Facultade de Informática
UNIVERSIDADE DA CORUÑA

TRABALLO FIN DE GRAO
GRAO EN ENXEÑARÍA INFORMÁTICA
MENCIÓN EN COMPUTACIÓN

Registro de imagen oftalmológica utilizando Deep Learning

Estudante: Xin Lin
Dirección: José Rouco Maseda
Manuel Francisco González Penedo

A Coruña, setembro de 2021.

A mis padres

Agradecimientos

A todos los docentes por las enseñanzas que he recibido. A Manuel Franscisco González Penedo por darme la oportunidad de realizar este proyecto, a José Rouco Maseda y a Jorge Novo Buján por su guía y dedicación.

A mi familia, amigos y compañeros de trabajo por el apoyo y la paciencia.

Resumen

El registro de imagen es una etapa fundamental en el análisis de imagen médica para realizar diagnósticos o seguimiento de enfermedades en los pacientes. Los métodos clásicos de registro, requieren tiempo y, a menudo, tienen componentes específicas para cada tipo de imagen. Recientemente se han propuesto técnicas basadas en *Deep Learning* (DL) para eliminar la necesidad de diseñar extractores de características específicos para cada modalidad de imagen, además con estos modelos el tiempo de registro se reduce drásticamente. En este Trabajo Fin de Grado, se evalúa el funcionamiento de un modelo basado en DL sobre imágenes oftalmológicas. Se consiguen resultados que indican que la utilización de estos modelos sobre las mencionadas imágenes es posible, a través de un entrenamiento supervisado. Se experimentan con diferentes modalidades de imagen, aunque de forma independiente (registro monomodal), en los que se obtienen resultados similares, por lo sugiere que se puede utilizar una misma arquitectura para diferentes tipos de imágenes.

Abstract

Image registration is a fundamental step in medical image analysis for diagnosis or disease monitoring in a patient. Classical methods for image registration are of high time complexity and require specifically crafted components for different image types. Recently, techniques based on *Deep Learning* (DL) were proposed to solve the need of designing specific feature extractors for each image modality, also these models can perform the registration with lower execution times. In this work, these models are evaluated using ophthalmic images of different modalities. The results obtained indicate that these models can be used for training ophthalmic images for image registration in a supervised manner. Experimentation is performed using different modalities, although trained as monomodal registration tasks, where the similar results suggests that different type of images can be trained using the same network architecture.

Palabras clave:

- registro de imagen
- red neuronal convolucional
- imagen oftalmológica
- transformación afín
- aprendizaje máquina

Keywords:

- image registration
- convolutional neural network
- ophthalmic image
- affine transformation
- machine learning

Índice general

1 Introducción **1**

1.1 Motivación 1

1.2 Objetivos 2

1.3 Estructura de la memoria 2

2 Contexto **3**

2.1 Imagen oftalmológica 3

2.1.1 Modalidades 3

2.1.2 Utilidad clínica 4

2.2 Registro de imagen 5

2.2.1 Registro de imagen en la medicina 5

2.2.2 Metodología para el registro de imagen 6

3 Estado del arte **9**

3.1 Métodos clásicos 9

3.2 Métodos basados en *Deep Learning* 10

3.3 Diferencias con el trabajo propuesto 11

4 Metodología y planificación **13**

4.1 Metodología de desarrollo 13

4.2 Tecnologías empleadas 14

4.3 Planificación del proyecto 16

4.3.1 Fases y tareas 16

4.3.2 Planificación inicial 17

4.3.3 Coste estimado 19

4.3.4 Seguimiento 19

4.3.5 Coste final 20

5 Trabajo desarrollado **21**
5.1 Vista general del sistema 21
5.2 Descripción de datos 22
5.2.1 DRIVE 22
5.2.2 Dataset privado Retino-OCTA 23
5.3 Módulo de preprocesado 23
5.3.1 Esquema del módulo 23
5.3.2 Preprocesado de datos 24
5.4 Módulo principal 25
5.4.1 Esquema del módulo 25
5.4.2 Generador 26
5.4.3 Entrenador 31
5.5 Módulo de reporte 34
5.5.1 Esquema del módulo 34
5.5.2 Error cometido en el flujo de desplazamiento 34
5.5.3 Tasa de éxito 35
5.5.4 Diferencia en el flujo de desplazamiento 35

6 Experimentos y resultados **37**
6.1 Definiciones generales 37
6.2 Tipos de generadores 37
6.2.1 Por estrategia de generación 38
6.2.2 Por resultado generado 38
6.3 Esquema experimental 39
6.3.1 Experimentos no supervisados 40
6.3.2 Experimentos supervisados 40
6.3.3 Esquema general de los experimentos 40
6.4 Propuesta experimental 40
6.5 Registro de vasos sanguíneos 42
6.5.1 Registro no supervisado de *vessels* con Gti 44
6.5.2 Registro no supervisado de *vessels* con GtaBa 46
6.5.3 Registro no supervisado de *vessels* con GtaBa aplicando SDT 47
6.5.4 Registro supervisado de *vessels* con GtaBa 49
6.5.5 Conclusiones generales 50
6.6 Eficacia usando otras modalidades de imagen 51
6.6.1 Registro supervisado de *retinal* con GtaBa 52
6.6.2 Registro supervisado de *angiography* con GtaBa 52
6.6.3 Conclusiones generales 53

6.7 Extensión del rango de transformación 53
6.7.1 Registro con rango extendido en vasos sanguíneos 54
6.7.2 Registro con rango extendido en retinografías 57
6.7.3 Registro con rango extendido en angiografías 61
6.7.4 Conclusiones generales 65
6.8 Discusión 65

7 Conclusiones **67**
7.1 Conclusiones 67
7.2 Trabajo futuro 68
7.3 Relación con la titulación 69
7.4 Lecciones aprendidas 70

Lista de acrónimos **73**

Glosario **77**

Bibliografía **79**

Índice de figuras

2.1 Imagen oftalmológica de diferentes modalidades 4
2.2 Ejemplo de registro de imagen 6
2.3 Pasos para el registro de imagen 7

4.1 Esquema de un ciclo de la metodología de desarrollo iterativo incremental . . 14
4.2 Diagrama de Gantt del proyecto 18

5.1 Esquema general del sistema 22
5.2 Esquema del módulo de preprocesado 23
5.3 Ejemplo de angiografías y su calidad 25
5.4 Esquema del módulo principal 26
5.5 Esquema de la estrategia de generación GpTInd 29
5.6 Esquema de la estrategia de generación GpTAsoc 30
5.7 U-Net arquitectura 32
5.8 Esquema del módulo reporte 35

6.1 Tipos de generadores por resultado generado 39
6.2 Esquema genérico del entrenamiento de los experimentos 41
6.3 Ejemplo de registro con modelo no supervisado con Gti 45
6.4 Ejemplo de registro con modelo no supervisado con GtaBa 47
6.5 Ejemplo de registro con modelo no supervisado con GtaBa aplicando SDT . . 49
6.6 Ejemplo de registro con modelo supervisado con GtaBa 51
6.7 Resultado del coeficiente de Dice en *vessels* 55
6.8 Variación de la Tasa de Éxito en función del umbral elegido, en los experimentos de *vessels* 56
6.9 Diferencia en el error cometido en el flujo de los modelos SVtaAm y SVtaAm-rf comparados con SVtaBa 56
6.10 Ejemplos de registro en *vessels* 57

6.11 Resultado del coeficiente de Dice en *retinal* 59
6.12 Variación de la Tasa de Éxito en función del umbral elegido, en los experimentos de *retinal* 59
6.13 Diferencia en el error cometido en el flujo de los modelos SRtaAm y SRtaAm-rf comparados con SRtaBa 60
6.14 Ejemplos de registro en *retinal* 60
6.15 Comparativa resultados de registro entre *vessels* y *retinal* 61
6.16 Resultado de SSIM en *angiography* 62
6.17 Variación de la Tasa de Éxito en función del umbral elegido, en los experimentos de *angiography* 63
6.18 Diferencia en el error cometido en el flujo de los modelos SAtaAm y SAtaAm-rf comparados con SAtaBa 63
6.19 Ejemplos de registro en *angiography* 64

Índice de tablas

4.1 Relación de las librerías más relevantes usadas en el proyecto 15
4.2 Especificación del hardware utilizado en el proyecto 15
4.3 Estimación de la duración de las tareas 17
4.4 Coste estimado del proyecto 19
4.5 Coste final del proyecto 20

5.1 Número de imágenes buenas y malas en Retino-OCTA 24
5.2 Imágenes empleadas según la modalidad de entrenamiento 25
5.3 LT para los generadores en función de la modalidad de imagen 28

6.1 Variantes de generadores implementando la estrategia GenParTAso 38
6.2 Características de los ejemplos de registro en el grupo de experimentos de registro de vasos sanguíneos 43
6.3 Hiperparámetros del registro no supervisado de *vessels* con Gti 44
6.4 Resultados cuantitativos del experimento registro no supervisado con Gti . . . 44
6.5 Métricas de los ejemplos de registro no supervisado de *vessels* con Gti 45
6.6 Hiperparámetros del registro no supervisado de *vessels* con GtaBa 46
6.7 Resultados cuantitativos del registro no supervisado de *vessels* con GtaBa . . . 46
6.8 Métricas de los ejemplos de registro no supervisado de *vessels* con GtaBa . . . 46
6.9 Resultados cuantitativos del registro no supervisado de *vessels* con GtaBa aplicando SDT 48
6.10 Métricas de los ejemplos de registro no supervisado de *vessels* con GtaBa aplicando SDT 48
6.11 Hiperparámetros de experimentos supervisados 49
6.12 Resultados cuantitativos del registro supervisado de *vessels* con GtaBa 50
6.13 Métricas de los ejemplos de registro supervisado de *vessels* con GtaBa 50
6.14 Resultados cuantitativos del registro supervisado de *retinal* con GtaBa 52
6.15 Resultados cuantitativos del registro supervisado de *angiography* con GtaBa . 53

6.16 Resultados de los experimentos supervisados de *vessels* con generadores en diferentes rangos 55
6.17 Resultados de los experimentos supervisados de *retinal* con generadores en diferentes rangos 58
6.18 Resultados de los experimentos supervisados de *angiography* con generadores en diferentes rangos 61

Capítulo 1

Introducción

En este capítulo se explica la motivación y los objetivos del trabajo, así como la estructura que tiene la presente memoria.

1.1 Motivación

En el ámbito oftalmológico se analizan imágenes de distintas modalidades para obtener una visión más detallada de diferentes aspectos anatómicos y funcionales para realizar un diagnóstico sobre un paciente. Estas modalidades son resultado de la aplicación de diversas técnicas y metodologías para la captura de las imágenes, como por ejemplo la retinografía, la angiografía, técnicas basadas en la Tomografía de Coherencia Óptica (o *Optical Coherence Tomography* (OCT)), refracción de infrarrojos, etc [1, 2]. Además, estas imágenes se toman en varias regiones oculares y en diferentes instantes temporales, que el especialista debe compilar y analizar para dar un diagnóstico.

Uno de los pasos principales para realizar este análisis es el de registrar las diferentes imágenes para tenerlas ubicadas en una misma referencia espacial. Esta tarea es repetitiva y requiere una cantidad significativa de tiempo del especialista, por lo que se propone asistir computacionalmente en la realización de la misma mediante un registro automático de las diferentes imágenes, alineando los datos disponibles para facilitar un análisis integrado.

Los métodos clásicos de registro de imágenes requieren desarrollos *ad-hoc* dependiendo de la modalidad de la imagen y su contenido, lo que dificulta su aplicación generalizada ante la creciente diversidad de casos de uso de alineamiento y modalidades de imagen.

Recientemente, se han propuesto métodos de alineamiento automático de imagen basados en *Deep Learning* (DL) que permiten entrenar de forma automática el algoritmo en base a datos de ejemplo. Esto supone una ventaja para lograr que el algoritmo se pueda adaptar fácilmente a las diferentes casuísticas para el registro identificadas. Por consiguiente, para este trabajo de fin de grado se propone la investigación de técnicas de registro basadas en DL para el caso

de la imagen oftalmológica.

1.2 Objetivos

El trabajo tiene como objetivo principal responder la siguiente pregunta: ¿Son las técnicas basadas en DL convenientes para resolver la tarea del registro de imagen oftalmológica?

A continuación se enumeran los diferentes objetivos concretos:

- Estudio y análisis de modelos de DL para el registro de imagen.
- Experimentar con imágenes oftalmológicas para responder la anterior pregunta.

1.3 Estructura de la memoria

En esta sección se presenta la estructura con la que continúa el resto de la memoria:

- Capítulo 2 (Contexto): Se lleva a cabo una explicación de los conceptos básicos de este trabajo, qué es la imagen oftalmológica y en qué consiste el proceso del registro de imagen.
- Capítulo 3 (Estado del arte): Se revisan las diferentes técnicas, tanto tradicionales como basadas en DL, que se utilizan en el ámbito del registro de imagen médica.
- Capítulo 4 (Metodología y planificación): En este capítulo se explica la forma de trabajo y la hoja de ruta que se sigue para realizar el proyecto, así como los costes asociados y un seguimiento.
- Capítulo 5 (Trabajo desarrollado): En este capítulo se explican los diferentes componentes que conforman el sistema desarrollado para llevar a cabo los experimentos del Capítulo 6.
- Capítulo 6 (Experimentos y resultados): Se exponen las diferentes propuestas experimentales del trabajo con los resultados que se obtienen y una discusión sobre ellos.
- Capítulo 7 (Conclusiones): En este capítulo se concluye el trabajo realizado, junto a unas propuestas de trabajo futuro y la relación que tiene este proyecto con la titulación.

Capítulo 2

Contexto

En este capítulo se explica el contexto en el que se enmarca el presente trabajo. Empezando por describir qué son y para qué sirven las imágenes oftalmológicas y en qué consiste el registro de imagen.

2.1 Imagen oftalmológica

En el campo de la medicina, la oftalmología es una especialidad en la que se estudia el ojo y sus enfermedades. Los oftalmólogos emplean diversidad de equipamiento y técnicas para capturar imágenes del ojo para realizar un diagnóstico o llevar a cabo un seguimiento de alguna patología. Este tipo de imagen que se conoce como imagen oftalmológica.

