convolucionales de predicción del par estereográfico

BC	**32**	0.0019
	64	0.0127

Tabla 5.3: Métrica L2 para los resultados de las redes convolucionales de predicción del par estereográfico en escala de grises

En la tabla 5.2 (página 58) quedan registrados los resultados de la métrica L2 obtenidos para cada red.

Por último, la tabla 5.3 (página 58) muestra los valores de la misma métrica L2 pero para las imágenes en escala de grises.

5.2.3 Discusión de los resultados

Los resultados obtenidos se consideran satisfactorios, aunque las soluciones obtenidas sean algo simplificadas, pues al ser una tarea preliminar en la que la red únicamente tiene que trasladar los píxeles de la imagen de entrada en función de la profundidad a la que se encuentren, no es necesario que llegue a ningún compromiso como en el caso de la red de predicción de mapa de profundidad y podemos ver la similitud de características entre el *ground-truth* y la imagen obtenida (figura 5.6, página 64).

Sin embargo, en los resultados obtenidos para la red de 64 canales base podemos ver que las características de color no son las correctas. Se plantea que sea éste el motivo por el cual se obtiene una métrica L2 más alto (tabla 5.2, página 58) para esta red en concreto. Esta variación puede ser debida a las operaciones de data augmentation, entre las cuales se encuentra una modificación de las características de color de la imagen de entrada, que puede haber influido a la hora de aprender dichas características por parte de la red.

Los resultados obtenidos para la métrica L2 también nos dicen que la representación obtenida por la red de 32 canales base es sustancialmente mejor que la obtenida para la red de 64 canales base.

Además, al calcular la métrica L2 con las imágenes en escala de grises (tabla 5.3, página 58) podemos descartar la hipótesis de que los resultados únicamente son peores por el hecho de tener diferentes características de color, quedando demostrado que también hay mayor error a la hora de predecir la traslación espacial de la imagen de entrada. Podemos concluir

pues que los resultados obtenidos en la red de 32 canales son más satisfactorios que los de la red de 64 canales.

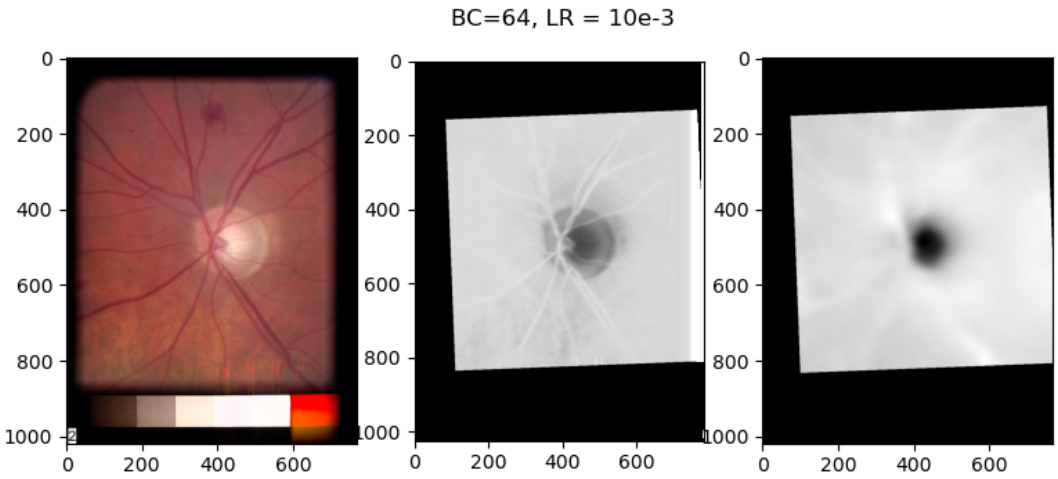

Figura 5.2: Resultados en el problema de predicción del mapa de profundidad de la red convolucional de 64 canales

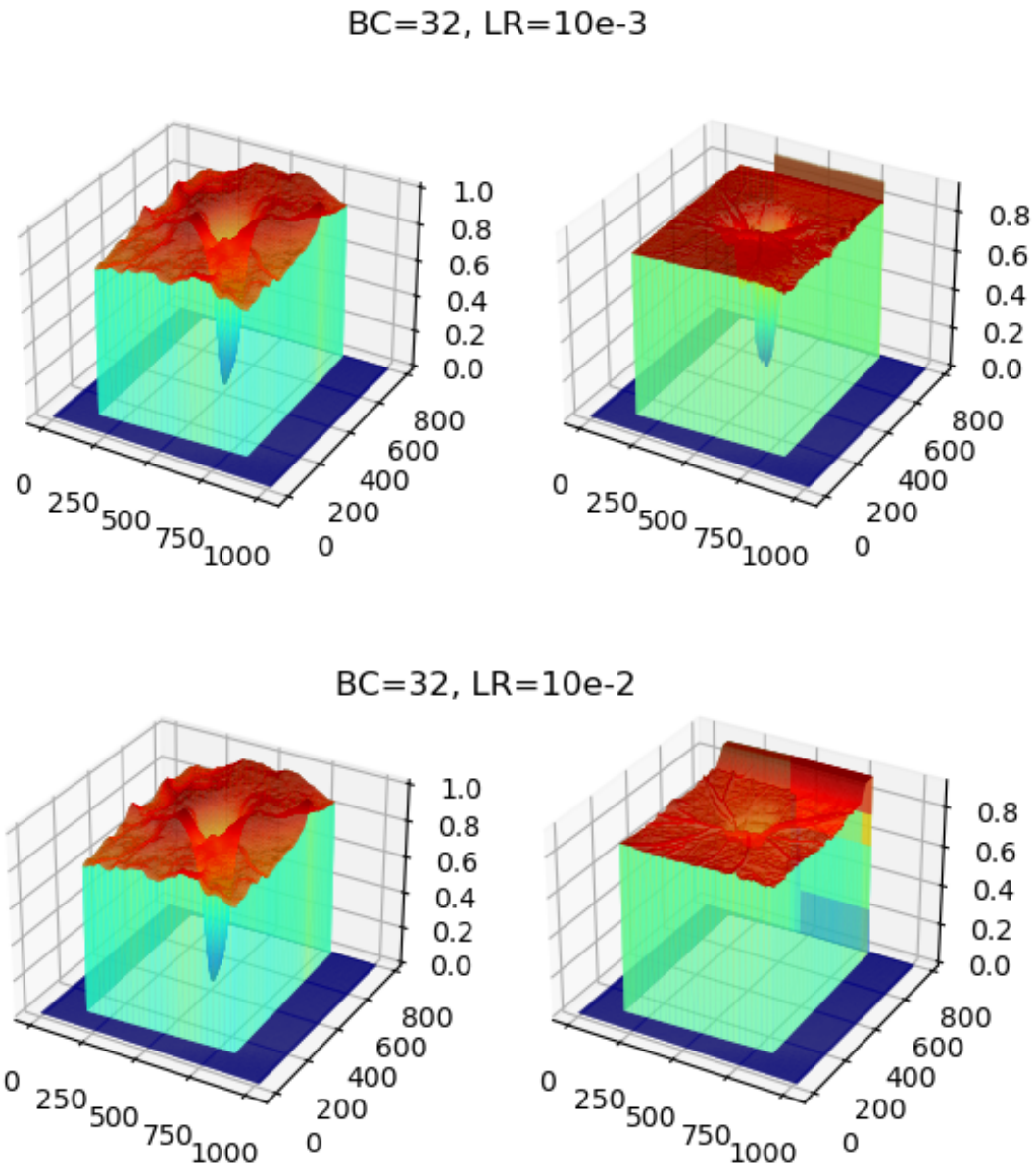

Figura 5.3: Representación tridimensional de los resultados obtenidos por las redes convolucionales de 32 canales base

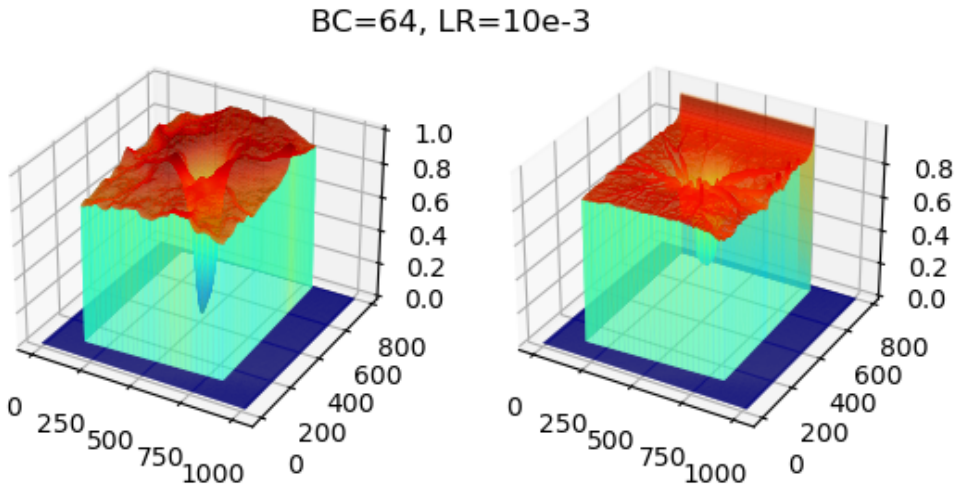

Figura 5.4: Representación tridimensional de los resultados obtenidos por las redes convolucionales de 64 canales base

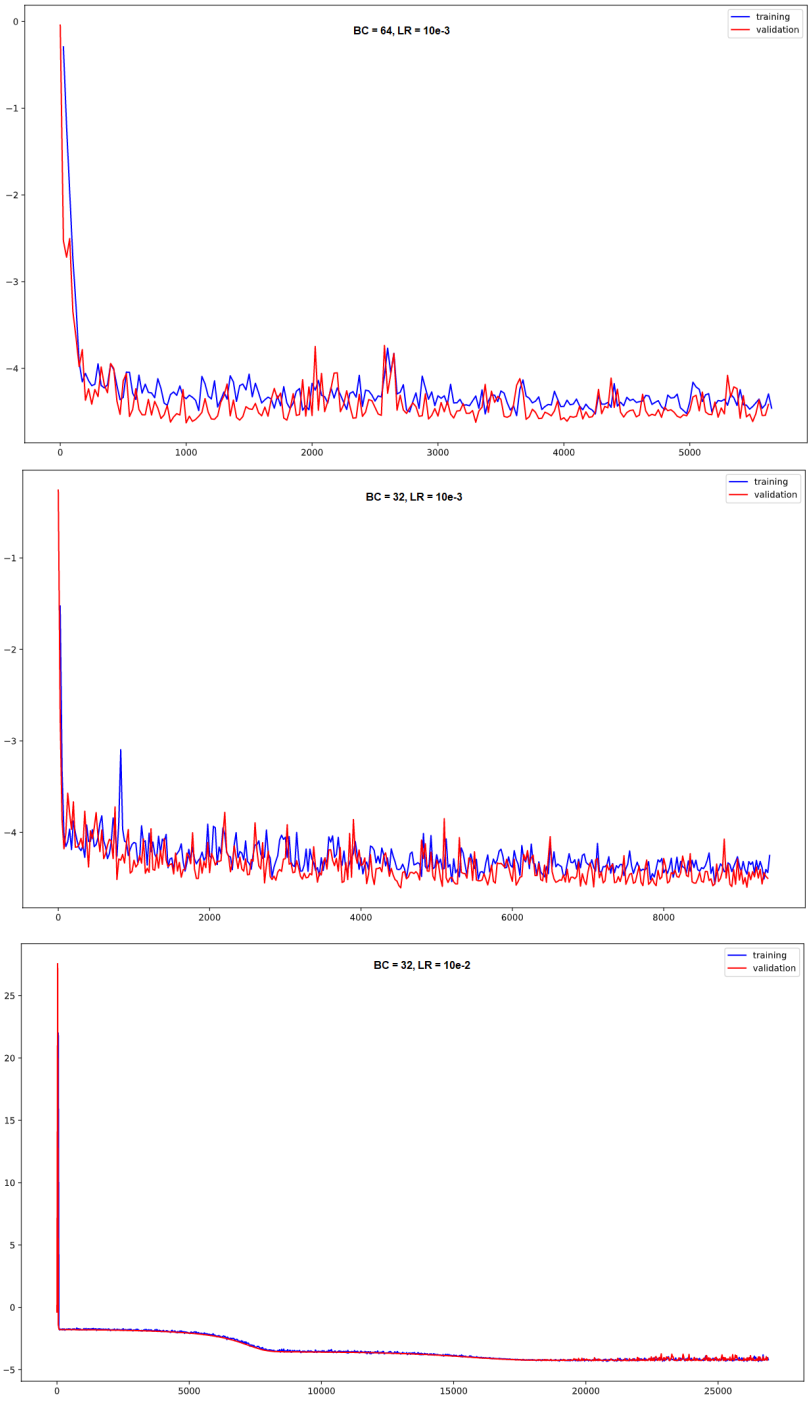

Figura 5.5: Curvas de aprendizaje de las diferentes redes convolucionales en el problema de predicción del mapa de profundidad

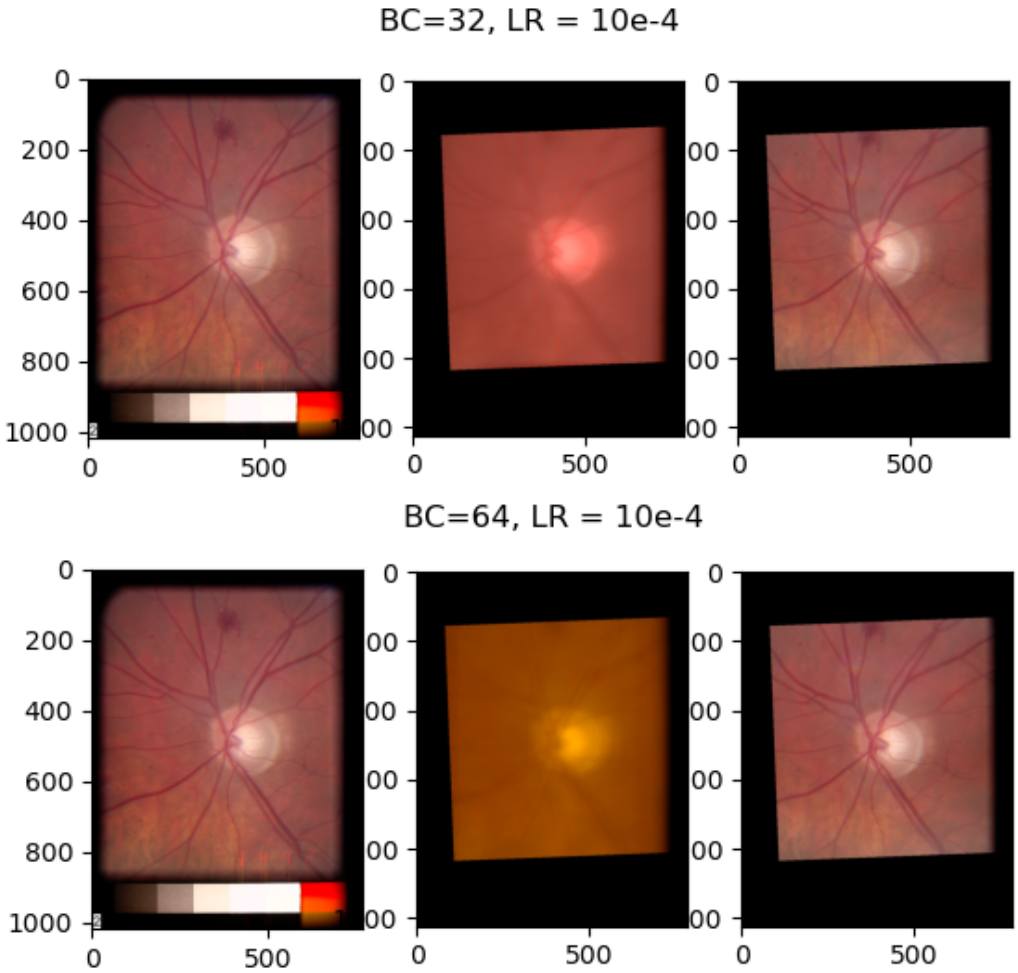

Figura 5.6: Representación de los resultados obtenidos por las redes convolucionales de 32 y 64 canales base en la tarea de predicción del par estereográfico

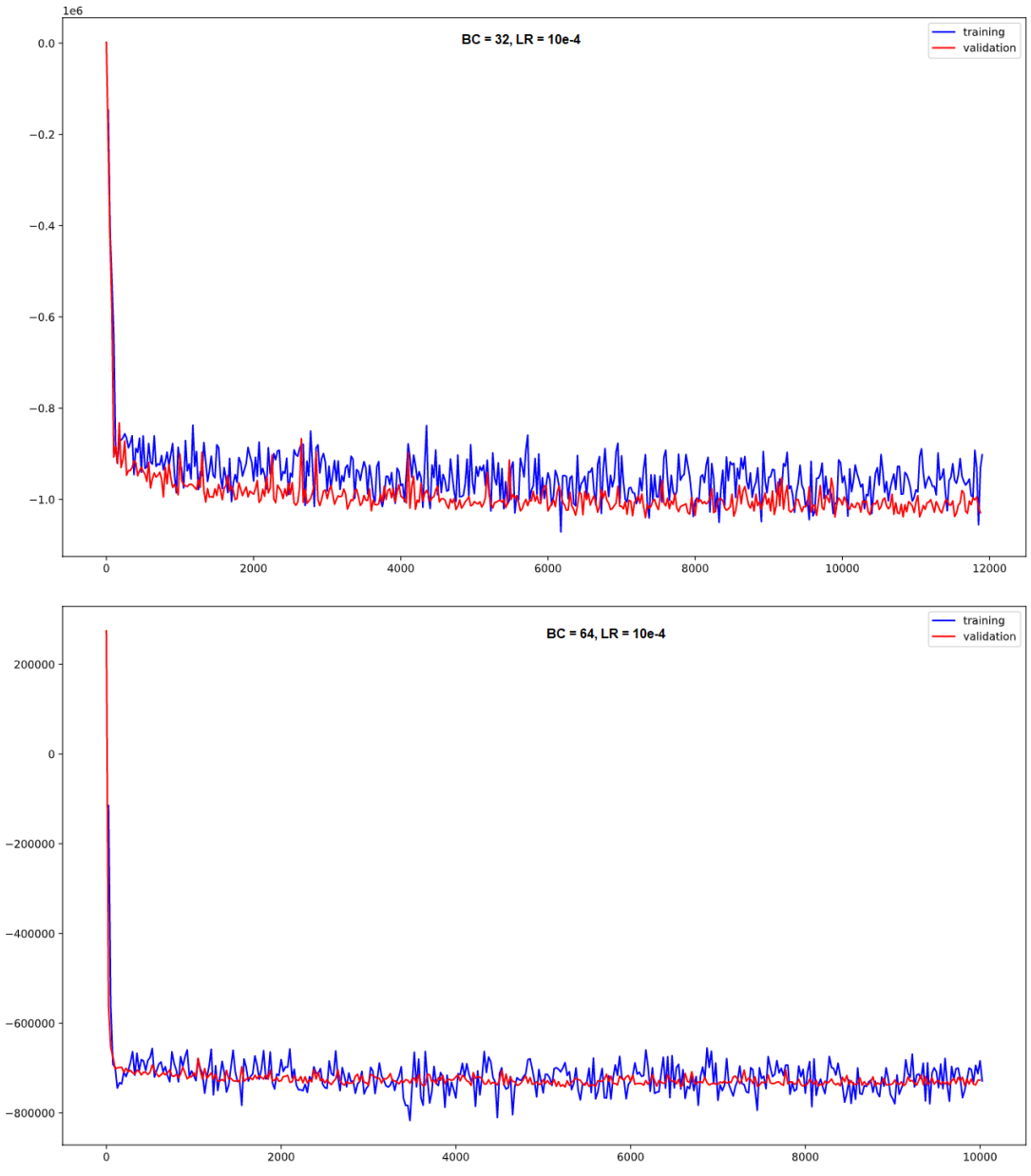

Figura 5.7: Curvas de aprendizaje de las diferentes redes convolucionales en el problema de predicción del par estereográfico

Capítulo 6

Predicción de segmentación disco-copa a partir de retinografía monocular

La tarea finalista del proyecto, como se ha explicado con anterioridad (sección 4.7, página 44), es conseguir la segmentación de disco y copa de a partir de una retinografía monocular. En este capítulo explicaremos la experimentación abordada para este fin, mostraremos los resultados obtenidos y los discutiremos brevemente antes de realizar la conclusión final.