2.1.1 Modalidades

Existe una amplia variedad de metodologías para obtener las imágenes oftalmológicas. Desde microscopios de baja potencia con una cámara anexada, al uso de tintes que se excitan a una determinada frecuencia, hasta la refracción de luz casi infrarroja, entre otros; se logran obtener imágenes con diferentes características, lo que resulta en diversas modalidades de imagen en la oftalmología. Algunas de estas son: la retinografía, la angiografía, la tomografía por coherencia óptica, etc. [1–4]

A continuación se explica las modalidades de imagen oftalmológica que se tratan en este trabajo.

Retinografía

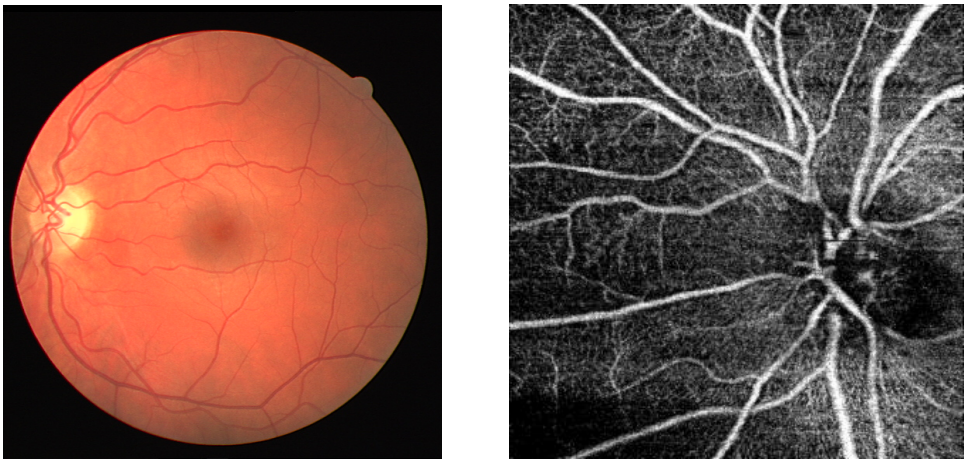
Son imágenes del fondo del ojo, capturadas mediante cámaras especializadas [2]. Los oftalmólogos emplean estas imágenes para diagnosticar, tratar y realizar un seguimiento de enfermedades del ojo, como por ejemplo la retinopatía diabética [4].

En la Figura 2.1a, se observa un ejemplo de retinografía a color.

Angiografía por Tomografía de Coherencia Óptica

La Angiografía por Tomografía de Coherencia Óptica (o *Optical Coherence Tomography Angiograhy* (OCTA)), es una variante no invasiva de las técnicas que se usan para tomar angiografías [3], que son imágenes que realzan la estructura vascular del área estudiada, en este caso del ojo. Los oftalmólogos usan este tipo de imágenes para realizar detecciones tempranas de enfermedades, como por ejemplo la degeneración macular neovascular relacionada con la edad [5].

En la Figura 2.1b, se muestra un ejemplo de una angiografía.



(a) (b)

Figura 2.1: Imagen oftalmológica de diferentes modalidades siendo la (a) una retinografía y la (b) una OCTA.

2.1.2 Utilidad clínica

El oftalmólogo captura imágenes del ojo del paciente usando diferentes metodologías para tener una visión más detallada de las estructuras del ojo. Así se tienen imágenes tomadas en de un mismo paciente con diferentes modalidades, en diferentes instantes temporales (para realizar seguimiento, por ejemplo) y también conjuntos de imágenes de una misma modalidad tomadas sobre regiones diferentes del fondo de ojo.

Para poder utilizar esta información y poder realizar un diagnóstico teniendo una visión global del estado del paciente, se tiene que fusionar estas imágenes en un espacio de referencia común. Esta tarea se conoce como registro de imagen.

2.2 Registro de imagen

El registro de imagen es una técnica que busca alinear dos o más imágenes dentro de una misma referencia espacial. Estas imágenes representan una misma escena capturada en diferentes instantes de tiempo, desde diferentes perspectivas o mediante diferentes sensores, lo que da lugar a registro de imagen multitemporal, multivista o multimodal, respectivamente [6]. Para simplificar la explicación, se hablará del registro de dos imágenes. Estas imágenes son la fija, f, que se utiliza como referencia y la móvil, m, que se intenta alinear sobre la primera mediante una transformación. Estas transformaciones se dividen de forma general en dos grupos, las transformaciones lineales (e.g. rígidas o afines) y las transformaciones no lineales (e.g. deformables).

Un ejemplo de registro de imagen con transformaciones lineales, podría ser la unión de capturas del terreno tomadas desde un dron. En cuanto al registro de imagen con transformaciones no lineales, un ejemplo podría ser la alineación de imágenes tomadas del pulmón de un paciente, dónde las deformaciones ocurren debido al movimiento del paciente o por el efecto de la respiración durante la captura de las imágenes.

En definitiva, el registro de imagen es una técnica que se aplica para encontrar una transformación a partir de dos imágenes, m y f, cuya aplicación sobre m hace que esta quede alineada geométricamente sobre f.

En la Figura 2.2 se puede ver un ejemplo de registro afín (con translación y rotación, es decir equivalente a un registro rígido) de dos parches de vasos sanguíneos del ojo.

2.2.1 Registro de imagen en la medicina

En el ámbito de la medicina, tanto en el diagnóstico como seguimiento de patologías en el paciente, el análisis de las imágenes capturadas de las estructuras del organismo es una etapa fundamental. Por lo tanto, también lo es el proceso de registro de imagen. En imagen médica prevalecen las técnicas de registro de imagen no rígido debido a la naturaleza de las imágenes, que presentan alteraciones locales debido a la respiración (e.g. al tomar imágenes de los pulmones), por la evolución de un tumor (e.g. al hacer seguimiento de uno, que con el tiempo evoluciona), o por movimientos voluntarios o involuntarios del paciente durante la toma de imágenes, entre otras causas.

En el registro de imagen médica se encuentran diferentes categorías de registro, por ejemplo: intrapaciente (las imágenes son de un mismo paciente), interpaciente (las imágenes son de múltiples pacientes), monomodal (las imágenes provienen de una técnica), multimodal (las imágenes provienen de diferentes técnicas), etc. [7]

En este trabajo se usan imágenes oftalmológicas, y el registro que se intenta llevar a cabo es de tipo monomodal con transformaciones afines (aunque se usan solo translaciones y

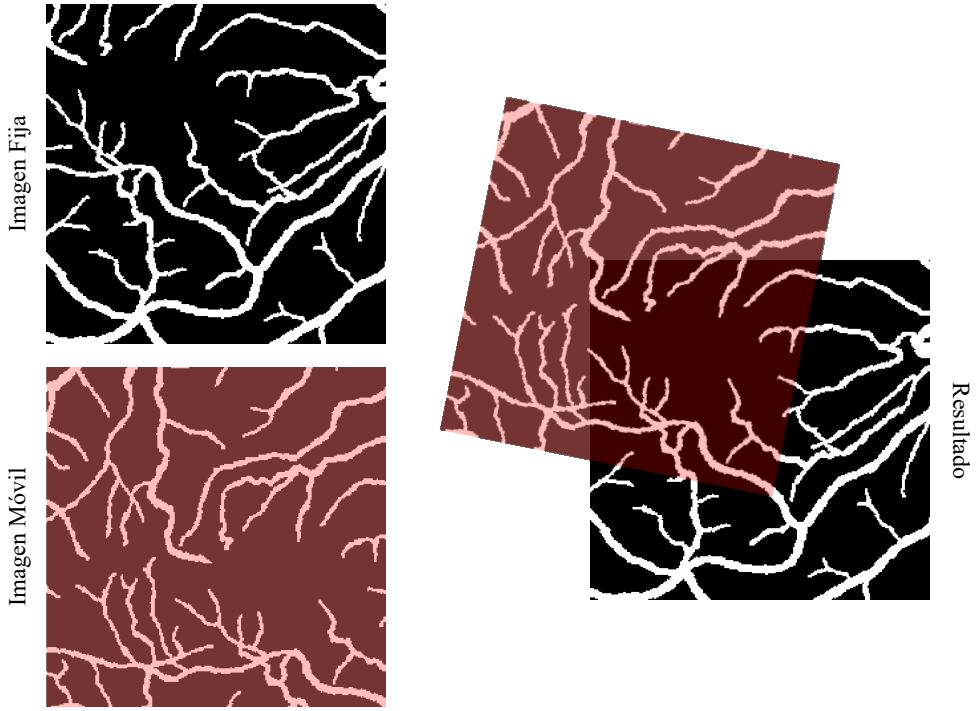


Figura 2.2: Ejemplo de registro de imagen.

rotaciones, representados mediante una matriz afín, por lo que se podría categorizar como rígido) e intrapaciente (debido a que los pares m y f se generan de manera sintética, partiendo de una misma imagen, por lo tanto de un mismo paciente). En el Capítulo 5 se detallan las características de estas imágenes y el porqué de la generación sintética.

2.2.2 Metodología para el registro de imagen

En esta sección se explica los pasos típicos que se siguen para realizar el registro de imagen, servirá para entender mejor el proceso que se sigue.

Para realizar el registro de imagen los pasos clásicos que se toman son los siguientes [6]:

1. Detección de características: Selección de características (típicamente regiones o puntos de referencia con un patrón específico) de manera manual o automática de las imágenes que se quieren registrar.

2. Correspondencia entre características: De las características anteriormente seleccionadas, se establecen relaciones entre las obtenidas a partir de ambas imágenes para emparejarlas.

3. Diseño de la función de mapeo: Se elige una función de mapeo acorde a las necesidades del problema (e.g. mapeo afín, mapeo elástico, etc). Los parámetros de esta función se

calculan con las correspondencias encontradas.

4. Transformación y remuestreo: Con la función de mapeo se transforma la imagen y se remuestrea mediante una interpolación (e.g. vecino más cercano, bilineal, etc).

En la Figura 2.3 se ilustra de forma esquemática la secuencia de los pasos habituales que se toman en el registro de imagen.

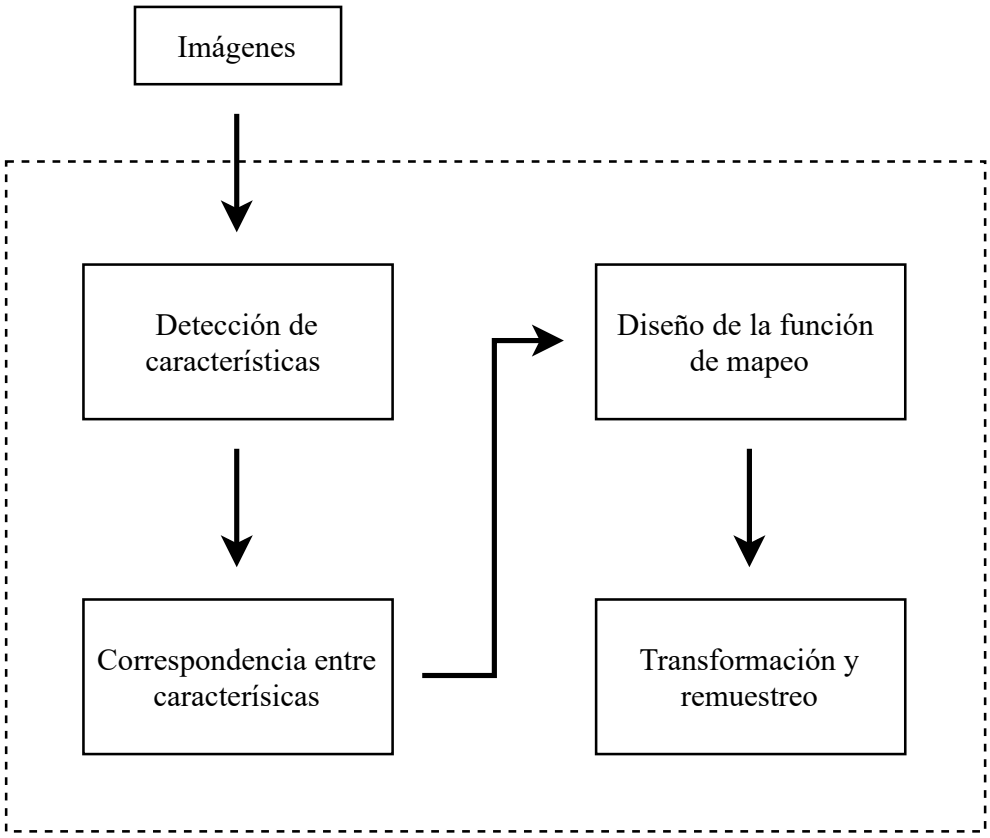


Figura 2.3: Pasos para el registro de imagen.

Capítulo 3

Estado del arte

En este capítulo se revisará el estado del arte en el ámbito del registro de imagen médica de forma general y en concreto se centrará en los estudios que se han encontrado que están más relacionados con la temática de este trabajo, es decir el registro de imagen oftalmológica. Se empieza por la revisión de los métodos de registro clásicos, seguido de los que están basados en *Deep Learning* (DL).

3.1 Métodos clásicos

Las técnicas clásicas de registro de imagen se basan en la intensidad de las imágenes y/o en las características que se pueden extraer de ellas. Con esta información, comienza un ciclo de optimización para cada par de imagen que se quiera registrar, dicho ciclo se repite por cada nuevo par de imagen que se quiera registrar.

Para la extracción de características existen diversos algoritmos generales como por ejemplo *Scale-Invariant Feature Transform* (SIFT) [8] o *Speed-Up Robust-Features* (SURF) [9]. También surgen descriptores como el *Partial Intensity Invariant Feature Descriptor* (PIIFD), que junto al detector de esquinas de Harris [10] forman parte del *framework* Harris-PIIFD [11], que se diseñó explícitamente para mejorar la detección de bifurcaciones en los vasos sanguíneos en imagen multimodal de la retina.

Posteriormente se usa un algoritmo para encontrar la correspondencia entre las características que se han extraído, con métodos como por ejemplo *Random Sampling Consensus* (RANSAC) [12], *Best Bin First* (BBF) [13], etc.

Para realizar un registro automático de OCTA para hacer un fotomontaje con el campo de visión ampliado de las angiografías, Wang et al. [14] propone un algoritmo que utiliza SURF y RANSAC.

Tang et al. [15] propone una metodología de registro no rígido de imagen oftalmológica basada en *Mixture Feature and Structure Preservation* (MFSP), en el que se utiliza como extrac-

tor de características Harris-PIIFD y las correspondencias iniciales se calculan con BBF, para posteriormente aplicar iterativamente el método de registro no rígido basado en conjuntos de puntos MFSP. Esta técnica es aplicable también para casos de registro multimodal.