6.1 Experimentación

El experimento consiste en sucesivas ejecuciones de la red convolucional preparada para la tarea de segmentación. Por un lado, ejecutaremos la red inicializando los pesos de manera aleatoria, estableciendo una línea base que servirá para comparar con las otras ejecuciones. Una vez establecida la línea base, ejecutaremos la red con los pesos obtenidos por el subsistema de predicción de información de profundidad, tanto de la aproximación de predicción del mapa de profundidad como de la aproximación de la predicción del par estereográfico.

En concreto, los pesos que se utilizarán serán los obtenidos en las siguientes redes por haber obtenido mejores resultados (secciones 5.1.3 y 5.2.3, páginas 56 y 58):

- Mapa de profundidad: 32 y 64 canales base con *learning rate* 10^{-3}.
- Par estereográfico: 32 y 64 canales base con *learning rate* 10^{-4}.

Una vez obtenidos los resultados de las seis ejecuciones, pues se realizará el experimento tanto en una arquitectura con 32 como con 64 canales base, se mostrarán frente a las entradas

		PRE-ENTRENAMIENTO		
		Línea base	**Mapa profundidad**	**Par estereo**
BC	**32**	0.8984	0.0951	0.9098
	64	0.9060	0.8949	0.9044

Tabla 6.1: Métrica IOU obtenida en las diferentes redes convolucionales para la segmentación del disco

y *ground-truths* para poder hacer un análisis cualitativo de los mismos. Además, obtendremos las métricas de *intersection over union* y las curvas ROC correspondientes, así como el AUC o área bajo la curva, de manera que también podamos estudiar los resultados obtenidos de forma cuantitativa.

De nuevo, mostraremos las curvas de aprendizaje de las redes, calculadas en escala logarítmica.

Por último, discutiremos los resultados obtenidos, mostrando si el pre-entrenamiento planteado supone una mejora respecto a la línea base de la experimentación o no.

6.2 Resultados parciales

En la figura 6.1 (página 69) se pueden observar ejemplos de los resultados obtenidos por la red convolucional, imagen del centro, frente a la entrada, imagen de la izquierda, y el *ground-truth*, imagen de la derecha. Se dividen en filas dependiendo de a que experimento corresponden, es decir, la primera fila es para la línea base de experimentación, la segunda para la ejecución utilzando los pesos de la predicción del mapa de profundidad y la última para la ejecución utilizando los pesos de la predicción del par estereográfico.

La figura 6.2 (página 70) es una representación de las mismas características pero para los resultados obtenidos por la arquitectura de 64 canales base.

La figura 6.3 (página 71) muestra las curvas de aprendizaje obtenidas para las redes durante el entrenamiento.

La tabla 6.1 (página 68) registra los valores medios de la métrica IOU para las segmentaciones de disco obtenidas en todas las imágenes del conjunto de validación por cada uno de los modelos obtenidos en el entrenamiento. La tabla 6.2 (página 71) es análoga pero para la segmentación de la copa.

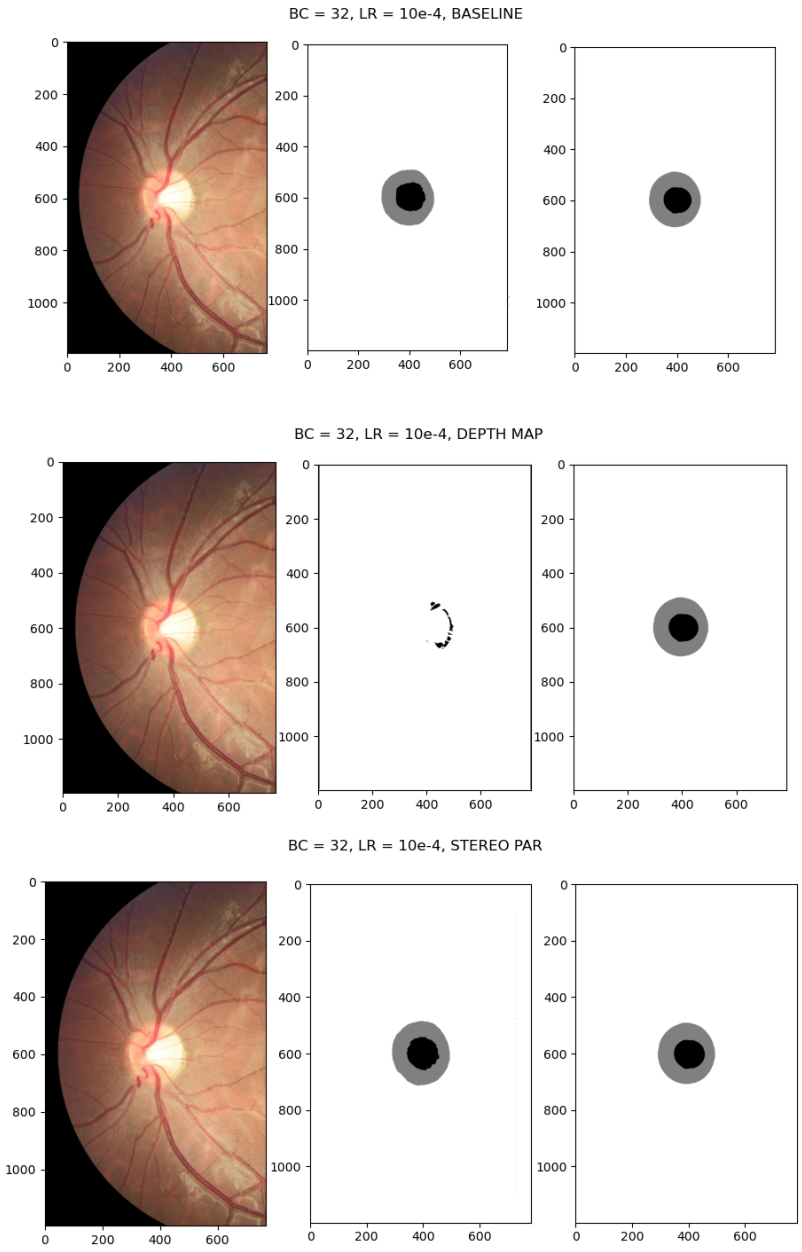

Figura 6.1: Resultados en el problema de segmentación disco-copa de las redes convolucionales de 32 canales

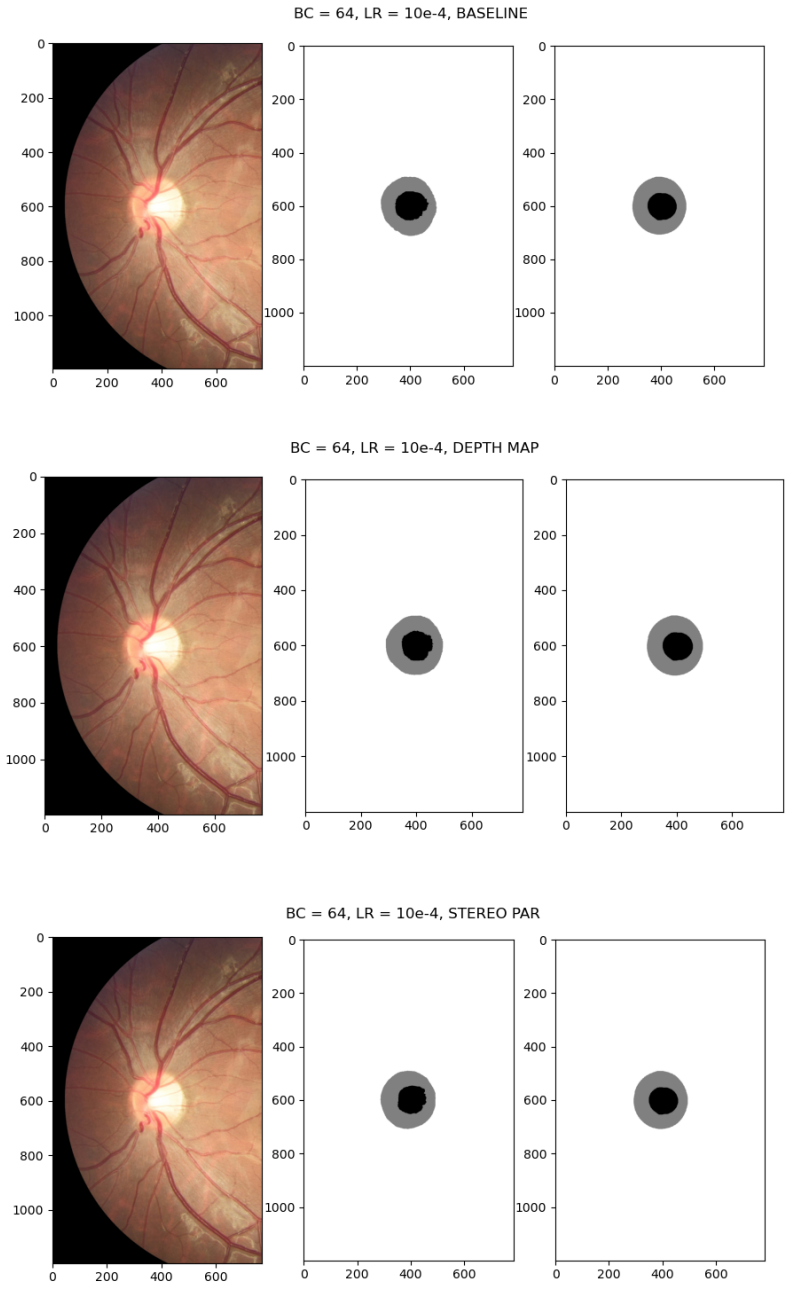

Figura 6.2: Resultados en el problema de segmentación disco-copa de las redes convolucionales de 64 canales

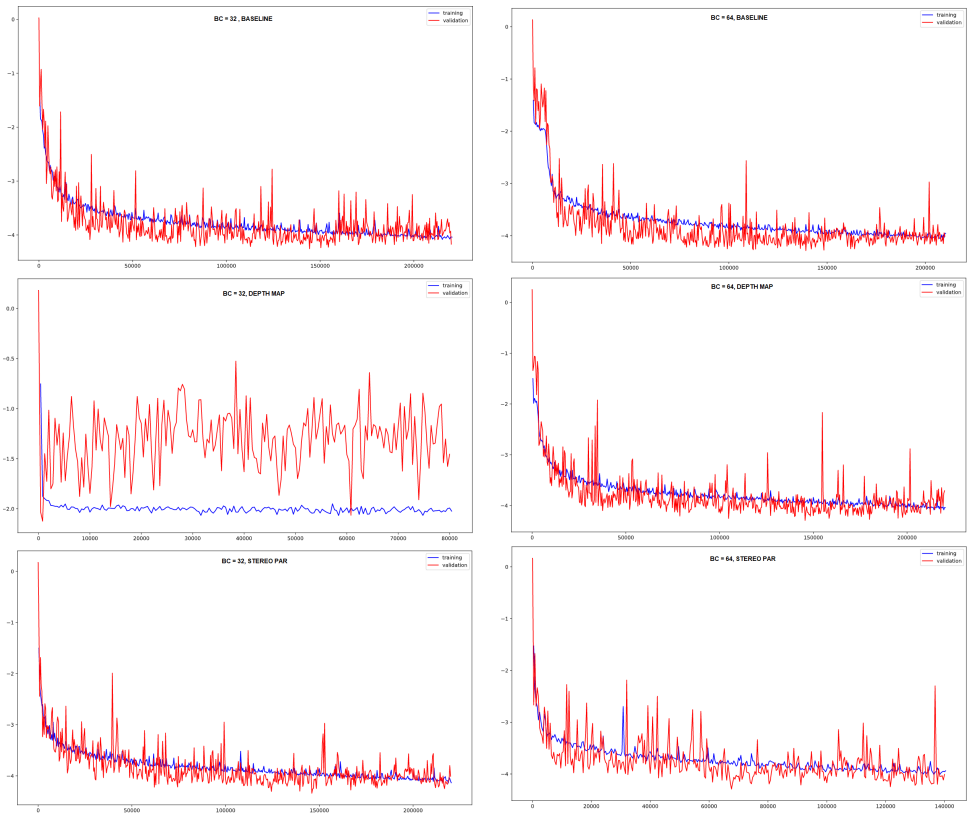

Figura 6.3: Curvas de aprendizaje en la tarea de segmentación para las redes convolucionales con 32 y 64 canales base

		PRE-ENTRENAMIENTO		
		Línea base	**Mapa profundidad**	**Par estereo**
BC	**32**	0.7672	0.1252	0.7803
	64	0.7677	0.7851	0.7797

Tabla 6.2: Métrica IOU obtenida en las diferentes redes convolucionales para la segmentación de la copa

En las figuras 6.4 y 6.5 (páginas 72 y 73) podemos observar ejemplos de curvas roc para los resultados obtenidos en el conjunto de validación.

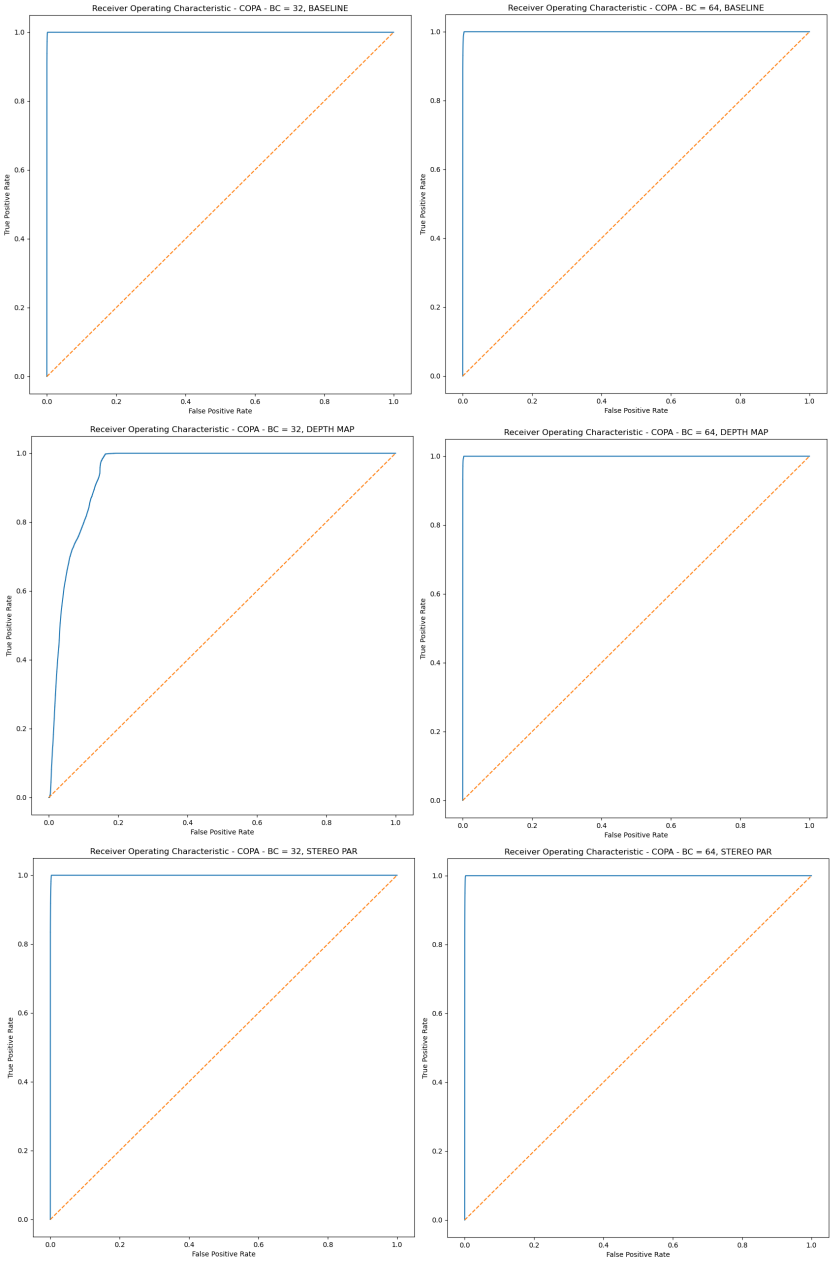

Figura 6.4: Curvas ROC para la segmentación de la copa obtenida a la salida de las redes convolucionales

Por último, las tablas 6.3 y 6.4 (páginas 74 y 74) registran los valores medios de la métrica

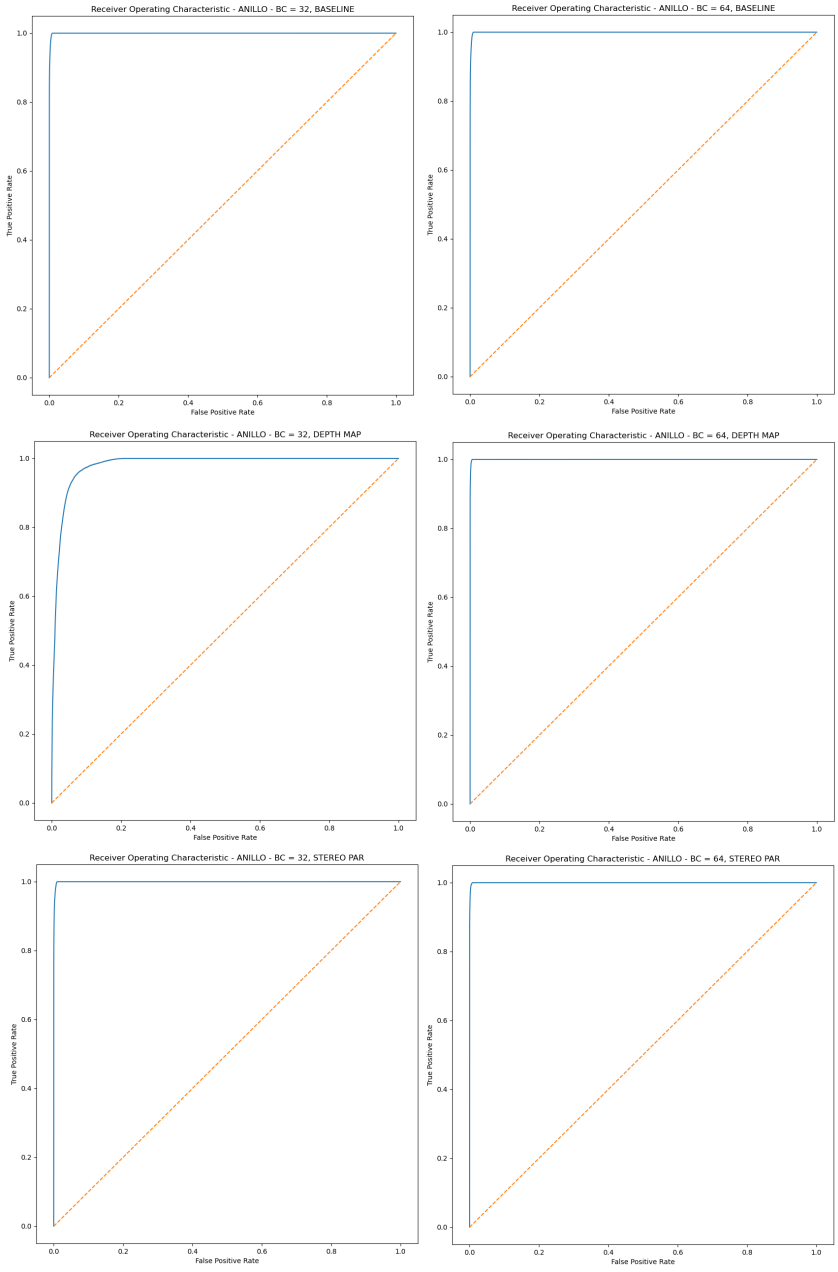

Figura 6.5: Curvas ROC para la segmentación del anillo neurorretiniano obtenida a la salida de las redes convolucionales

		PRE-ENTRENAMIENTO		
		Línea base	**Mapa profundidad**	**Par estereo**
BC	**32**	0.9997	0.9812	0.9997
	64	0.9997	.9997	0.9997

Tabla 6.3: Métrica AUC obtenida en las diferentes redes convolucionales para la segmentación de la copa

		PRE-ENTRENAMIENTO		
		Línea base	**Mapa profundidad**	**Par estereo**
BC	**32**	0.9990	0.9709	0.9991
	64	0.9990	0.9990	0.9991

Tabla 6.4: Métrica AUC obtenida en las diferentes redes convolucionales para la segmentación del anillo

AUC (área bajo las curvas ROC) para las imágenes del conjunto de validación.

6.3 Discusión de los resultados

En general, como podemos ver en las imágenes que representan las salidas frente a las entradas y los *ground-truths* (figuras 6.1 y 6.2, páginas 69 y 70), los resultados obtenidos para la segmentación de disco y copa son satisfactorios excepto para el caso de la red de 32 canales base y pre-entrenamiento en la tarea de predicción del mapa de profundidad, hecho que discutiremos a continuación. Para el resto de configuraciones de la red los resultados son bastante similares y a simple vista no podemos sacar conclusiones al respecto.