El principal problema que tienen estas metodologías clásicas es que el tiempo de ejecución es lento, y que es específica para cada tipo de imagen y para cada par de registro, pues cada registro es independiente y las técnicas empleadas no aprenden, ni cambian dinámicamente. Es decir, son técnicas desarrolladas de forma específica para solventar un problema de registro dado.

Adicionalmente, existen diferentes paquetes software con funcionalidades de registro y algoritmos de registro como por ejemplo: SimpleITK[1] o ANTS[2]. Sin embargo en este trabajo no se realizan comparativas con los resultados que se pudieran obtener al utilizar este software.

3.2 Métodos basados en *Deep Learning*

Para mejorar los resultados de registro se han propuesto diferentes métodos basados en DL, entre las cuales se clasifican en diferentes categorías en función del objetivo que tiene la red [16]. Los más relevantes son los que se emplean de forma complementaria a las metodologías clásicas, donde, por ejemplo el modelo de DL es usado para medir la similaridad de las imágenes resultantes, dando lugar a redes conocidas como *Deep Similarity* (DS) [17–19] o utilizando estas redes como extractores de características [20, 21]. Por otro lado, hay otros modelos que infieren directamente la transformación necesaria para que una imagen se alinee con otra, haciendo que el registro lo haga una red de DL en su totalidad, sin necesidad de un proceso de optimización [22–24].

Con los modelos de DS ya no sería necesario diseñar funciones de similaridad *ad hoc* que mejor se adecuen a un problema dado, sino que sería la propia red la que encuentra una manera fidedigna de medir la similaridad. Esto supone una ventaja, sobre todo, para el registro multimodal, donde las imágenes pueden y suelen tener rangos de intensidad diferentes [17, 25].

Por otra parte, las redes entrenadas para regresión directa de los modelos de transformación pueden ser lineales o no lineales en función de cómo se construya la red. Los modelos se entrenaban, en su mayoría, con supervisión; aunque últimamente se están utilizando cada vez más técnicas no supervisadas [26], en parte para evitar la necesidad de disponer de la transformación real que asocia a un par de imágenes, pues en imagen médica es una tarea complicada y normalmente costosa de realizar. Al principio, en las técnicas supervisadas, estas transformaciones se obtenían con las metodologías clásicas de registro, o mediante una generación sintética de pares de registro, donde la transformación viene dada. El problema

[1] https://simpleitk.org/
[2] http://stnava.github.io/ANTs/

de utilizar imágenes sintéticas es que se necesita que la transformación sea lo más realista posible, para que la red funcione en casos reales, pues sino es así, la red haría un buen trabajo en imágenes donde la distribución de la transformación es similar al visto en el entrenamiento pero no en la realidad. Además, el tipo de registro en imagen médica más habitual es el deformable, debido a la naturaleza de las imágenes empleadas, esto dificulta aún más conseguir las transformaciones para poder entrenar de forma supervisada.

Un modelo clave, fue la *Spatial Transformer Network* (STN) [27], que permitió el entrenamiento no supervisado de modelos de registro de imagen, este módulo aplica la transformación inferida por la red a la imagen móvil para medir de forma directa la similaridad con la imagen fija.

Se han encontrado varios trabajos basados en DL que traten sobre el registro de imagen oftalmológica [24, 28–30], pero en comparación con otros tipos de imágenes médicas resultan escasos.

En [28] se entrena un modelo basado en *Convolutional Neural Network* (CNN) para extraer características de imágenes oftalmológicas para registro multimodal.

En [29] se presenta un *framework* basado en CNN para realizar el registro afín, de imágenes de diferentes modalidades, en dos etapas. En la primera etapa se realiza una segmentación del disco óptico para usarlo como característica y hacer una primera aproximación de registro, después, en la segunda etapa se utiliza una red para inferir los parámetros de una matriz de transformación afín.

En [30] se extraen segmentaciones de la estructura vascular del ojo para realizar un registro multimodal. El entrenamiento descrito en este trabajo tiene supervisión débil y se entrena generando parches de estas segmentaciones para una *SuperPoint network* con la que se extraen características, para después estimar una matriz de transformación proyectiva.

En [24] realiza un registro deformable de retinografías mediante una CNN profunda de forma no supervisada. La función de pérdida se calcula a diferentes escalas y está motivada por las características estructurales de los vasos sanguíneos que segmentan de las retinografías. A diferencia de [30], en este trabajo las imágenes que se utilizan son los escáneres completos, es decir, dónde se aprecia la circunferencia que separa el contenido del fondo; esto podría tener un impacto en los resultados que se han obtenido.

3.3 Diferencias con el trabajo propuesto

El trabajo que se propone se basa en el *framework* presentado en [23], donde se entrenará el modelo sobre imagen oftalmológica de diferentes modalidades de manera monomodal. Principalmente esto se hace para comprobar que estos modelos basados en DL pueden realizar registro de imagen de diferentes modalidades utilizando una misma arquitectura, es decir

que el tipo de imagen no influya, al menos en parte, en la capacidad de la red para completar el registro. Con el objetivo de realizar un mosaico de imágenes, las transformaciones que tienen las imágenes de entrada serán lineales. Por otro lado, las imágenes suministradas para el entrenamiento serán parches del contenido con información de las imágenes oftalmológicas, para que el borde circular no influya de ninguna manera en el entrenamiento.

Otro punto diferenciador del trabajo realizado con respecto a los que se han encontrado en la literatura, es la utilización de un modelo deformable para realizar registro lineal. En primer lugar se realiza para verificar que aún con este carácter deformable, la red puede hacer un registro lineal, algo que podría ser de utilidad en investigaciones futuras; pues las imágenes oftalmológicas también pueden presentar deformaciones, por ejemplo en los extremos cuando se captura con un campo de visión extendido [24]. En segundo lugar, si bien es cierto que hay modelos que realizan un registro afín seguido de uno deformable, como por ejemplo en [22] donde se entrenan estos componentes de forma secuencial en un planteamiento de registro en múltiples etapas. En este trabajo se intenta observar si sería posible realizar el paso afín con un modelo deformable para posteriormente en investigaciones futuras añadir componentes deformables para realizar el entrenamiento de este registro en un sólo paso.

Capítulo 4

Metodología y planificación

En este capítulo se detallan los aspectos relacionados con la metodología de desarrollo, la planificación del proyecto, los costes del mismo y las tecnologías empleadas en este trabajo.

4.1 Metodología de desarrollo

La naturaleza del trabajo no coincide totalmente con un proyecto de desarrollo software, pues aparte de construir un software sobre la que se apoya el trabajo, es necesario definir y ejecutar diferentes experimentos para evaluar que resultados se están obteniendo y en qué dirección dirigir la investigación. Por otro lado, en un proyecto de investigación los requisitos son dinámicos a lo largo del desarrollo del mismo.

La metodología que puede encajar en un proyecto de investigación, es la metodología de desarrollo iterativo incremental. Esta metodología se fundamenta en la elaboración de un producto a través de ciclos repetitivos a pequeña escala, es decir, realizar iteraciones de manera repetitiva en la que el producto al final de esta es utilizable. Una de las ventajas que tiene esta metodología, es que el desarrollador puede aplicar la experiencia adquirida en iteraciones pasadas en la actual. En la Figura 4.1 se ilustra un ciclo en esta metodología de desarrollo.

Esta metodología de desarrollo encajaría en un proyecto de investigación de la siguiente manera: un ciclo empezaría por la definición de experimentos que se quieren realizar (*Initial Planning, Requirements, Analysis & Design*), se implementa el experimento (*Implementation*) y a continuación se ejecutaría (*Deployment*), después se analizan los resultados (*Testing, Evaluation*), y en base a estos se definirían nuevos experimentos (*Planning, Requirements, Analysis & Design*).

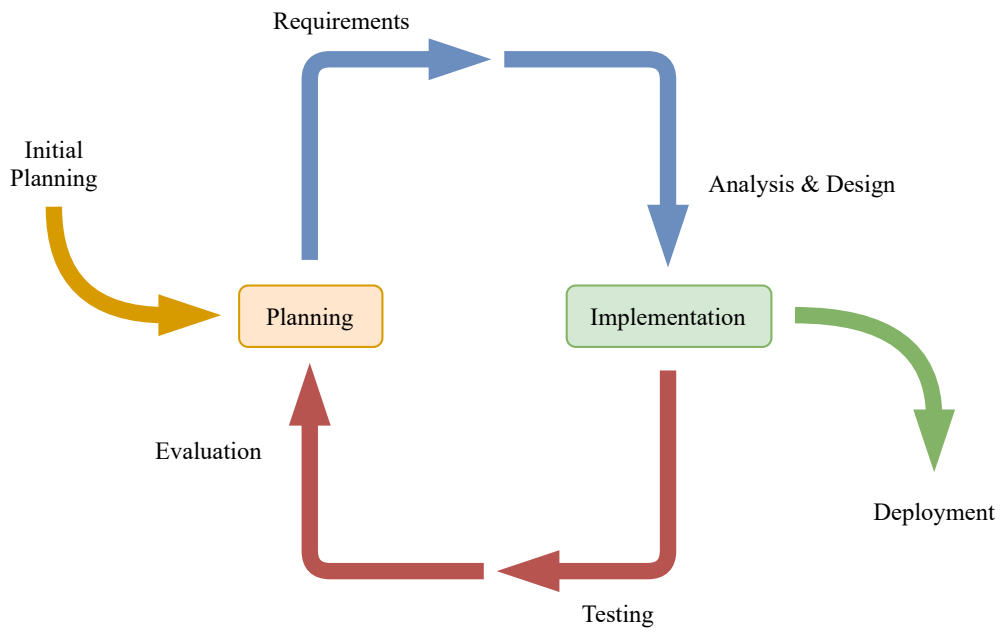


Figura 4.1: Esquema de un ciclo de la metodología de desarrollo iterativo incremental.

4.2 Tecnologías empleadas

En esta sección se hace alusión al software y a los equipos informáticos utilizados para la realización de este trabajo.

Software

En el ámbito del *Deep Learning* (DL) hay multitud de algoritmos a implementar, por lo que para facilitar el desarrollo de este trabajo se hace uso de librerías dedicadas al DL, de las cuales se tienen dos alternativas principales: PyTorch[1] y Tensorflow[2]. De estas opciones, se escoge PyTorch por su semejanza a desarrollar un proyecto Python haciendo que su curva de aprendizaje sea menor en comparación con Tensorflow. La versión elegida es la 1.5.1+cu92 debido a que en el servidor, que se mencionará a continuación, donde se ejecuta el proyecto tiene instalado CUDA[3] con la versión 9.2.

A mayores se utilizan otras librerías Python que simplifican y ayudan en el desarrollo del proyecto, como por ejemplo scikit-image para funciones relacionadas con imágenes, Matplotlib para visualizar gráficas de rendimiento de la red y Tensorboard para simplificar el seguimiento de los experimentos realizados. En la Tabla 4.1 están recogidas las librerías men-

[1] https://pytorch.org/
[2] https://www.tensorflow.org/
[3] https://developer.nvidia.com/cuda-zone

cionadas con sus respectivas versiones.

Librería	Versión
PyTorch	1.5.1+cu92
torchvision	0.6.1+cu92
scikit-image	0.18.0
Matplotlib	3.3.3
Tensorboard	2.4.0

Tabla 4.1: Relación de las librerías más relevantes usadas en el proyecto.

Por otra parte, también se utilizan las siguientes aplicaciones y servicios:

- **PyCharm**: IDE para desarrollar en Python.
- **Git**: Para control de versiones del proyecto.
- **Overleaf**: Para la escritura de la memoria.
- **Diagrams.net**[4]: Para la elaboración de los esquemas que hay en la memoria.
- **Mendeley**[5]: Para gestionar referencias bibliográficas.

Hardware

Se han usado dos equipos informáticos para la realización de este trabajo. El equipo principal, un ordenador portátil, en el que se lleva a cabo la programación, la investigación y la escritura de la memoria. En cuanto a la ejecución de los experimentos, se utiliza un servidor facilitado por el grupo de investigación VARPA[6] (Visión Artificial y Reconocimiento de Patrones), y parte de la infraestructura de cómputo GPU del CITIC[7] (Centro de Investigación en Tecnologías de la Información y las Comunicaciones). Esto es debido a que dichos experimentos son intensivos computacionalmente y para ejecutarlos, en un tiempo razonable, se precisa de hardware adecuado.

En la Tabla 4.2 se detallan los aspectos más importantes de ambos equipos.

	Ordenador portátil	Servidor VARPA
CPU	Intel Core i7-8550U	Intel Xeon E5-2650 v4
GPU	NVIDIA GeForce MX150 4 GB	NVIDIA Tesla P100 16 GB

Tabla 4.2: Especificación del hardware utilizado en el proyecto.

[4] https://www.diagrams.net/
[5] https://www.mendeley.com/
[6] http://www.varpa.es/
[7] https://citic.udc.es/

4.3 Planificación del proyecto

A continuación se describen las fases que constituyen este Trabajo Fin de Grado. A su vez cada una de estas fases se componen de tareas.

4.3.1 Fases y tareas

Primera fase: Estudio del dominio del trabajo

En esta primera fase del trabajo el objetivo es la comprensión del dominio del registro de imagen médica, aprender sobre las imágenes oftalmológicas y la lectura de artículos que traten sobre el registro de imagen en general y en las que se aplican técnicas basadas en DL. Esta fase se divide en las siguientes tareas:

1. Estudio sobre las imágenes oftalmológicas
2. Estudio sobre el registro de imagen
3. Estudio sobre técnicas basadas en DL para el registro de imagen

Segunda fase: Desarrollo del sistema, experimentación y análisis

Esta es la fase principal del trabajo, en la que se realizan desarrollos incrementales del sistema. A partir del segundo incremento se empieza con la experimentación y su respectivo análisis para decidir cómo proceder en el siguiente incremento y qué nuevos experimentos plantear. Cuando se habla de experimentos se hace referencia a los que forman parte del planteamiento experimental definido en la iteración del desarrollo. Las tareas que componen esta fase son:

1. Análisis y filtrado manual de los datos
2. Desarrollo de funcionalidades para el preprocesado de los datos
3. Desarrollo del bucle de entrenamiento genérico
4. Desarrollo experimental
5. Desarrollo de funcionalidades para visualización de resultados y cálculo de métricas

Tercera fase: Recopilación de resultados y redacción de la memoria

La última fase de este trabajo consiste en recopilar los resultados que se obtuvieron de los experimentos realizados y de la redacción de esta memoria.

1. Recopilación de resultados
2. Redacción de la memoria

4.3.2 Planificación inicial

Para realizar la planificación inicial del proyecto, se hacen unas estimaciones sobre la duración de las diferentes tareas enumeradas en la sección anterior. En la Tabla 4.3 se puede ver la estimación de la duración de cada tarea en horas, haciendo el cálculo se obtiene una duración total estimada de 325 horas.

Código	Tarea	Duración en horas
F1T1	Estudio sobre las imágenes oftalmológicas	10
F1T2	Estudio sobre el registro de imagen	10
F1T3	Estudio sobre *Deep Learning*	30
	Total primera fase:	50
F2T1	Análisis y filtrado manual de los datos	20
F2T2	Desarrollo de funcionalidades de preprocesado	20
F2T3	Desarrollo del bucle de entrenamiento genérico	30
F2T4	Desarrollo experimental	100
F2T5	Desarrollo de funcionalidades de reporte	20
	Total segunda fase:	190
F3T1	Recopilación de resultados	5
F3T2	Redacción de la memoria	80
	Total tercera fase:	85

Tabla 4.3: Estimación de la duración de las tareas.

Debido a circunstancias laborales del estudiante, la planificación se ha diseminado a lo largo de un año. Con una fecha de inicio marcada el 22 de junio del 2020 y una fecha de fin estimada para el 22 de junio del 2021, debido a que la entrega se efectuaría al día siguiente. De esta forma, queda ilustrada en la Figura 4.2 el diagrama de Gantt del proyecto. No hay una dedicación diaria homogénea, aunque se puede estimar una media de entre 1 y 2 horas diarias.

TFG	Comienzo	Fin
Primera Fase	22 de JUN	17 de SEP
F1T1	22 de JUN	14 de JUL
F1T2	14 de JUL	5 de AGO
F1T3	5 de AGO	17 de SEP
Segunda Fase	17 de SEP	31 de MAR
F2T1	17 de SEP	20 de OCT
F2T2	20 de OCT	14 de NOV
F2T3	14 de NOV	7 de DIC
F2T4	7 de DIC	31 de MAR
F2T5	5 de MAR	31 de MAR
Tercera Fase	31 de MAR	22 de JUN
F3T1	31 de MAR	15 de ABR
F3T2	5 de ABR	22 de JUN

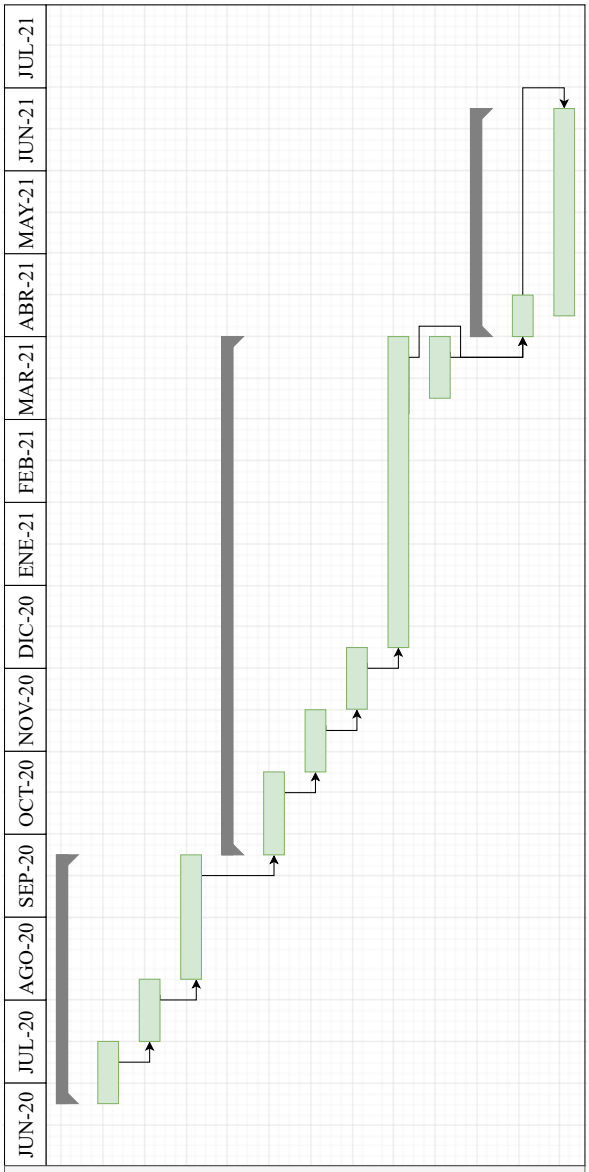


Figura 4.2: Diagrama de Gantt del proyecto.

4.3.3 Coste estimado

La estimación del coste del proyecto se ha realizado en base a los recursos humanos que han trabajado en el proyecto, es decir, un alumno y dos tutores. El coste derivado del hardware utilizado no se considera, ya que son recursos de los que ya se disponían (el portátil por parte del alumno, y el servidor por parte del grupo de investigación).

En la Tabla 4.4 se muestra la estimación de la dedicación de cada recurso en todo el proyecto, resultando en un coste estimado del proyecto de 11 125 €.

Recurso	Dedicación (horas)	Coste (€/hora)	Coste total (€)
Estudiante	325	25	8 125
Tutores	50	60	3 000

Tabla 4.4: Coste estimado del proyecto.

4.3.4 Seguimiento

Han surgido algunos imprevistos durante el transcurso del proyecto que se enumeran a continuación:

1. Disponibilidad de la GPU para realizar entrenamientos. Durante la tarea de experimentación del proyecto ha coincidido que había otros estudiantes ejecutando experimentos. Esto implicó un pequeño retraso de la fase experimental del trabajo.

2. Consecución de un modelo funcional. Se ha tardado más de lo esperado en lograr un modelo que realizase el registro planteado. Se invirtió demasiado tiempo en pruebas con diferentes hiperparámetros en los primeros modelos utilizados, sin conseguir registros satisfactorios, hasta que se comenzó a cambiar el planteamiento.

3. Factores externos al proyecto. Debido a la situación laboral del estudiante hubo algunas pausas disgregadas por todo el año, sobre todo en épocas festivas.

Estos factores han implicado un retraso para terminar de desarrollar el trabajo y en consecuencia conlleva el aplazamiento de la entrega para la siguiente convocatoria (i.e. septiembre de 2021). Teniendo en cuenta estos eventos se ha hecho un reajuste de la duración del proyecto y a causa del segundo imprevisto, hubo un incremento de 20 horas adicionales de trabajo. El primer y tercer imprevisto no han supuesto un cambio en el esfuerzo dedicado al trabajo, sino en la duración total del mismo, resultando en una extensión temporal del mismo.

4.3.5 Coste final

Debido al incremento en el tiempo de dedicación del estudiante, se ha hecho un nuevo cálculo en el coste del proyecto, así en la Tabla 4.5 se muestra este ajuste. El coste final calculado para el proyecto es de 11 625 €.

Recurso	Dedicación (horas)	Coste (€/hora)	Coste total (€)
Estudiante	325 + 20	25	8 125 + 500
Tutores	50	60	3 000

Tabla 4.5: Coste final del proyecto.

Capítulo 5

Trabajo desarrollado

En este capítulo se explica el sistema que se ha desarrollado, con el fin de facilitar la ejecución de los experimentos que se verán en el Capítulo 6. Se comienza viendo, de forma general, la composición del sistema; luego se describen los datos que se van a utilizar en el trabajo y finalmente se suceden una serie de secciones donde se detallan los diferentes módulos del sistema. Se explica como se generan los pares para el entrenamiento del registro, debido a que los datos seleccionados no constituyen en sí mismos pares con solapamiento para ser registrados, al menos no una cantidad significativa de ellos. Por último, se detalla también el modelo elegido como base para realizar los experimentos.

5.1 Vista general del sistema

En esta sección se describe de manera general el sistema desarrollado, que está ilustrado en la Figura 5.1. Se compone de tres partes principales, que son:

- **El módulo de preprocesado**: Es el encargado de realizar el preprocesado de los datos antes de ser usados para entrenar al modelo. La salida es un fichero NPZ[1], una extensión en la que se almacena un diccionario de arrays, con los datos listos para el siguiente módulo.
- **El módulo principal**: Se encarga de sintetizar la entrada y de gestionar el entrenamiento de los modelos.
- **El módulo de reporte**: Contiene funcionalidades para evaluar el modelo entrenado para obtener un fichero NPZ con métricas, así como utilidades para visualizar resultados y gráficas.

En las próximas secciones se irá concretando más cada uno de los módulos.

[1] https://numpy.org/doc/1.20/reference/generated/numpy.savez.html

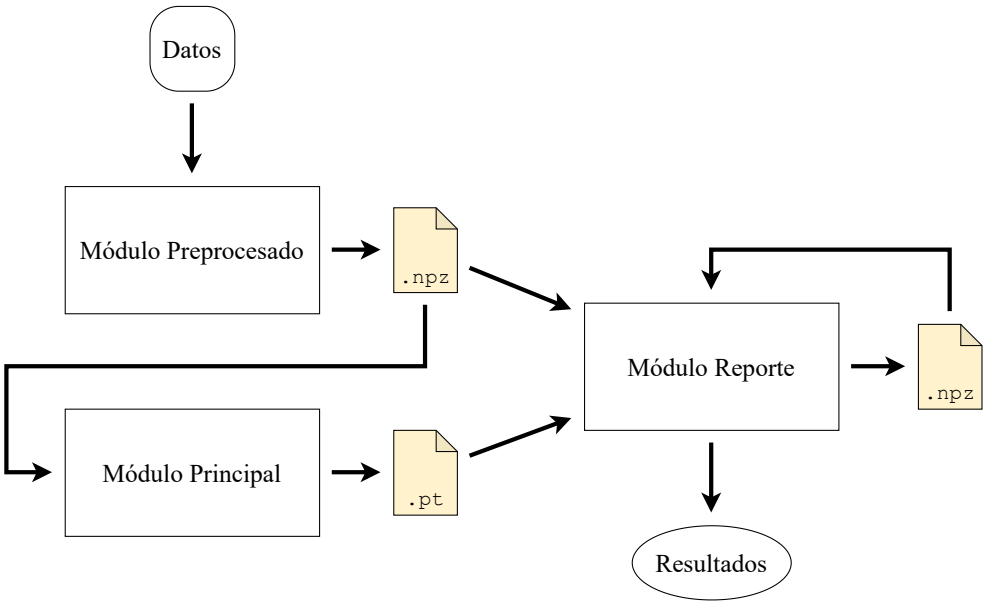


Figura 5.1: Esquema general del sistema.

5.2 Descripción de datos

Antes de proseguir con el detalle de cada uno de los módulos, se va a realizar la descripción de los datos que se han utilizado en este trabajo.

5.2.1 DRIVE

Uno de los *datasets* utilizados es *Digital Retinal Images for Vessel Extraction* (DRIVE) que, tal como indica su nombre, es un conjunto de datos elaborado con la idea de ser utilizada para entrenar modelos que segmenten vasos sanguíneos en imágenes de retina.

Para esto, dichos datos se estructuran en dos conjuntos, uno para entrenamiento y otro para test, ambas contienen 20 imágenes del fondo del ojo con sus respectivas máscaras de la región de interés (o *Region Of Interest* (ROI)) y segmentaciones de los vasos sanguíneos realizadas por oftalmólogos. De esta segmentación en el conjunto de entrenamiento hay una sola versión, mientras que en el conjunto de test cuenta con dos versiones (cada versión fue realizada por un experto diferente). Dicho de otro modo, hay en total 40 imágenes del fondo del ojo y 60 de los vasos segmentados.

En este trabajo se hace uso de las 40 imágenes del fondo del ojo con sus respectivas máscaras y segmentaciones (se toman 20 del conjunto de test, haciendo un total de 40 segmentaciones también) para la modalidad de entrenamiento usando el fondo del ojo, que se identificará como *retinal* y en el caso de emplear únicamente las segmentaciones, será identificado como *vessels*.

Además, las imágenes de este *dataset* presentan otras características, como por ejemplo que algunas de estas tienen indicios de enfermedades como la retinopatía diabética, pero no se le da importancia en los experimentos propuestos.

5.2.2 Dataset privado Retino-OCTA

Estos datos son proporcionados por los tutores y son parte del *dataset* OCTAGON [31]. En este subconjunto hay imágenes de dos modalidades, retinografías y angiografías OCT, tomadas de 12 pacientes.

De las angiografías se disponen de muestras capturadas en diferentes instantes temporales.

De este *dataset* se usan las angiografías como otra modalidad de entrenamiento, identificada como *angiography*. Estas angiografías fueron tomadas usando la técnica de OCT con el dispositivo DRI OCT Triton.

5.3 Módulo de preprocesado

5.3.1 Esquema del módulo

En la Figura 5.2 aparece el esquema del módulo de preprocesado, como se puede observar es un módulo bastante simple que recibe como entrada los datos crudos a través de una ruta y tiene como salida un fichero NPZ. Este fichero contiene los datos después del preprocesado y división en en dos conjuntos, el de entrenamiento y el de test.

El *Dataset Manager* es una interfaz que consta de un método que se tiene que implementar. Este método tiene como objetivo la creación del fichero anteriormente mencionado. Dependiendo del conjunto de datos tratado tendrá una implementación u otra, se explicará en la siguiente sección.

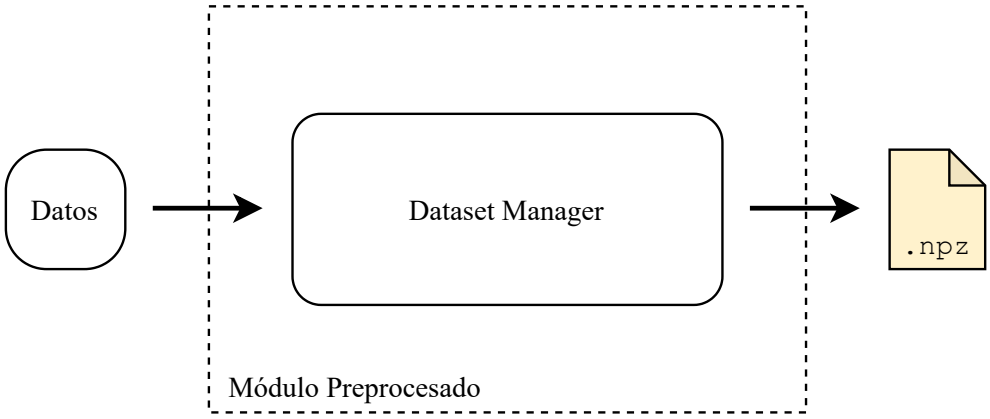


Figura 5.2: Esquema del módulo de preprocesado.