A la hora de analizar la métrica IOU (tablas 6.1 y 6.1, páginas 68 y 68) podemos ver que los resultados obtenidos son todos (exceptuando de nuevo el mismo caso mencionado antes) muy buenos (en torno a 0.9) en el caso de la segmentación del disco y aceptables (en torno a 0.7) en la segmentación de la copa. Estos son los resultados esperados pues es la segmentación de la copa la tarea con más dificultad debido a las características morfológicas y de color de la misma, resultando menos trivial de segmentar que el disco.

Si bien es verdad que en la segmentación de disco no podemos decir que el pre-entrenamiento suponga una mejora significativa, en la segmentación de la copa sí que podemos ver cómo los valores de la métrica IOU son mejores en torno a un 1-2%, valor significativo. El mejor resultado en la segmentacion de copa se obtuvo para la red con pre-entrenamiento en la tarea de predicción del mapa de profundidad y 64 canales base, seguida de las redes con pre-entrenamiento en la tarea de predicción del par estereográfico, quedando en último lugar los experimentos

de línea base. De esta manera, podemos concluir que a priori y en base a estos resultados, el *transfer-learning* desde redes que realizan tareas de predicción de información de profundidad en retinografías monoculares mejora significativamente la tarea de segmentación de la copa, al menos en los casos utilizados para la exerimentación.

Para las redes entrenadas con pre-entrenamiento en la tarea de predicción del par estereográfico, podemos ver que la que mejor resultado obtuvo fue la red de 32 canales base, pre-entrenada con los pesos de la red que obtuvo mejores resultados en la propia tarea de predicción del par estereográfico, como vimos en la sección 5.2.2 (página 57). Esto puede indicar que esta tarea de pre-entrenamiento es adecuada a la hora de realizar *transfer-learning* a una red de segmentación de disco y copa, de manera que cuanto mejor sea capaz de realizar su tarea la red de pre-entrenamiento y más refinados se obtengan los pesos, obtendremos una mejor segmentación.

En cuanto a los resultados obtenidos para las redes pre-entrenadas en la tarea de predicción del mapa de profundidad, es fácil de observar que la red de 32 canales base arrojó resultados sustancialmente peores que el resto de redes. Como podemos observar en la curva de aprendizaje de dicha red (figura 6.3, página 71), el valor de la función de coste se queda oscilando sin disminuir a lo largo de todo el entrenamiento, que se detiene mucho antes que el de las redes análogas.

Podemos atribuir esto a dos causas. Por un lado, existe la posibilidad de que la red pueda ser refinada de forma que el entrenamiento sea más largo y consiga salir del mínimo local en el que supuestamente se encontraría. Un *learning rate* más bajo, así como el forzar a que la red se ejecute durante más iteraciones podrían arrojar nuevos resultados en esta red. Sin embargo, por otro lado, resulta más plausible creer que el problema está en la tarea de pre-entrenamiento. Como hemos visto en el apartado 5.1.2 (página 54), la red de 32 canales base y *learning rate* 10^{-3}, que es la que se utiliza como pre-entrenamiento para la red de segmentación de 32 canales base, es la que obtiene mejores resultados tanto cualitativos como cuantitativos en la tarea de predicción de profundidad. Aún así, es la que peores resultados obtiene al ser usada como pre-entrenamiento en esta tarea de segmentación. La hipótesis que se plantea es que al utilizar la tarea de predicción del mapa de profundidad como tarea de pre-entrenamiento, cuantos más detalles de bajo nivel de la imagen de entrada aprenda a representar la red, peores resultados dará la segmentación utilizando sus pesos como pre-entrenamiento. Esto se debería a que la red está aprendiendo detalles relacionados con su tarea que no ayudan a la resolución de la segmentación por estar demasiado ligados a las imágenes de entrada en función de la salida que se desea obtener, siendo conocimiento inadecuado para su transferencia.

Esta hipótesis se apoya en el hecho de que la red de 64 canales base, que como vimos se quedaba más a medias entre aprender a representar la características detalladas de la imagen de entrada y la generalización de profundidad, consigue mejores resultados a la hora de ser utilizada como pre-entrenamiento en la segmentación, mejorando incluso los resultados de la línea base y del pre-entrenamiento de predicción del par estereográfico.

Para comprobar esta hipótesis, sería conveniente por un lado intentar refinar la red de segmentación todo lo posible para ver si se pueden obtener buenos resultados con los pesos obtenidos en la red de predicción del mapa de profundidad de 32 canales base. Si una vez hecho esto se siguiesen sin obtener buenos resultados, sería adecuado plantear un nuevo entrenamiento en la tarea de profundidad en el que se realizasen menos iteraciones (aproximadamente la mitad, pues la red de 64 canales entrenó la mitad que la de 32, como podemos ver en la figura 5.5, página 63) y ver si con este aprendizaje menos exhaustivo la red es capaz de adquirir los conocimientos generales que le permitan ayudar a mejorar la tarea de segmentación de copa al realizar *transfer-learning*.

En cuanto a los resultados representados mediante curvas ROC (figuras 6.4 y 6.5, páginas 72 y 73), podemos ver que las curvas se acercan mucho al valor deseado en la mayoría de los casos, excluyendo de nuevo la red con pre-entrenamiento en la tarea de predicción del mapa de profundidad y 32 canales base. En las propias curvas es difícil apreciar si realmente existe una mejora con el pre-entrenamiento, pues todas tienen valores muy cercanos. Lo que sí que podemos observar es que en esta representación, la segmentación de la copa obtiene mejores resultados que la segmentación del anillo, pues como podemos ver, las curvas son más acentuadas en el caso del anillo.

Las tablas que muestran la métrica AUC (tablas 6.3 y 6.4, páginas 74 y 74) nos sirven para confirmar este hecho, pues vemos que los valores para la copa son ligeramente más altos que para el anillo.

Capítulo 7

Conclusiones

Este último capítulo hablará sobre las conclusiones resultantes de la elaboración del proyecto, así como sobre posible trabajo futuro a abordar para completar la línea de investigación que plantea su desarrollo.

7.1 Conclusiones

El proyecto ha sido claramente diferenciado en dos partes. Por un lado, tenemos un sistema mediante el cual queremos predecir información de profundidad a partir de retinografías monoculares. Por otro lado, tenemos un sistema que se aprovecha del conocimiento obtenido en el primero para obtener una segmentación de disco y copa también sobre retinografías monoculares. El objetivo final que se buscaba alcanzar consistía en demostrar la validez del *transfer-learning* a la hora de realizar tareas contenidas dentro del mismo campo semántico y de manera más práctica utilizarlo para la ayuda a la hora de diagnosticar la enfermedad del glaucoma mediante una segmentación disco-copa más eficiente y correcta.

Para resolver ambas tareas se planteó el estudio, diseño e implementación de un modelo de aprendizaje profundo, en concreto una red convolucional utilizando la arquitectura U-Net, de la que se modificaron diferentes aspectos funcionales para adecuarla a cada tarea a resolver, poniendo en relieve la importancia que tiene a la hora de resolver tareas de diferente índole el uso adecuado de funciones de coste, funciones de activación e incluso número de neuronas u otros parámetros relacionados con la propia red (capítulo 4).

La primera parte del proyecto está dividida a su vez en dos partes: la predicción de mapas de profundidad y la predicción de pares estereográficos (capítulo 5).

A la hora de desarrollar el sistema de predicción de mapas de profundidad hemos visto que los resultados obtenidos no son del todo satisfactorios, achacando estos resultados a la naturaleza y complejidad del problema, pues obligamos a la red a establecer una relación

entre las características de la imagen de entrada y la información de profundidad presente en el *ground-truth*, proveniente de imágenes externas a la propia retinografía. La relación entre la imagen de entrada y la salida de profundidad deseada ha demostrado ser de gran complejidad, por lo que la red encuentra dificultades a la hora de inferir dicha relación y opta por una de dos soluciones simplificadas: o bien mantener características irrelevantes de la imagen de entrada (como pueden ser las estructuras vasculares) o bien proporcionar una salida de profundidad media para el conjunto de datos de entrenamiento. Esta tarea sirvió para ver que no todos los problemas son abordables de forma directa aplicando modelos de aprendizaje profundo y que muy probablemente sea necesario hacer modificaciones en el procedimiento experimental y en la arquitectura para obtener resultados más satisfactorios. Aún así, las redes fueron capaces de aprender características de las imágenes de entrada y esbozar los mapas de profundidad de forma más o menos genérica, obteniendo algún resultado destacable.

Debido a los resultados poco satisfactorios de esta primera aproximación, se planteó la tarea de predicción del par estereográfico. En este caso, al tratarse de una tarea de traslación en la que la información que ha de estar presente en la imagen de salida ya está presente en la imagen de entrada, los resultados fueron mucho más satisfactorios. Conseguimos obtener representaciones simplificadas de los pares estereográficos, obteniendo así la información de profundidad que se deseaba.

Los modelos obtenidos en estas dos aproximaciones fueron utilizados para, mediante *transfer-learning*, transferir el conocimiento obtenido en sus tareas a la tarea de segmentación de disco y copa (capítulo 6).

Primero se estableció una línea base de experimentación, inicializando los pesos de manera aleatoria, para después utilizar los pesos obtenidos en las redes de los sistemas de predicción de información de profundidad como punto de partida y comparar los resultados obtenidos.

De esta experimentación en la segunda parte del proyecto, pudimos concluir que si bien los resultados obtenidos no demuestran que el pre-entrenamiento sea beneficioso a la hora de segmentar el disco, sí lo son a la hora de segmentar la copa, que es realmente la tarea que supone cierta dificultad debido a sus características. Se manifiesta pues la relevancia del uso de *transfer-learning* a la hora de abordar diferentes problemáticas dentro del mismo campo semántico y en concreto del uso de la información de profundidad en el reconocimiento de patrones relevantes para la resolución del problema planteado.

En concreto, en la tarea de traslación utilizada como pre-entrenamiento parece haber una tendencia a una mejor segmentación de la copa cuanto mejor sea el resultado en su tarea inicial.

Sin embargo, si bien los mapas de profundidad no eran correctos, la utilización como pre-entrenamiento de los modelos obtenidos por las redes que aprendieron a partir de ellos sí que

supone una mejora respecto a la línea base, pero quedó plasmado que si la representación que aprende la red es demasiado detallada, el resultado es el contrario, obteniendo resultados mucho peores que con inicialización aleatoria de pesos.