5.3.2 Preprocesado de datos

En esta sección se explica el preprocesado realizado sobre los dos *datasets* definidos anteriormente.

Preprocesado de DRIVE

Estas imágenes vienen con bastante margen y está divido en dos conjuntos, así que en primer lugar se agrupan las 40 muestras en una misma ubicación y se procede a la eliminación del margen haciendo uso de las máscaras ROI que acompañan a cada imagen. Las retinografías son a color (RGB), en este trabajo se selecciona el canal verde por presentar mejor contraste en comparación con los otros canales [32]. Obteniendo así un conjunto homogéneo de imágenes con un tamaño de 542×542 que se guarda en un fichero para su posterior uso.

Preprocesado de Retino-OCTA

En este *dataset* se han visto algunas imágenes que podrían ser poco convenientes de utilizar, por lo que se realiza en primer lugar un filtrado manual previo en la que una angiografía se considera buena si presenta las siguientes características:

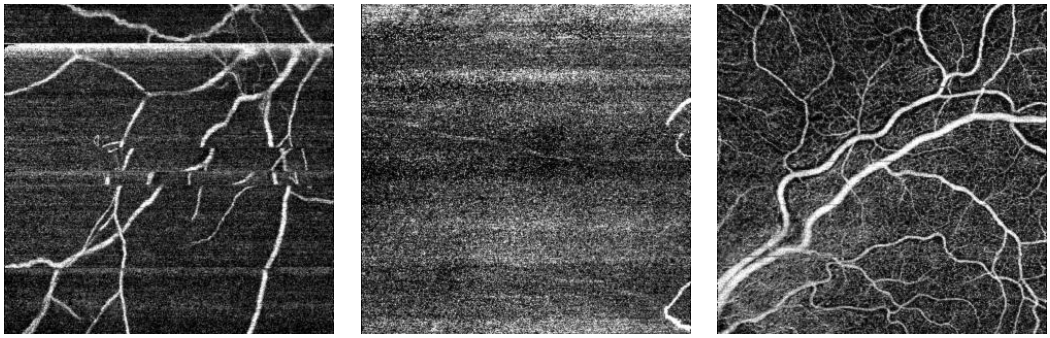
1. Tiene una cantidad significativa de vasos sanguíneos, y
2. No presenta desalineaciones dentro de la propia imagen.

En la Tabla 5.1 está listado el conteo de las buenas y malas por paciente, y en la Figura 5.3 se pueden observar ejemplos de angiografías. Hay un total de 416 imágenes buenas, de las cuales 32 tienen un tamaño de 512×512 y 384 tienen un tamaño de 320×320, de estas, se seleccionan las últimas por ser más abundantes. Así se obtiene un conjunto de imágenes 320×320 que se guarda en un fichero para su posterior uso.

Paciente	1	2	3	4	5	6	7	8	9	10	11	12
Buenas	67	13	29	27	37	14	35	51	22	40	21	60
Malas	9	36	4	0	7	17	5	11	20	17	15	6
Total	76	49	33	27	44	31	40	62	42	57	36	66

Tabla 5.1: Número de imágenes buenas y malas en Retino-OCTA

A modo de resumen, en la Tabla 5.2, se pueden ver las imágenes que se usarán en los experimentos.



(a) (b) (c)

Figura 5.3: Ejemplo de angiografías. (a) presenta desalineamiento dentro de la propia imagen, (b) no tiene casi vasos sanguíneos y (c) se considera buena.

Modalidad	Dataset	Número	Tamaño
Vessels	DRIVE	40	542×542
Retinal	DRIVE	40	542×542
Angiography	Retino-OCTA	384	320×320

Tabla 5.2: Imágenes empleadas según la modalidad de entrenamiento

5.4 Módulo principal

5.4.1 Esquema del módulo

La Figura 5.4 muestra el esquema del módulo principal del sistema, en el que se ven los dos componentes principales del mismo, el generador y el entrenador.

- **El generador**: Componente que sintetiza pares de imágenes a partir del fichero que se obtiene en el anterior módulo.

- **El entrenador**: Componente que realiza el entrenamiento y guardado del modelo (en un fichero PT[2]).

También aparecen representados componentes auxiliares como puede ser el parser de la configuración, ya que los experimentos se definen en un fichero YAML, que lee el fichero mencionado preparando el resto de componentes para ejecutar un entrenamiento. En este fichero se encuentran los hiperparámetros que se utilizan en el experimento, los parámetros de los generadores y la configuración del modelo.

[2] https://pytorch.org/docs/1.5.1/torch.html#torch.save

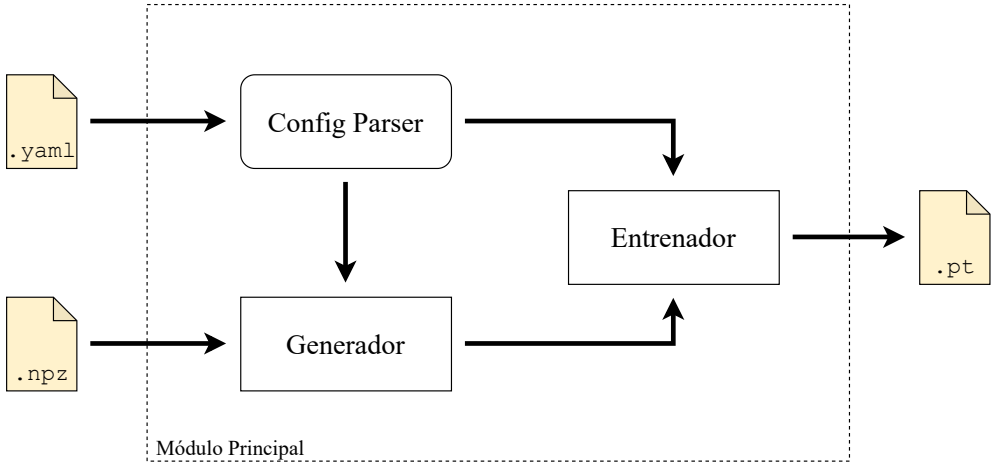


Figura 5.4: Esquema del módulo principal.

5.4.2 Generador

El `Generador` es un componente que tiene la funcionalidad de sintetizar pares de imágenes para el registro de imagen. Este componente es necesario el sistema debido a que los datos utilizados no constituyen pares de imágenes para registro o hay una cantidad limitada de ellos (como ocurre con las angiografías del dataset Retino-OCTA).

Desde el punto de vista de la implementación, el `Generador` es una clase que tiene los siguientes atributos:

- ***data***: Es el fichero NPZ obtenido del módulo anterior.
- ***transforms***: Es el Listado de Transformaciones (LT) que contiene las funciones de transformación que se aplican durante la síntesis de los pares.

También hay que definir una estrategia de generación, es decir, el algoritmo que emplea para sintetizar los pares. La salida que devuelve este componente, es en general, la imagen fija (f), la imagen móvil (m) y el flujo de transformación (Φ) que los asocia. Esto quiere decir, que: $f = m \circ \Phi$, dónde $m \circ \Phi$ es la transformación de m con el flujo Φ.

Listado de Transformaciones

El LT es una lista que contiene funciones de transformación, que se componen mediante la clase `Compose`[3] de la librería `torchvision`. La composición de transformaciones consiste en encadenar las funciones para su aplicación secuencial.

[3] https://pytorch.org/docs/1.5.1/torchvision/transforms.html#torchvision.transforms.Compose

A continuación, se enumeran las funciones de transformación implementadas en el trabajo:

- `Rotate(min_value, max_value)`. Es una función que aplica una rotación desde el centro de la imagen. El grado de la rotación se obtiene de forma aleatoria dentro del rango especificado.
- `Normalization()`. Cambia el rango de los valores de intensidad de la imagen al rango $[0, 1]$.
- `InscribedSquareCrop()`. Se hace la suposición de que la imagen de entrada es cuadrada, que se correspondería con el cuadrado circunscrito de la circunferencia de una imagen oftalmológica (e.g. retinografía en DRIVE tras el preprocesado). Asumiendo lo anterior, se recorta el cuadrado inscrito de la imagen de entrada.
- `SquareCrop(side)`. Esta función recorta la imagen de entrada desde una posición aleatoria con el tamaño especificado. El usuario deberá proporcionar un `side` acorde al tamaño de la imagen de entrada para que, al aplicar esta función dos veces sobre una misma imagen de manera independiente, se obtengan dos parches con solapamiento para realizar el registro de imagen. Por ejemplo, en el caso de DRIVE las imágenes tras el preprocesado tienen un tamaño de 542×542, por lo tanto el `side` especificado deberá ser mayor que $542/2 = 271$ para que, en el peor de los casos (primer recorte en una esquina, segundo recorte en la esquina opuesta) haya un mínimo de solapamiento.
- `SimpleBinaryConversion()`. Es una función que hace una umbralización de una imagen para convertirla en una imagen binaria. Básicamente el píxel será 1 si el valor original en la imagen es superior a 0. Por ejemplo, en las imágenes de los vasos sanguíneos de DRIVE, se emplearía esta función como última transformación para asegurarse de que se mantiene como una imagen binaria, ya que tras una rotación podría dejar de ser binaria por la interpolación aplicada.

En la Tabla 5.3 se define el LT que reciben los generadores según el tipo de imagen que va a tratar. Se puede ver que el resultado será un par de imágenes de tamaño $256 \times 256 \times 1$, es decir, una parche cuadrado de lado 256 con un único canal (imagen binaria o escala de grises).

En la siguiente sección se explican las estrategias de generación planteadas en este trabajo.

Estrategias de generación

En este sección se definen las dos estrategias de generación propuestas en el trabajo de forma genérica, posteriormente en el desarrollo experimental se concretará la salida del generador según el tipo de imagen que se utilice.

Vessels	Retinal	Angiography

`Rotate(-45, 45)`
`InscribedSquareCrop()`
`SquareCrop(256)`
`SimpleBinaryConversion()`
	`Rotate(-45, 45)`
	`InscribedSquareCrop()`
	`SquareCrop(256)`
		`Normalization()`
		`SquareCrop(256)`

Tabla 5.3: LT para los generadores en función de la modalidad de imagen.

Se proponen dos estrategias de generación:

- Generación de Pares mediante Transformación Individual (GenParTInd): Consiste en generar el par, aplicando el LT, a partir de una misma imagen.
- Generación de Pares mediante Transformación Asociada (GenParTAso): Una imagen del par se genera con la aplicación del LT, y la otra imagen del par se obtiene a través de una transformación asociada al primero.

La GenParTInd permite una generación de pares con transformaciones más heterogéneas, por ejemplo, en *vessels* o *retinal*, a las imágenes del par se aplican `Rotate(-45, 45)` y `SquareCrop(256)`, mientras que con la GenParTAso, la segunda imagen del par está condicionada a la primera. Por otro lado, debido a cómo fueron implementadas estas estrategias, con la GenParTInd las dos imágenes pueden tener información adicional que no tiene la otra, mientras que, con la GenParTAso la segunda imagen presenta regiones sin información (más o menos pronunciadas, según la transformación aplicada) ya que deriva de la primera.

A continuación, se explica con más detalle estas estrategias de generación.

Estrategia de GenParTInd

Se realiza una selección aleatoria de una imagen del fichero, y a continuación se aplica la composición del LT dos veces de manera individual sobre la imagen seleccionada. Se obtienen dos imágenes con solapamiento junto a los metadatos de las dos transformaciones, con los que se puede calcular el flujo de desplazamiento sintético que lleva una imagen a alinearse con la otra. En la Figura 5.5 se representa de forma esquemática este proceso.

Los metadatos de cada una de las imágenes tienen tres valores: el ángulo de rotación y las coordenadas, x e y, donde se realiza el corte. Sea $MTD_f = (\theta_f, x_f, y_f)$ los metadatos de la imagen fija, y $MTD_m = (\theta_m, x_m, y_m)$ los de la imagen móvil, el ángulo (θ) y el vector de traslación (x, y) que asocia ambas imágenes está definida por las siguientes ecuaciones:

$$\theta = \theta_f - \theta_m \tag{5.1}$$

$$(x,\ y) = (x_f - x_m,\ y_f - y_m) \tag{5.2}$$

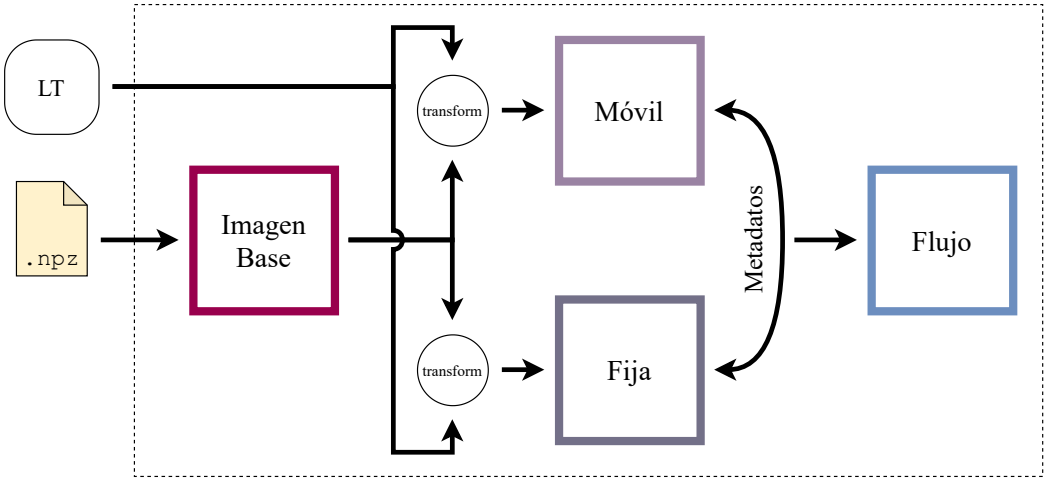


Figura 5.5: Esquema de la estrategia de generación GenParTInd.

En esta estrategia de generación, la rotación se aplica desde el centro de la imagen cuando la imagen aún es completa, es decir la rotación es previo al recorte. Por lo tanto para calcular la matriz de transformación, M, se calcula la matriz de rotación, R, desde el centro real de la imagen previo al corte y se multiplica por T, la matriz de translación. Para trasladar la imagen al centro previo al corte, se calcula de la misma manera que en la Ecuación 5.2, donde las coordenadas iniciales, (x_c, y_c), es la esquina superior izquierdo del cuadrado 256×256 centrado en la imagen completa, y las coordenadas finales las definidas en MTD_m, y luego con las coordenadas permutadas para devolverlo a su posición. Así se tiene la siguiente ecuación:

$$R' = C_1 \times R \times C_2, \text{ donde } C_i = \begin{bmatrix} 1 & 0 & x_i \\ 0 & 1 & y_i \\ 0 & 0 & 1 \end{bmatrix} \text{ y } R = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{5.3}$$

con $(x_1,\, y_1) = (x_c - x_m,\, y_c - y_m)$ y $(x_2,\, y_2) = (x_m - x_c,\, y_m - y_c)$. Por lo tanto la ecuación para la matriz de transformación en este generador es:

$$M = R' \times T, \text{ donde } T = \begin{bmatrix} 1 & 0 & x \\ 0 & 1 & y \\ 0 & 0 & 1 \end{bmatrix} \tag{5.4}$$

Estrategia de GenParTAso

Del mismo modo, el primer paso es la selección aleatoria de una imagen, posteriormente se aplica una sola vez la composición de transformaciones. De esta manera se obtiene una

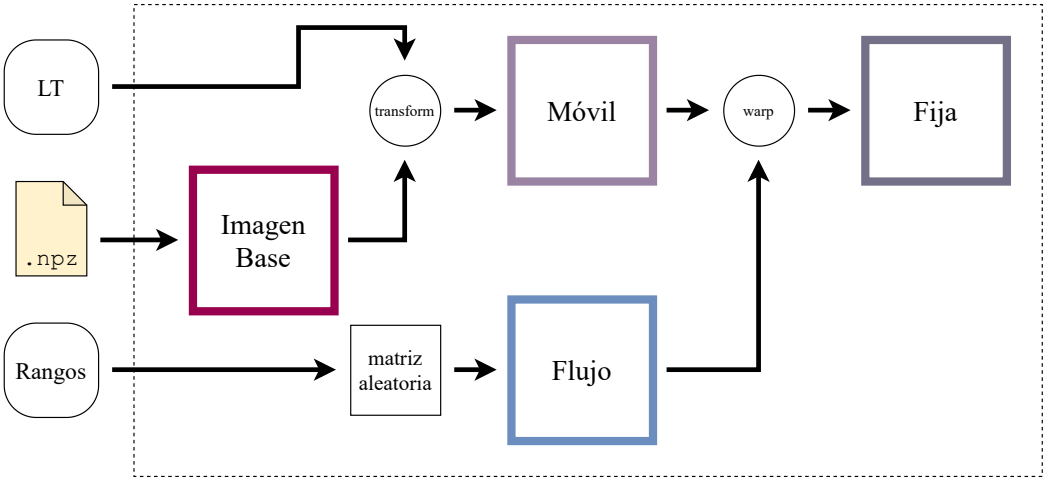


Figura 5.6: Esquema de la estrategia de generación GenParTAso.

imagen del par, la segunda se calcula a partir de la primera mediante unos valores (θ, x, y) aleatorios. Estos valores están dentro de unos rangos definidos en parámetros adicionales. En la Figura 5.6 está el esquema de esta estrategia. El cálculo de la matriz de transformación (M) se puede ver en la Ecuación 5.5, donde R es la matriz de rotación, T es la matriz de translación. A diferencia de la anterior estrategia, en este caso es un cálculo directo.

$$M = R \times T, \text{ donde } R = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \text{ y } T = \begin{bmatrix} 1 & 0 & x \\ 0 & 1 & y \\ 0 & 0 & 1 \end{bmatrix} \tag{5.5}$$

Cálculo del campo de desplazamiento sintético

Con la matriz de transformación calculada en cualquiera de los dos estrategias de generación explicadas, se calcula el campo de desplazamiento correspondiente mediante la función `affine_grid`[4] de PyTorch. Esta función espera recibir un parámetro ***theta***, que será la matriz de transformación de $N \times 2 \times 3$ (siendo N el número de matrices, que en este caso es 1), y un ***size***, que será el tamaño de salida objetivo de la imagen, especificado con $N \times C \times H \times W$ (siendo N el número de imágenes, C el número de canales, H y W el alto y ancho de la imagen correspondientemente, que en este caso será $1 \times 1 \times 256 \times 256$).

Por lo que a la matriz de transformación M se le quita la última fila, para que quede una matriz 2×3. Está matriz representa una transformación afín 2D (es decir, 6 parámetros) compuesta por una translación y una rotación.

[4] https://pytorch.org/docs/1.5.1/nn.functional.html#affine-grid

5.4.3 Entrenador

Este componente se encarga de gestionar el entrenamiento del modelo con el generador seleccionado. En este trabajo el modelo que se usa como base es el implementado en Voxelmorph[5]. A continuación se explicarán los conceptos básicos de este modelo, y se hará una definición de las funciones de pérdida utilizadas en el entrenamiento.

Modelo base: Voxelmorph

En realidad Voxelmorph no es un modelo en sí mismo, sino que es un *framework* basado en aprendizaje para la tarea de registro de imagen deformable [23]. Sin embargo, los autores proporcionan una implementación de un modelo en Voxelmorph que es el modelo que se utiliza en este trabajo como base.

Método. El método propuesto por Balakrishnan et al. [23] consiste en entrenar una red que tiene como entrada dos imágenes concatenadas, la fija y la móvil. La red inferirá, a partir de ellos, un flujo de desplazamiento que relacione una con la otra. Ellos aplican este método para hacer registro deformable sobre imágenes del cerebro, con la condición de que el par de entrada tiene que estar previamente alineado.

Arquitectura. La arquitectura del modelo presentado, consiste en una red similar a una U-Net para la obtención del flujo de desplazamiento y de una *Spatial Transformer Layer* (STL), que aplica el flujo previamente inferido sobre la imagen móvil para obtener la imagen movida.

- **U-Net**: Es una red introducida por Ronneberger et al. [33], se trata de una CNN con el objetivo de realizar la segmentación de imágenes biomédicas. En [23], se utiliza para inferir los parámetros del flujo de desplazamiento. En la Figura 5.7a aparece representada la U-Net original, y en la Figura 5.7b la versión que se usa en Voxelmorph.

- **STL**: Esta basado en la *Spatial Transformer Network* (STN) [27], este módulo se compone de tres partes diferenciadas: la *localisation net.* el *grid generator* y el *sampler.* A través de un proceso de regresión, con la *localisation net* se obtienen los parámetros de la transformación que se busca, con el cual se genera un grid y se aplica a la imagen mediante el *sampler.* En el modelo de Voxelmorph, la U-Net actuaría como la *localisation net* y el STL sería la parte del *grid generator* y el *sampler.*

Debido a la forma de implementar un componente utilizado para la construcción de la STN en PyTorch[6], donde las localizaciones de los píxeles se representan normalizados en un rango

[5] http://voxelmorph.csail.mit.edu
[6] https://pytorch.org/docs/1.5.1/nn.functional.html#torch.nn.functional.grid_sample

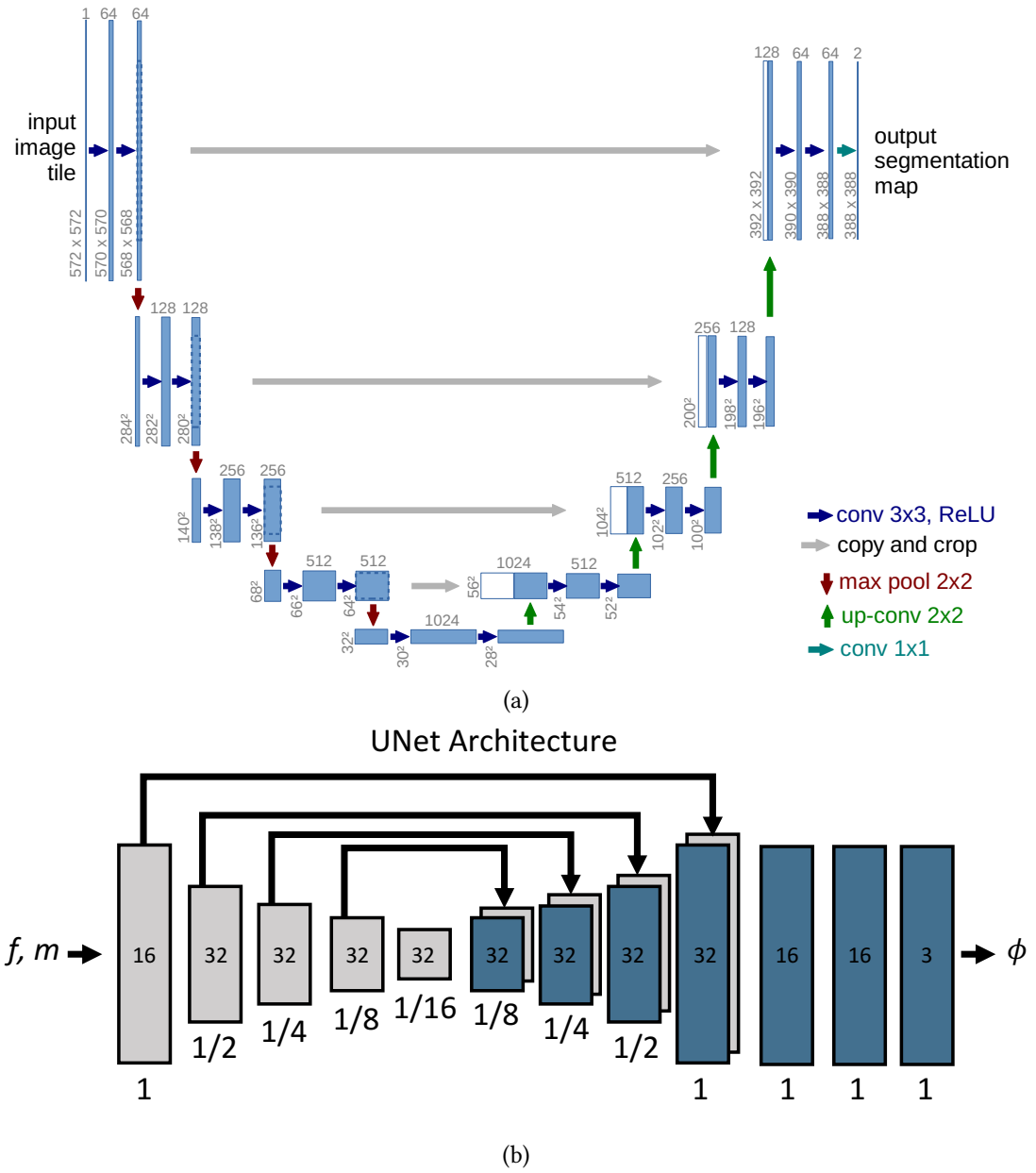


Figura 5.7: U-Net diagrama original [33] (a) y versión de Voxelmorph [23] (b).

de $[-1, 1]$, hace que en la STL de Voxelmorph se realicen una serie de operaciones previas. Por lo tanto en el entrenamiento supervisado el flujo con el que se compara tiene que ser de la forma a cómo estaría antes de estas operaciones. Por lo que en el caso de los experimentos supervisados que se verán a continuación, al flujo de desplazamiento calculado en Cálculo del campo de desplazamiento sintético (pág. 30) se le aplica el proceso inverso de esta serie de operaciones.

Definición de métricas del sistema

En este módulo se utilizan tres métricas para evaluar el rendimiento de la red, que son: el Error Cuadrático Medio (o *Mean Squared Error* (MSE)), el coeficiente de Dice (Dice) [34] y el Índice de Similitud Estructural (o *Structural Similarity Index* (SSIM)) [35]. La primera se usa durante el entrenamiento para la optimización de pesos de la red, y las dos últimas se usan en la evaluación de los resultados de registro.

- MSE: Es una función que calcula la media de las diferencias cuadradas. En este trabajo se utiliza para estimar la similaridad entre la imagen fija y la imagen movida, también se usa para estimar el error cometido en el flujo devuelto por la red con respecto al real.

- Dice: Se utiliza en la medición de solapamiento entre la imagen fija e imagen movida. Es usado en los casos dónde ambas imágenes son binarias, es decir se utiliza en las imágenes de los vasos sanguíneos segmentados de DRIVE. Se calcula con la siguiente ecuación:

$$\text{Dice}(f, m \circ \hat{\Phi}) = \frac{2 \cdot |f \cap (m \circ \hat{\Phi})|}{|f| + |(m \circ \hat{\Phi})|}, \tag{5.6}$$

  donde f y m son las imágenes fija y móvil, $\hat{\Phi}$ es el flujo que infiere la red, y $m \circ \hat{\Phi}$ es la imagen movida. En el numerador aparece el número de píxeles de la intersección entre la imagen fija y la imagen movida, en el denominador se refleja el número total de píxeles.

- SSIM: Es un método para medir la similaridad entre dos imágenes de forma que se separa la información estructural de la intensidad y contraste. Es usado en los casos dónde la aplicación de Dice no es posible, es decir en las angiografías. La implementación utilizada es la que se encuentra en la librería de `pytorch-msssim`[7].

Definición de funciones de pérdida

Según la forma de entrenar que tenga la red, la función de pérdida que se utiliza para optimizarla durante el mismo es diferente. A continuación se enumeran las dos maneras de

[7] https://github.com/VainF/pytorch-msssim

entrenar que se plantean en el trabajo:

- **Entrenamiento no supervisado**: La función de pérdida que se utiliza en este tipo de entrenamiento es el mismo que proponen en [23]. Que se define con la siguiente ecuación:

$$\mathcal{L}_{unsup}(f, m, \hat{\Phi}) = \mathcal{L}_{sim}(f, m \circ \hat{\Phi}) + \lambda \mathcal{L}_{smooth}(\hat{\Phi}), \tag{5.7}$$

  donde λ es el término regularizador. La métrica que se usa en $\mathcal{L}_{sim}$ es MSE, y el $\mathcal{L}_{smooth}$ es un regularizador L1.

- **Entrenamiento supervisado**: Con la generación sintética de pares, el flujo real de desplazamiento del par es calculable y por tanto se puede usar para guiar el entrenamiento. La función de pérdida para este caso es:

$$\mathcal{L}_{sup}(\Phi, \hat{\Phi}) = \mathcal{L}_{sim}(\Phi, \hat{\Phi}), \tag{5.8}$$

  donde Φ es el flujo sintético. La métrica que se usa en $\mathcal{L}_{sim}$ es MSE.