Entonces, podemos concluir que, en base a los experimentos realizados, aunque ambas tareas de pre-entrenamiento mejoraron la segmentación de copa de manera significativa, es relevante la forma en que el modelo de pre-entrenamiento aprende los detalles de su tarea, pudiendo variar de una tarea a otra y teniendo que encontrar la representación más adecuada que ayude a mejorar la segmentación posterior.

7.2 Trabajo futuro

En primer lugar, los resultados han sido obtenidos operando sobre el conjunto de validación. Si bien es cierto que este conjunto no influye de manera directa en el entrenamiento, al utilizar un *scheduler* que termina el entrenamiento cuando se recude el coste en este conjunto, tiene una influencia indirecta sobre los resultados obtenidos. Aunque las métricas que se obtuvieron durante el proyecto son válidas y apuntan las tendencias generales, sería más apropiado utilizar un nuevo conjunto de datos de *test* para evaluar los modelos obtenidos, de manera que los resultados que se obtengan puedan ser utilizados de manera más absoluta y no solo como indicadores.

Por otro lado, los resultados obtenidos en la tarea de predicción del mapa profundidad (5.1.3, página 56) plantean la posibilidad de refinar la arquitectura U-Net para la resolución del problema. Sería conveniente probar diferentes combinaciones de hiperparámetros, como el *learning rate*, los canales base, e incluso modificar algunos hiperparámetros más concretos como pueden ser las variables beta del optimizador Adam. Posteriormente, podría ser conveniente modificar la propia arquitectura U-Net para adecuarla más al problema de forma que el resultado final sea capaz de integrar mejor la información de profundidad sin depender tanto de la estructura de la retinografía. También podría ser conveniente cambiar las funciones de coste por otras que hayan demostrado funcionar en ese tipo de problemas (como la BerHu en el caso de la predicción de mapas de profundidad), aunque se ha intentado utilizar las más apropiadas.

Además, como se ha dicho en la sección 6.3 (página 74) sería conveniente plantear un entrenamiento más corto para la red de 32 canales y *learning rate* 10^{-3}, de forma que no aprenda detalles tan específicos y comprobar si era realmente esto lo que estaba haciendo que el *transfer-learning* a la tarea de segmentación produjese malos resultados.

También podría refinarse la red de traslación para comprobar la hipótesis (5.2.3, página 5.2.3) de que cuanto mejor se realice esta tarea, mejores resultados da la segmentación utilizando ese pre-entrenamiento.

Por otro lado, podrían utilizarse otras tareas dentro del campo del análisis de retinografías como pre-entrenamiento para comprobar si la tendencia a mejorar la segmentación de la copa es algo general o únicamente producto de las tareas de predicción de información de profundidad. También podrían utilizarse pre-entrenamientos en tareas de predicción de información de profundidad en otras tareas que puedan aprovechar esta información.

De manera más general, incluso se plantea la posbilidad de la utilización de la misma metodología utilizada en el desarrollo del proyecto para otras tareas finalistas dentro del campo del análisis de imágenes oftalmológicas y, en concreto, en aquellos enfocados en la región del disco óptico, pudiendo combinar distintas tareas y pre-entrenamientos para ver dónde se consigue un mayor beneficio de las técnicas de *transfer-learning*.

Finalmente, se plantea el uso de los modelos obtenidos por las redes de segmentación de disco y copa para utilizar sus imágenes de salida como entrada a un clasificador que permita discernir entre pacientes enfermos y pacientes sanos, de forma que pueda comprobarse que la mejora en la segmentación supone una ayuda para el diagnóstico, objetivo final del proyecto.

Apéndices

Apéndice A

Redes de neuronas artificiales

En este apéndice se explicarán los conceptos básicos teóricos de las redes de neuronas artificiales para apoyar la comprensión del proyecto de forma más profunda.

A.1 Redes de neuronas artificiales

Los modelos computacionales que se utilizan en aprendizaje profundo reciben el nombre de **redes de neuronas artificiales** (RNAs). Son conocidas de esta manera porque están formadas por unas unidades de procesamiento llamadas neuronas artificiales que se conectan entre sí formando una arquitectura en red.

Las **neuronas artificiales** o **neuronas de McCulloch-Pitts** son unidades de cálculo inspiradas en el comportamiento de las neuronas del cerebro humano. El cálculo que realizan consiste en la suma ponderada de las entradas y un valor añadido llamado bias a cuyo resultado se le aplica una función no lineal, conocida como **función de activación** (sección A.1.1, página 84).

Matemáticamente, podemos expresarlas como:

$$output = s(w_1x_1 + w_2x_2 + ... + w_nx_n - \theta) \tag{A.1}$$

donde s es la función de activación, w_i son los pesos que ponderan la entrada, x_i es el propio vector entrada de la neurona y θ es el bias.

En la figura A.1 (página 84) podemos ver este funcionamiento de manera más esquematizada.

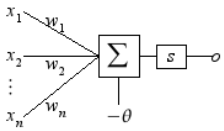

Figura A.1: Esquema del funcionamiento de una neurona artificial

Las redes de neuronas artificiales pueden combinar estas unidades de procesamiento de formas muy diferentes, tanto en número como en tipos de conectividad entre ellas y con la entrada y la salida. Conocemos estas combinaciones como **arquitecturas** y uno de los principales objetivos de este campo del aprendizaje automático es encontrar las arquitecturas que mejor resuelven cada tipo específico de tarea.

En concreto, en el desarrollo del proyecto utilizaremos arquitecturas que están englobadas dentro del marco de las **redes de neuronas convolucionales** (sección **??**, página **??**), pues ha sido demostrado a lo largo de los años que es un tipo de arquitectura que obtiene buenos resultados a la hora de procesar imágenes y en concreto en el ámbito del análisis de imágenes médicas, como podemos ver en la revisión realizada por *Laith Alzubaidi et al.* en 2021, [26].

A.1.1 Función de activación

Dentro de los conjuntos de datos con los que podemos trabajar utilizando un modelo de aprendizaje profundo, podemos encontrarnos con distribuciones muy diferentes, algunas de las cuales es imposible separar de forma lineal. En la figura A.2 (página 84) podemos ver una ilustración en la que se compara un caso separable linealmente con uno que no lo es.

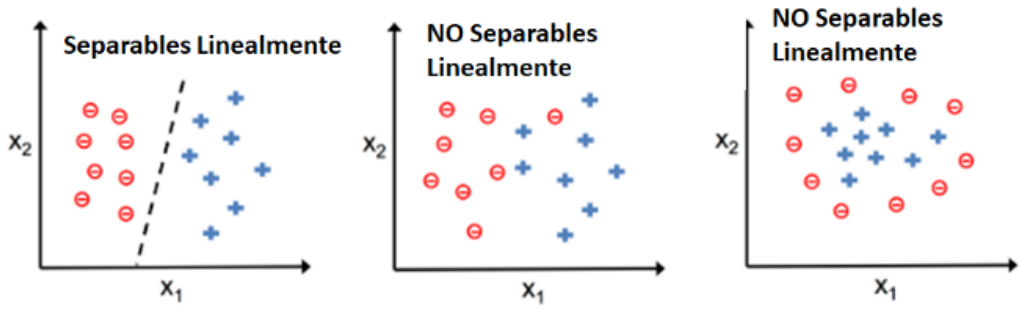

Figura A.2: Datos linealmente separables frente a datos no linealmente separables

Por este motivo, si únicamente utilizásemos una suma ponderada, operación de carácter lineal, no podríamos obtener predicciones acertadas para la gran mayoría de tareas. Es necesario entonces añadir una función no lineal que permita hacer más flexible el modelo y poder

abarcar conjuntos de datos que se distribuyan de forma no lineal en el espacio de muestras. A esta función no lineal, que se suele aplicar al final de las unidades de cómputo, se la conoce como **función de activación**. Además, las funciones de activación no lineales son necesarias para que la composición de neuronas en capas sucesivas no sea equivalente a una única capa, pues una composición lineal de funciones lineales da lugar a una función lineal.

Existen diferentes funciones de activación, como por ejemplo la función umbral, la función sigmoide, la función tangente hiperbólica, etc. Sin embargo, se escapa del propósito del proyecto explicarlas en detalle. Simplemente conviene conocer su existencia y saber que a la hora de implementar la arquitectura seleccionada será necesario elegir funciones de activación que se adecúen a las necesidades del problema justificando su uso de manera apropiada.

A.1.2 Descenso del gradiente

Hasta ahora, hemos estado hablando de que los modelos de *machine learning* aprenden a partir de los cojuntos de datos, pero no hemos dicho en qué consiste dicho aprendizaje. Si bien es verdad que los sistemas que utilizan aprendizaje automático son consideramos sistemas inteligentes, los métodos que utilizan para lo que llamamos aprender no son más que algoritmos matemáticos.

El **descenso del gradiente** es el algoritmo que se utiliza en el entrenamiento de modelos de aprendizaje profundo. Como hemos visto, el aprendizaje de las redes de neuronas artificiales consiste en el cálculo de los pesos que utilizan dichas neuronas. Este algoritmo se basa en el cálculo del gradiente de una **función de coste** (sección A.1.3, página 87) de forma iterativa en función de estos pesos.

El **gradiente** consiste en el cálculo de las derivadas parciales de una función para todos sus diferentes parámetros. Matemáticamente se expresa como:

$$\nabla f(\theta_1, \theta_2, \ldots, \theta_n) = \begin{bmatrix} \dfrac{\partial f}{\partial \theta_1}(\theta_1, \theta_2, \ldots, \theta_n) \\ \dfrac{\partial f}{\partial \theta_2}(\theta_1, \theta_2, \ldots, \theta_n) \\ \vdots \\ \dfrac{\partial f}{\partial \theta_n}(\theta_1, \theta_2, \ldots, \theta_n) \end{bmatrix} \tag{A.2}$$

donde f es la función a optimizar (función de coste), ∇ denota gradiente, θ_i son las entradas de la función (en este caso los parámetros optimizables) y $\frac{\partial}{\partial \theta_i}$ representa las derivadas parciales.

A grandes rasgos, podemos decir que la derivada de una función indica la pendiente de dicha función en el punto que se evalúe, por lo que si calculamos las derivadas parciales de una función de coste, que nos indica cuánto se diferencia la salida obtenida de la deseada, podemos saber en qué dirección (positiva o negativa) hemos de cambiar el valor de los parámetros para acercarnos a un valor mínimo de dicha función de coste, siendo esta la dirección contraria a la pendiente. Si aplicamos el proceso de manera iterativa, en cada paso, de forma ideal, estaremos más cerca del mínimo. La cantidad de cambio en las variables de entrada, es decir, lo mucho o poco que nos movemos en la función hacia el mínimo en funcion del gradiente, viene determinado por un parámetro conocido como **tasa de aprendizaje** o ***learning rate***.

Podemos ver más claramente este concepto en la figura A.3 (página 86). Es un ejemplo trivial pues se aplica a una función de una sola variable ($y = x^2$), pero sirve de ilustración para entender el funcionamiento del algoritmo.

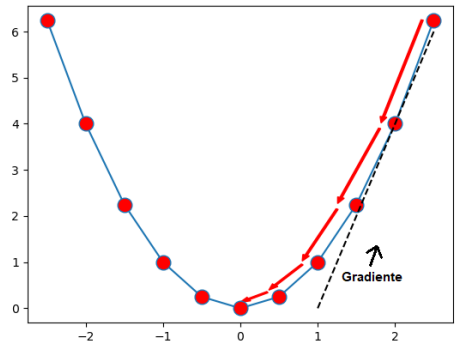

Figura A.3: Intuición del algoritmo de descenso del gradiente

No obstante, una red de neuronas artificiales está compuesta por numerosas capas aplicadas de manera sucesiva, formando una estructura compleja. El gradiente de la función de coste para todos los pesos de la red no puede calcularse aplicando directamente la definición de derivada. Para el cálculo del gradiente se utiliza un algoritmo conocido como **backpropagation** que propaga el error obtenido a la salida a todas la capas anteriores, donde cada neurona recibe una fracción del error proporcional a su aporte a la salida. Estas medidas del

error aportado por cada una de las neuronas al error final nos permiten calcular el gradiente de la función de coste respecto a los pesos. Básicamente, lo que hace el algoritmo es aplicar la regla de la cadena, que nos dice que para calcular la derivada parcial de una composición de funciones, multiplicaremos cada una de las derivadas intermedias. No entraremos en los detalles matemáticos del algoritmo pues queda fuera del alcance del proyecto, pero es interesante comprender que es gracias a él que las redes neuronales artificiales pueden utilizar el descenso del gradiente de forma eficiente utilizando estructuras tan complejas.

Por último, es importante destacar que existen múltiples variaciones del algoritmo de descenso del gradiente, conocidas como **optimizadores**. Generalmente, estas variaciones tienen como objetivo mitigar los efectos en el cálculo del gradiente de que las redes sean cada vez más grandes y utilicen cada vez mayor cantidad de pesos. Será necesario elegir cuál de estas variaciones es más eficiente a la hora de abordar el proyecto (sección 4.4.2, página 35).

A.1.3 Función de coste

La **función de coste** o ***loss function*** es una función que toma como entradas la salida de la red neuronal y el valor deseado para la misma y da como resultado un valor que es medida de la diferencia que existe entre ellas, es decir, del error cometido en la predicción.

Existen muchos tipos de funciones de coste, pero, como en el caso de las funciones de activación, lo que nos interesa es saber decidir cuáles de ellas son convenientes para las tareas que abordaremos en el proyecto, pudiendo justificar su uso.

Apéndice B

Redes neuronales convolucionales

Este apéndice tiene como objetivo explicar los conceptos básicos teóricos del funcionamiento de las redes de neuronas convolucionales, apoyando la comprensión de los métodos utilizados en la ejecución del proyecto.

B.1 Redes convolucionales

Este tipo de redes neuronales artificiales tienen como característica que los pesos que aprenden sus neuronas son **filtros convolucionales**, de forma que podemos pasar las imágenes directamente como entrada sin necesidad de extraer características previas. El objetivo de la red es aprender los valores de estos filtros, que se aplicarán sucesivamente en diferentes **capas** a las imágenes de entrada. En cada capa se irán obteniendo filtros para extraer determinadas características de las imágenes, tanto de color como morfológicas, que además atiendan a criterios espaciales. Así, de manera jerárquica, las características aprendidas por los filtros se pueden ir abstrayendo para formar con ellas estructuras más complejas y obtener por último redes capaces de realizar diferentes tareas sobre las imágenes que reciben como entrada, basándose en estas jerarquías. Por ejemplo, podemos construir clasificadores que etiqueten fotografías de gran tamaño dependiendo de su contenido, extrayendo de ellas características cada vez de más alto nivel a medida que se avanza por las capas de la red.

Podemos ver un ejemplo de arquitectura de una red convolucional convencional de clasificación en la figura B.1 (página 91). Concretamente, esta es la arquitectura LeNet, [27], y nos permite ver los tres principales tipos de capas de las redes convolucionales: las **capas de convolución**, las de **submuestreo** y **las completamente conectadas**.

B.1.1 Capas de convolución

La **convolución** es un proceso que consiste en la aplicación de un filtro convolucional a una matriz. Para la aplicación de este filtro convolucional lo que se hace es situar el centro

del filtro, que realmente es otra matriz, en cada uno de los píxeles de la matriz a la que se le está aplicando, de manera que operamos sobre todo los valores que se superponen para posteriormente integrar el resultado, que será el nuevo valor de la matriz de salida. Este proceso queda mejor detallado en la figura B.2 (página 91), [28].

En nuestro caso concreto, las matrices sobre las que aplicaremos estos filtros serán imágenes. En el campo de la visión artificial, el proceso de convolución es utilizado con frecuencia, pues la aplicación de filtros con valores determinados permite obtener características específicas de las mismas, produciendo nuevas imágenes en las que se plasma la presencia de dichas características.

En las redes de neuronas convencionales, el entrenamiento consiste en aprender los pesos de las neuronas artificiales que multiplican a las entradas. En el caso de las redes de neuronas convolucionales, lo que aprendemos son los valores de estos filtros de convolución. Por simplicidad, muchas veces haremos referencia a ellos también como pesos, pues son análogos.

Aprender estos filtros significa obtener operadores que indican la presencia de determinadas características en las imágenes de entrada, de manera que posteriormente puedan ser, por ejemplo, clasificadas.

Como a una misma imagen de entrada podemos aplicarle diferentes filtros y obtener así diferentes tipos de características, en cada capa convolucional se aplica más de un filtro convolucional de forma paralela. De esta manera, obtenemos un número de salidas igual al número de filtros aplicados. Cada uno de estas salidas se conoce como **canal de características** o ***feature channel***.

Como puede observarse, es un proceso que, si no se añaden píxeles de relleno a lo ancho y a lo alto (*padding*) sobre la imagen original, reduce el tamaño de la misma, pues el filtro no cabe en los bordes.

B.1.2 Capas de submuestreo

Generalmente, entre cada capa de filtros convolucionales se encuentran capas intermedias de mapeo no lineal y, además, **filtros/neuronas de reducción de muestreo**, que lo que hacen es reducir la resolución de la imagen de entrada en esa capa, de forma que condensan espacialmente la información relevante contenida en las imágenes y ayudan a mitigar el efecto de ruidos o variaciones en imágenes con las mismas características de alto nivel.

Actualmente, el tipo de operación de reducción de muestreo que ha demostrado mejor rendimiento es el **max-pooling**, mostrado en la figura B.3 (página 91) , que consiste en escoger el valor máximo dentro de una ventana aplicada sucesivamente a los píxeles de la imagen. Aplicando esta operación reducimos el tamaño de la imagen en un factor igual al tamaño de la ventana que utilicemos, siendo la más común la ventana de 2×2, que reduce el tamaño de la imagen a la mitad.