5.5 Módulo de reporte

5.5.1 Esquema del módulo

El módulo de reporte aparece representada en la Figura 5.8, en el se ve el Evaluador, que prácticamente es idéntico al Entrenador del anterior módulo, salvo que en este caso el modelo está en el modo de evaluación y por tanto no se actualizan sus parámetros. También se observa el *Analyzer* y el *Visualization* que básicamente contienen diferentes utilidades para la lectura del fichero `.npz` generado por el Evaluador y para la visualización de los resultados de registro, así como funciones para mostrar gráficas a partir de las métricas.

A continuación, siguen unos apartados explicando diferentes métricas específicas del trabajo realizado.

5.5.2 Error cometido en el flujo de desplazamiento

El Error en el Flujo (EF) que comete la red, es una medición usada para evaluar cómo de buenos son los flujos inferidos por el modelo. En este trabajo se utiliza la misma métrica que cuando se calcula la función de pérdida en el entrenamiento supervisado. A continuación se presenta la ecuación:

$$EF(\hat{\Phi}) = EF(\Phi, \hat{\Phi}) = MSE(\Phi, \hat{\Phi}), \tag{5.9}$$

donde Φ es el flujo objetivo y $\hat{\Phi}$ es el flujo obtenido de la red.

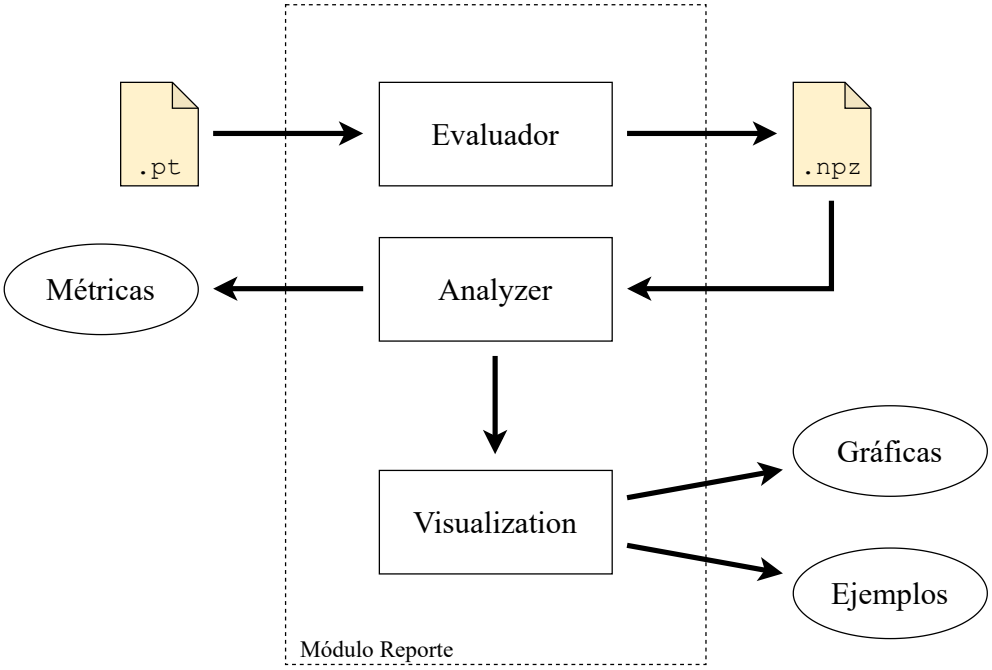


Figura 5.8: Esquema del módulo de reporte.

5.5.3 Tasa de éxito

La Tasa de Éxito (TE), es una métrica que se utiliza en el trabajo para medir el rendimiento del sistema. Esta tasa se calcula dividiendo el número de casos considerados como exitosos sobre el número total, N, de ejemplos de registro. Se establece en este trabajo, que un caso exitoso es aquel en el que el error cometido en la predicción del flujo, EF, es menor a 1. La TE queda definida de la siguiente manera:

$$TE = \frac{1}{N}\sum_{i=1}^{N} S(i), \text{ donde } S(i) = \begin{cases} 1, \text{si } EF(\hat{\Phi}_i) < 1. \\ 0, \text{en otro caso.} \end{cases} \tag{5.10}$$

5.5.4 Diferencia en el flujo de desplazamiento

Para comparar el rendimiento de diferentes configuraciones de modelos que realizan el registro sobre una misma modalidad de imagen, se define la diferencia entre el error cometido en el flujo por los modelos a comparar como:

$$\Delta(\hat{\Phi}_{m1}, \hat{\Phi}_{m2}) = EF(\hat{\Phi}_{m2}) - EF(\hat{\Phi}_{m1}), \tag{5.11}$$

donde $m1$ es el modelo sobre el que se quiere comparar el modelo $m2$. Teniendo en cuenta esto, un resultado negativo indica que $m2$ es mejor que $m1$ y, viceversa cuando el resultado

es positivo.

Capítulo 6

Experimentos y resultados

En este capítulo se explican los diferentes experimentos que se han realizado durante el transcurso de este trabajo así como la discusión sobre los resultados obtenidos.

6.1 Definiciones generales

En esta sección se definen algunos conceptos generales que tienen que ver con el entrenamiento de redes neuronales.

- **Batch Size**: Es el número de muestras que se utilizan en una iteración de entrenamiento.
- **Epoch**: Por definición, es una pasada completa del conjunto de datos de entrenamiento. Es un número que indica la cantidad de veces que se recorre todo el conjunto de datos en un entrenamiento.
- **Steps per Epoch**: Es el número de pasos que tiene un epoch, se calcula en función del batch size y el tamaño del conjunto de entrenamiento.
- **Learning Rate**: Es el tamaño del paso que realiza el optimizador en cada iteración.

En este trabajo, como el conjunto de datos de entrenamiento no es estático, los valores que toma el epoch y steps per epoch es meramente un indicativo para saber la cantidad de datos que procesa la red durante su entrenamiento.

6.2 Tipos de generadores

En esta sección se clasifican los tipos de generadores que se han usado en los diferentes experimentos. Se categoriza según la estrategia de generación empleada y según el resultado que genera.

6.2.1 Por estrategia de generación

En este trabajo se presentan dos estrategias de generación de pares, explicados en Estrategias de generación (pág. 27). Para los experimentos, se propone un generador que sigue la estrategia de GenParTInd, que se llamará Gti, y otro generador que sigue la estrategia de GenParTAso, que se llamará Gta.

Para Gta se proponen tres variantes:

- Gta Base (GtaBa): Generador con el rango de transformación base utilizado en los experimentos.
- Gta Ampliado (GtaAm): Generador con el rango de transformación ampliado.
- Gta Exclusivo (GtaEx): Generador con el rango de transformación, únicamente, en la parte extendida.

Los parámetros del rango de transformación de estos generadores se definen en la Tabla 6.1. En la que se puede apreciar como el generador GtaBa se compone de transformaciones livianas, mientras que GtaAm tiene el rango ampliado, y siendo GtaEx la diferencia de ambas, en la que se deja el rango ampliado sin incluir el rango de GtaBa.

Generador	Rango de transformación
GtaBa	$[-5, 5]$
GtaAm	$[-10, 10]$
GtaEx	$[-10, -5] \cup [5, 10]$

Tabla 6.1: Variantes de generadores implementando la estrategia GenParTAso.

Cabe mencionar, que Gti, GtaBa y GtaAm se aplican tanto en el entrenamiento como en el test de los modelos, mientras que GtaEx se usa solamente en el test.

6.2.2 Por resultado generado

El resultado que devuelve el generador varía en función de la modalidad de imagen tratada o de variantes de la misma, así pues aparecen dos tipologías de generadores. A continuación se describe cada una de ellas. y en la Figura 6.1 aparecen de forma esquemática.

- **Generador simple**: Este tipo de generador sintetiza una imagen móvil y una imagen fija, junto al flujo de transformación que relaciona la móvil con la fija.
- **Generador compuesto**: En esta tipología se realiza una generación compuesta, esto quiere decir que se sintetizan dos imágenes móviles y dos imágenes fijas, y un único flujo de transformación que asocia las imágenes móviles con sus correspondientes imágenes

fijas. Se obtienen dos pares que se pueden usar para el registro de imagen, más adelante se verá que uno de los pares se suministra como entrada al modelo, para la inferencia de flujo, mientras que el otro par se utiliza en el cálculo de las métricas del modelo.

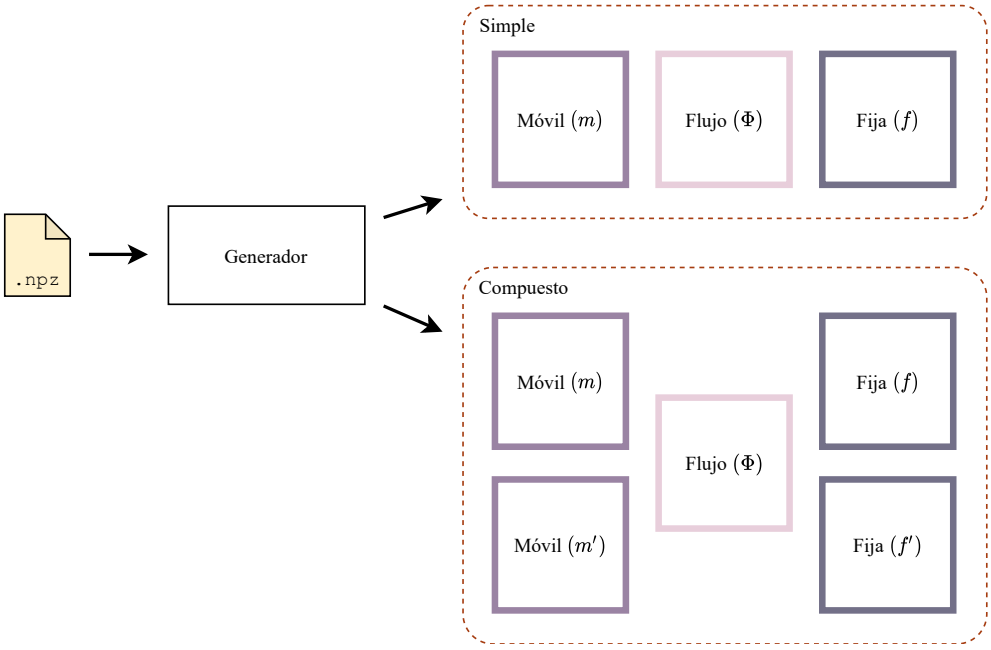


Figura 6.1: Tipos de generadores por resultado generado.

Resultado generado según la modalidad de imagen

Para ejemplificar el concepto de generador simple y compuesto se procede a enumerar el tipo de generador que es según la modalidad de la imagen utilizada.

- **Modalidad *vessels***: Lo habitual será tener un generador simple. Aunque hay un caso concreto, en el que será un generador compuesto (véase experimento de la Sección 6.5.3 (pág. 47)).
- **Modalidad *retinal***: En este caso, el generador será compuesto. La salida será el par de retinografías y el par, respectivamente, de los vasos sanguíneos segmentados.
- **Modalidad *angiography***: El generador es simple.

6.3 Esquema experimental

Los experimentos realizados en el trabajo se clasifican en dos grupos principales: no supervisados y supervisados. A continuación, se explican de manera general las características

de estos experimentos.

6.3.1 Experimentos no supervisados

En los experimentos no supervisados, el modelo se entrena optimizando la función de pérdida no supervisada, definida en la Ecuación 5.7. Estos experimentos siguen la misma estructura básica que en [23]. Se busca que la red maximice la similaridad entre la imagen fija y la imagen móvil, sin poner más restricciones al flujo a parte del regularizador.

6.3.2 Experimentos supervisados

Los experimentos supervisados, son aquellos en los que el modelo se entrena optimizando la función de pérdida supervisada, definida en la Ecuación 5.8. El objetivo es guiar al modelo hacia la generación de un flujo afín.

6.3.3 Esquema general de los experimentos

En la Figura 6.2 está representada de forma general el esquema de los experimentos, se puede observar el cambio del comportamiento cuando se usan generadores simples y cuando se usan generadores compuestos. Como se puede ver en el caso de los entrenamientos supervisados no hay distinción en si es simple o compuesto.

6.4 Propuesta experimental

Para dar respuesta a la pregunta planteada en la Sección 1.2 (pág. 2), se realiza la siguiente propuesta experimental.

En primer lugar, para comprobar que un modelo de registro de imagen basado en DL puede llevar a cabo el registro de imágenes oftalmológicas, se plantean experimentos sobre distintas modalidades de dichas imágenes. Se ha elegido la modalidad de imagen de los vasos sanguíneos segmentados como los datos que se van a usar para realizar diferentes experimentos, con el fin de conseguir un modelo capaz de registrar estas imágenes. Después, se emplea la configuración experimental que se haya encontrado, en los experimentos anteriores, para entrenar de nuevo el modelo sobre otras modalidades de imagen. Finalmente, con los resultados obtenidos, se intenta observar el funcionamiento de una estrategia de entrenamiento incremental para ver si estos modelos pueden ampliar su capacidad de registro.

A continuación, se enumeran estas propuestas:

- **Registro de vasos sanguíneos**: El objetivo de este planteamiento experimental es conseguir un modelo que realice el registro de imágenes de vasos sanguíneos.