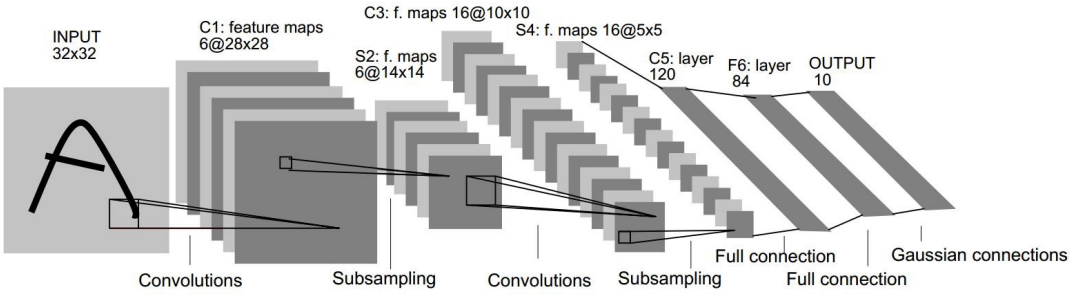

Figura B.1: Ejemplo de red neuronal convolucional: arquitectura LeNet

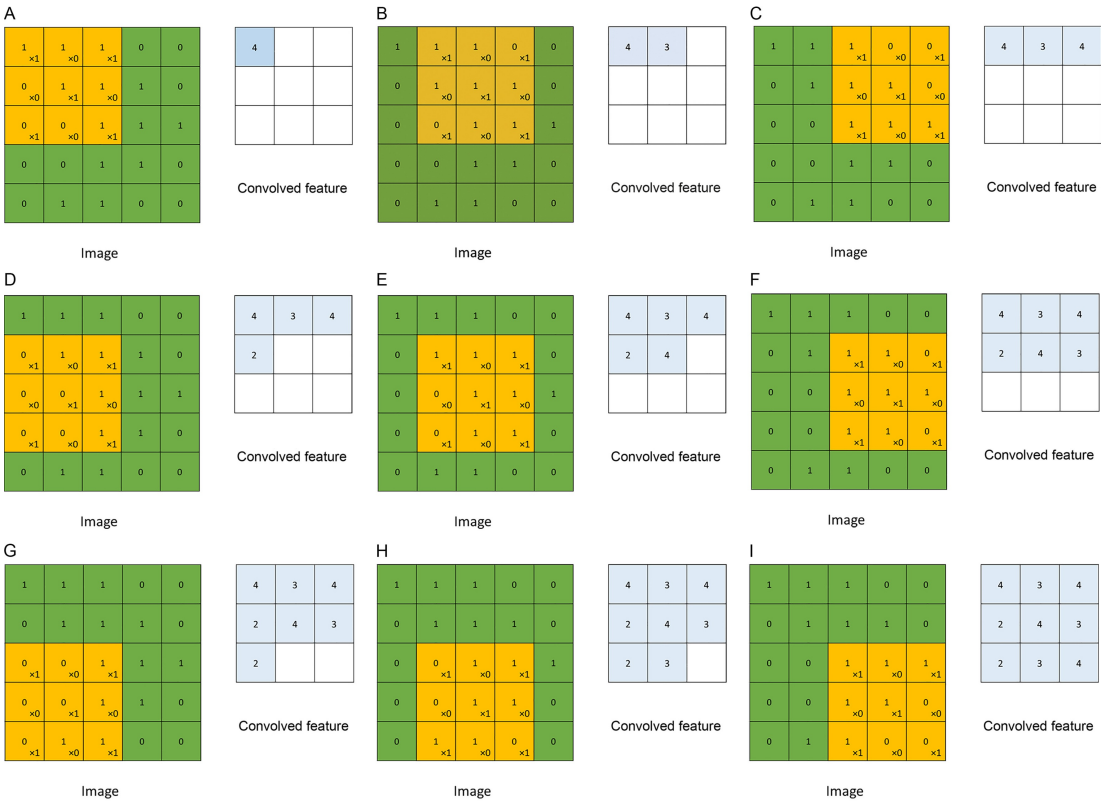

Figura B.2: Aplicación de un filtro convolucional

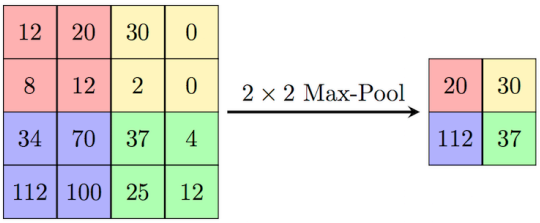

Figura B.3: Ejemplo de aplicación de max-pooling

B.1.3 Capas completamente conectadas

En las capas convolucionales las neuronas de un mismo nivel comparten pesos, es decir, utilizan el mismo filtro convolucional al aplicarlo a cada posición de la imagen. Cada neurona está conectada únicamente a una pequeña región de la entrada a esa capa de manera que en su conjunto abarquen la imagen en su totalidad. Conocemos esta región como **campo receptivo** y se corresponde con los píxeles que abarca el filtro, como podemos ver en la figura B.4 (página 92). Por ejemplo, si tenemos un filtro $3 \times 3 \times 3$, cada neurona en un nivel que aplique este filtro estará conectada a 27 píxeles de la imagen anterior, por lo que tendrá 27 pesos, compartidos con todas las neuronas de ese mismo nivel. El motivo de que las neuronas compartan pesos es que la lógica nos dice que si un filtro es útil para determinar la presencia de determinada característica en la imagen en un punto determinado de la misma, lo será también en todo el resto de la imagen. Aprender diferentes filtros para cada región de la imagen carecería de sentido, pues no aportaría a priori información sobre ninguna característica concreta.

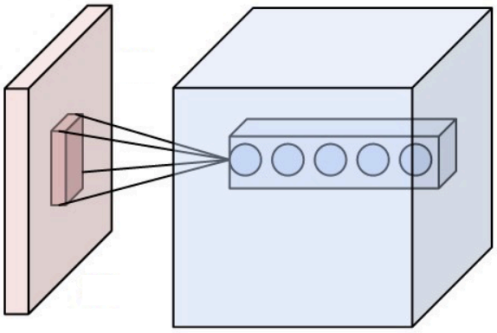

Figura B.4: Campo receptivo de una capa convolucional

Sin embargo, existen capas en las que cada neurona está conectada a la salida de todos los nodos de la capa anterior. Estas capas se conocen como **capas completamente conectadas**. Generalmente son capas que se colocan al final de la red y que actúan como lo harían las capas de las redes de neuronas artificiales convencionales, considerando la imagen de entrada como un vector de características y aprendiendo los pesos para, por ejemplo, realizar un ejercicio de clasificación.

Podemos ver la estructura de estas capas en la figura B.5 (página 93).

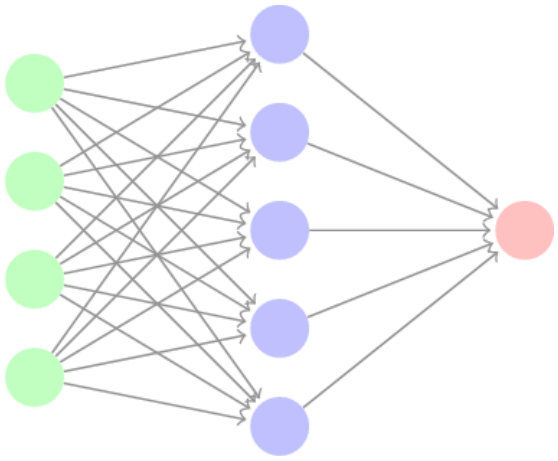

Figura B.5: Esquema de una capa completamente conectada

Lista de acrónimos

AUC Area under the curve. 51

FPR False positive rate. 51

IOU Intersection over union. 49

MSE Mean squared error. 38

OCT Optic coherence tomography. 9

ROC Receiver operating characteristic. 49

SSIM Structural Similarity. 42

TPR True positive rate. 51

Bibliografía

[1] O. Ronneberger, P. Fischer, and T. Brox, *U-Net: Convolutional Networks for Biomedical Image Segmentation*, 2015.

[2] C. Tan *et al.*, *A Survey on Deep Transfer Learning*, 2018.

[3] H. Veena, A. Muruganandham, and T. Kumaran, *A Review on the Optic Disc and Optic Cup Segmentation and Classification Approaches Over Retinal Fundus Images for Detection of Glaucoma*, 2020.

[4] R. Rangayyan, X. Zhu, F. Ayres, and A. Ells, *Detection of the optic nerve head in fundus images of the retina with gabor filters and phase portrait analysis.* J. Digit. Imag. 23(4):438–453, 2010.

[5] S. Maheshwari, V. Kanhangad, R. Pachori, S. Bhandary, and R. Acharya, *Automated glaucoma diagnosis using bit-plane slicing and local binary pattern techniques.* Pergamon, 105:72-80, 2019.

[6] M. Khunger *et al.*, *Automated detection of glaucoma using image processing techniques.* Springer Nature Singapore Pvt. Ltd., Singapore, 2019.

[7] H. Fu *et al.*, *Disc-aware ensemble network for glaucoma screening from fundus image.* IEEE transactions on medical imaging, 2018.

[8] S. Yu *et al.*, *Robust optic disc and cup segmentation with deep learning for glaucoma detection.* Comput. Med. Imag. Gr. 74:61–71, 2019.

[9] A. Sevastopolsky, *Optic disc and cup segmentation methods for glaucoma detection with modifcation of U-net convolutional neural network.* Pattern Recognit. Image Anal. 27(3):618–624, 2017.

[10] B. Al-Bander *et al.*, *Dense fully convolutional segmentation of the optic disc and cup in colour fundus for glaucoma diagnosis.* Symmetry 10:87, 2018.

[11] M. Juneja *et al.*, *Automated detection of glaucoma using deep learning convolution network (G-net).* Springer ScienceBusiness Media LLC, Berlin, 2019.

[12] H. Veena, A. Muruganandham, and T. Kumaran, *A novel optic disc and optic cup segmentation technique to diagnose glaucoma using deep learning convolutional neural network over retinal fundus images*, 2021.

[13] Y. Jiang, L. Duan, J. Cheng, Z. Gu, H. Xia, H. Fu, C. Li, and J. Liu, *JointRCNN: A Region-Based Convolutional Neural Network for Optic Disc and Cup Segmentation*, 2020.

[14] D. Natajaran, E. Sankaralingam, K. Balraj, and V. Thangaraj, *Automated segmentation algorithm with deep learning framework for early detection of glaucoma*, 2021.

[15] M. Khan and S. Anwar, *M-Net with Bidirectional ConvLSTM for Cup and Disc Segmentation in Fundus Images*, 2021.

[16] H. Almubarak, Y. Bazi, and N. Alajlan, *Two-stage mask-RCNN approach for detecting and segmenting the optic nerve head, optic disc, and optic cup in fundus images*, 2021.

[17] Álvaro S. Hervella, L. Ramos, J. Rouco, J. Novo, and M. Ortega, *Multi-Modal Self-Supervised Pre-Training for Joint Optic Disc and Cup Segmentation in Eye Fundus Images.* Centro de Investigacion CITIC, Universidade da Coruña, A Coruña, Spain.

[18] J. Morano, A. Hervella, J. Novo, and J. Rouco, *Simultaneous segmentation and classification of the retinal arteries and veins from color fundus images.* Centro de Investigacion CITIC, Universidade da Coruña, A Coruña, Spain, 2021.

[19] A. Hervella, J. Rouco, J. Novo, and M. Ortega, *Self-supervised multimodal reconstruction of retinal images over paired datasets.* Centro de Investigacion CITIC, Universidade da Coruña, A Coruña, Spain, 2020.

[20] ——, *Learning the retinal anatomy from scarce annotated data using self-supervised multimodal reconstruction.* Centro de Investigacion CITIC, Universidade da Coruña, A Coruña, Spain, 2020.

[21] A. Hervella, J. Rouco, J. Novo, M. Penedo, and M. Ortega, *Deep multi-instance heatmap regression for the detection of retinal vessel crossings and bifurcations in eye fundus images.* Centro de Investigacion CITIC, Universidade da Coruña, A Coruña, Spain, 2020.

[22] L. Tang *et al.*, *Robust Multi-Scale Stereo Matching from Fundus Images with Radiometric Differences.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 2011.

[23] "Retinal fundus glaucoma challenge dataset: https://refuge.grand-challenge.org/details/," 2018.

[24] D. P. Kingma and J. Ba, *Adam: A Method for Stochastic Optimization.* International Conference for Learning Representations, San Diego, 2014.

[25] S. M. Shankaranarayana *et al.*, *Fully Convolutional Networks for Monocular Retinal Depth Estimation and Optic Disc-Cup Segmentation.* IEEE Journal of Biomedical and Health Informatics, Volume 23, Issue 4, 2019.

[26] L. Alzubaidi *et al.*, *Review of Deep Learning: Concepts, CNN Architectures, Challenges, Applications, Future Directions.* Journal of Big Data, 2021.

[27] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, *Gradient-Based Learning Applied to Document Recognition.* Proceedings of the IEEE, 1998.

[28] V. N. Gudivada and C. Rao, *Handbook of Statistics (Volume 38): Computational Analysis and Understanding of Natural Languages: Principles, Methods and Applications.* Science Direct, 2018.