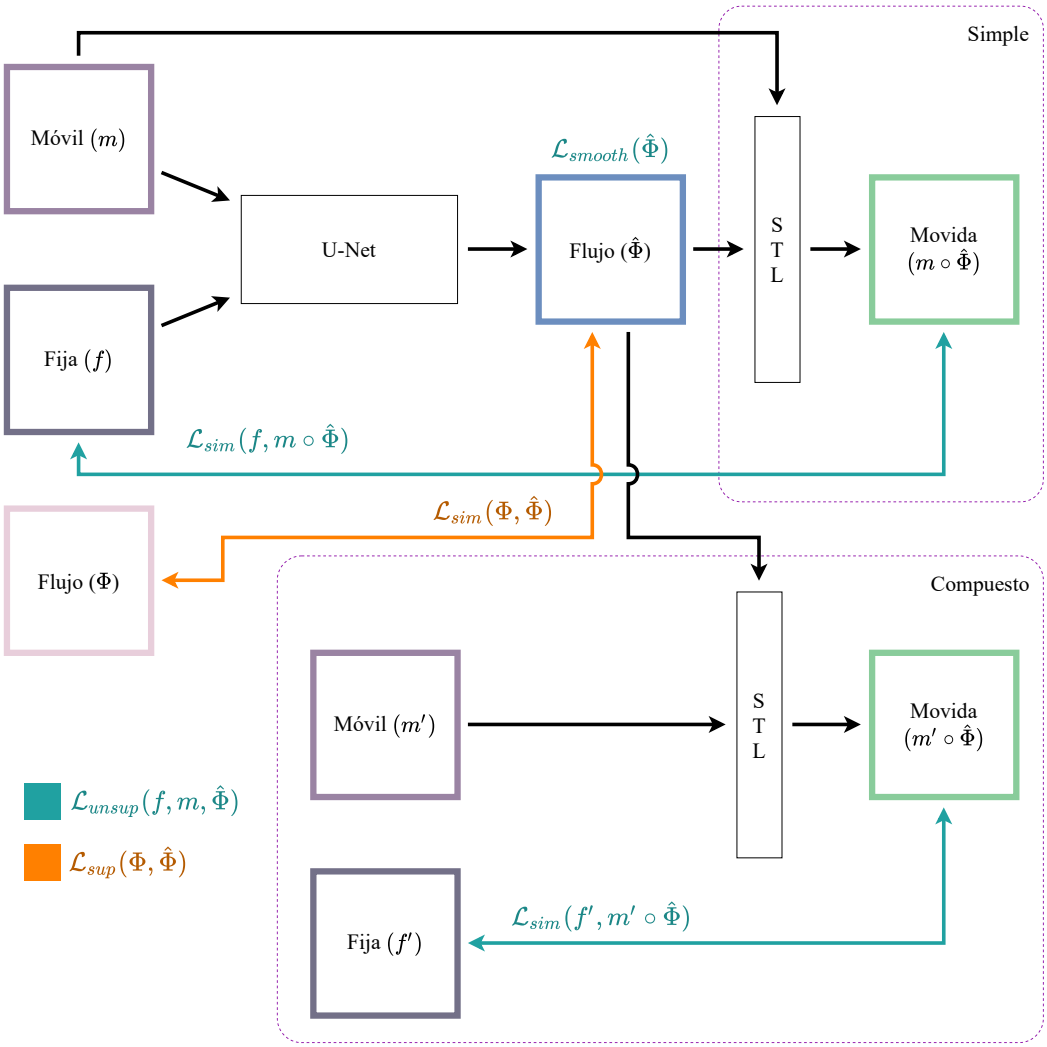


Figura 6.2: Esquema genérico del entrenamiento de los experimentos.

- **Eficacia usando otras modalidades de imagen**: Partiendo de las características del modelo logrado en el experimento anterior, se propone volver a realizar el entrenamiento del modelo, pero utilizando retinografías y angiografías.

- **Extensión del rango de transformación**: En este grupo de experimentos el objetivo es ver la efectividad que tiene usar una estrategia de entrenamiento incremental, que consiste en ir aumentando el rango de las transformaciones de los pares de entrada.

En las próximas secciones se explican con más detalle cada una de estas propuestas y sus experimentos.

Identificador de los experimentos

Antes de continuar con el desarrollo de cada experimento, se va a ver cómo es el identificador de estos experimentos. Este identificador consta de cuatro partes, que se enumeran a continuación:

1. **Tipo de entrenamiento**: La primera parte sirve para diferenciar el tipo de entrenamiento que se lleva a cabo en el experimento. Se disponen de dos opciones:
   - U: Entrenamiento no supervisado.
   - S: Entrenamiento supervisado.
2. **Modalidad de imagen**: Esta parte identifica la modalidad de imagen utilizada en el experimento. En este caso, hay tres opciones:
   - V: Modalidad de *vessels*.
   - R: Modalidad de *retinal*.
   - A: Modalidad de *angiography*.
3. **Generador**: La tercera parte indica el generador que se ha utilizado en el experimento. Las opciones que disponibles son:
   - ti: Generador Gti.
   - taBa: Generador GtaBa.
   - taAm: Generador GtaAm.
4. **Información adicional**: La cuarta y última parte es opcional, y sirve para incluir información adicional del experimento. En este trabajo se han utilizado las siguientes opciones:
   - dt: Indica que se ha utilizado la operación de *Signed Distance Transform* (SDT).
   - rf: Indica que se ha aplicado refinamiento.

El identificador del experimento se va a utilizar indistintamente para hacer referencia al propio experimento, como para referirse al modelo obtenido del mismo.

6.5 Registro de vasos sanguíneos

Este planteamiento experimental tiene como objetivo conseguir un modelo entrenado para registro de imagen de los vasos sanguíneos segmentados del ojo, que provienen del *dataset* de DRIVE. Esto significa que el modelo tiene que ser capaz de registrar nuevos pares de vasos sanguíneos, generando un flujo acorde a la transformación aplicada.

Listado de experimentos

Se han realizado un total de cuatro experimentos, hasta conseguir un modelo aceptable que cumple con el objetivo planteado. A continuación, se enumeran y se describen brevemente cada una de ellas:

- **Registro no supervisado de *vessels* con Gti (UVti)**: Es el punto de partida de los experimentos, se utiliza el esquema experimental presentado en [23]. El generador sigue la estrategia del GenParTInd, es decir, la transformación que vincula los pares de imágenes son laxas.

- **Registro no supervisado de *vessels* con GtaBa (UVtaBa)**: En este experimento se cambia el tipo de estrategia en el generador, usando ahora el GenParTAso. El motivo de este cambio es la reducción de la complejidad del problema, para intentar conseguir un modelo funcional.

- **Registro no supervisado de *vessels* con GtaBa aplicando SDT (UVtaBa-dt)**: Se introduce una operación adicional, la *Signed Distance Transform* (SDT), con el objetivo de expandir el área que contribuye en la inferencia del flujo de desplazamiento.

- **Registro supervisado de *vessels* con GtaBa (SVtaBa)**: Este experimento representa un cambio en el planteamiento. Ahora el entrenamiento es supervisado, con el objetivo de hacer que la red devuelva un flujo lo más parecido al sintético. La supervisión viene dada al guiar el modelo mediante la optimización de la similitud del flujo inferido con el flujo sintético asociado al par de entrada.

Evaluación cualitativa

En los siguientes experimentos, se evalúa de forma cualitativa de los resultados mediante tres ejemplos de pares de registro, cuyas características se resumen en la Tabla 6.2.

Ejemplo	Generador	Base	Móvil	Fija
1	Gti	DRIVE n.º 17	(5.88, 27, 26)	(-5.47, 121, 124)
2	GtaBa	DRIVE n.º 17	(-33.51, 127, 83)	(0.17, -1.64, 0.25)
3	GtaBa	DRIVE n.º 12	(10.87, 123, 24)	(4.46, -3.37, 2.35)

Tabla 6.2: Características de los ejemplos de registro en el grupo de experimentos de registro de vasos sanguíneos. La tupla representa el (θ, x, y) de la imagen, que en la imagen móvil es el ángulo de rotación y la esquina del recorte. En la imagen fija, para el ejemplo 1 tiene el mismo significado que la móvil, pero para los otros ejemplos, es el ángulo y el vector de desplazamiento aplicado sobre la imagen móvil.

Evaluación cuantitativa

En los siguientes experimentos, la evaluación cuantitativa se realiza sobre el conjunto sintetizado de 5000 pares, usando como base las imágenes del conjunto de test. El generador que se utiliza en esta evaluación es el mismo que el utilizado durante el entrenamiento del respectivo experimento. En las tablas se muestra el coeficiente de Dice (Dice), mostrando la media y entre paréntesis la desviación típica, y la Tasa de Éxito (TE) definida en la Ecuación 5.10.

6.5.1 Registro no supervisado de *vessels* con Gti

El identificador de este experimento es UVti, el generador es simple e implementa la estrategia GenParTInd, es decir, el generador es Gti.

Configuración experimental

En la Tabla 6.3 aparecen los valores de los diferentes hiperparámetros utilizados en el entrenamiento de este experimento. El λ es del valor que toma en la Ecuación 5.7. Se ha entrenado con un batch size de 32, durante 20 epochs con 100 steps cada uno, eso quiere decir que la red a observado un total de 64000 ejemplos de registro.

Learning Rate	Batch Size	Epochs	Steps per Epoch	Regularizador (λ)
1e-2	32	20	100	0.05

Tabla 6.3: Hiperparámetros del registro no supervisado de *vessels* con Gti

Resultados

En la Tabla 6.4 se puede ver el resultado del experimento, se puede observar que el Dice es bastante elevado, mientras que la TE es nula. La red consigue alinear el contenido pero no de la manera que se espera.

Dice	TE
0.9072 ($\pm$ 0.0175)	0.00%

Tabla 6.4: Resultados cuantitativos del registro no supervisado de *vessels* con Gti.

En la Figura 6.3 se muestran ejemplos del resultado de registro empleando esta red, en las columnas exteriores está el ejemplo Antes de Registro (AR) y el flujo real (Φ), y en la parte central el solapamiento después del registro (DR) y el flujo inferido por la red ($\hat{\Phi}$). En color verde se representa la imagen fija, en rojo la imagen móvil en el caso previo al registro y en el caso de después del registro representa la imagen movida, por último en color amarillo está representada la parte solapada de ambas imágenes. Visualmente parece que el registro tiene

éxito, pues el solapamiento que se obtiene es bastante bueno, sin embargo al mirar el flujo obtenido se puede ver que en definitiva no es lo que se esperaba. En la Tabla 6.5 aparecen las métricas de estos ejemplos. Se puede observar en la misma, que el ejemplo 2 cumpliría con el criterio para ser considerada como caso exitoso, sin embargo comparando de manera visual el flujo inferido con el real se ve que no es parecido.

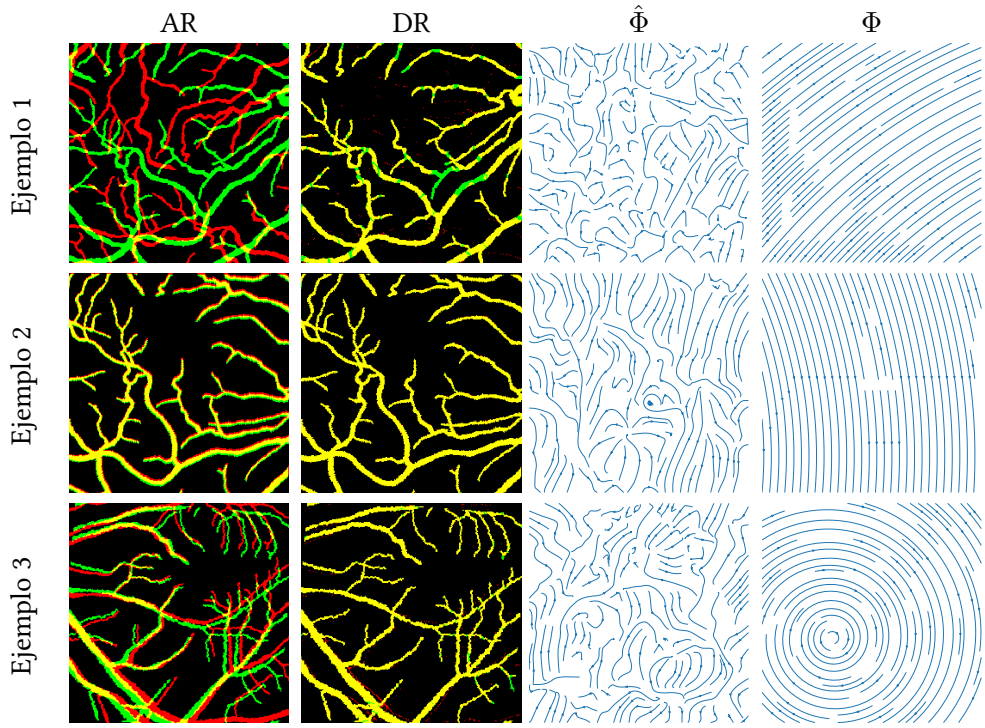


Figura 6.3: Ejemplo de registro con modelo no supervisado con Gti.

Ejemplo	Dice AR	Dice	$EF(\hat{\Phi})$
1	0.1846	0.9207	9315.4882
2	0.7131	0.9927	0.9712
3	0.3865	0.9630	37.1210

Tabla 6.5: Métricas de los ejemplos de la Figura 6.3.

Conclusiones

En vista de estos resultados, se interpreta que la red está aportando soluciones de transformación que, consiguiendo hacer encajar el contenido de las imágenes fija y móvil, no son realistas con respecto a la transformación afín aplicada.

Para minimizar la amplitud de las transformaciones que tiene que aprender la red, se

cambia la estrategia de generación por la GenParTAso. De esta forma se simplifica el problema, al hacer que las transformaciones sean más livianas.

6.5.2 Registro no supervisado de *vessels* con GtaBa

El identificador de este experimento es UVtaBa, el generador es simple e implementa la estrategia GenParTAso con la variante básica, es decir, el generador es GtaBa.

Configuración experimental

Los hiperparámetros utilizados en este entrenamiento están en la Tabla 6.6. La red ha observado un total de 4000 pares.

Learning Rate	Batch Size	Epochs	Steps per Epoch	Regularizador (λ)
1e-3	8	50	10	0.01

Tabla 6.6: Hiperparámetros del registro no supervisado de *vessels* con GtaBa.

Resultados

Las métricas del experimento se sitúan en la Tabla 6.7 en la que se puede apreciar que la TE aumenta ligeramente. Hay que tener en cuenta que las métricas obtenidas son de dos conjuntos diferentes (en comparación al anterior experimento, pues se ha cambiado el generador). Al analizar los casos en los que se determina como exitoso, se observa que la transformación a inferir es casi imperceptible, haciendo que el cálculo del MSE entre flujos sea menor a 1.

Dice	TE
0.8791 ($\pm$ 0.0312)	0.62%

Tabla 6.7: Resultados cuantitativos del registro no supervisado de *vessels* con GtaBa.

En la Figura 6.4, aparece el caso comentado anteriormente en el ejemplo 2, se puede ver que el flujo es algo similar al sintetizado, debido a la casi inexistente transformación. En la Tabla 6.8 se muestran las métricas de estos ejemplos.

Ejemplo	Dice AR	Dice	$EF(\hat{\Phi})$
1	0.1846	0.7659	9216.5859
2	0.7131	0.9487	0.6050
3	0.3865	0.8864	37.3237

Tabla 6.8: Métricas de los ejemplos de la Figura 6.4.

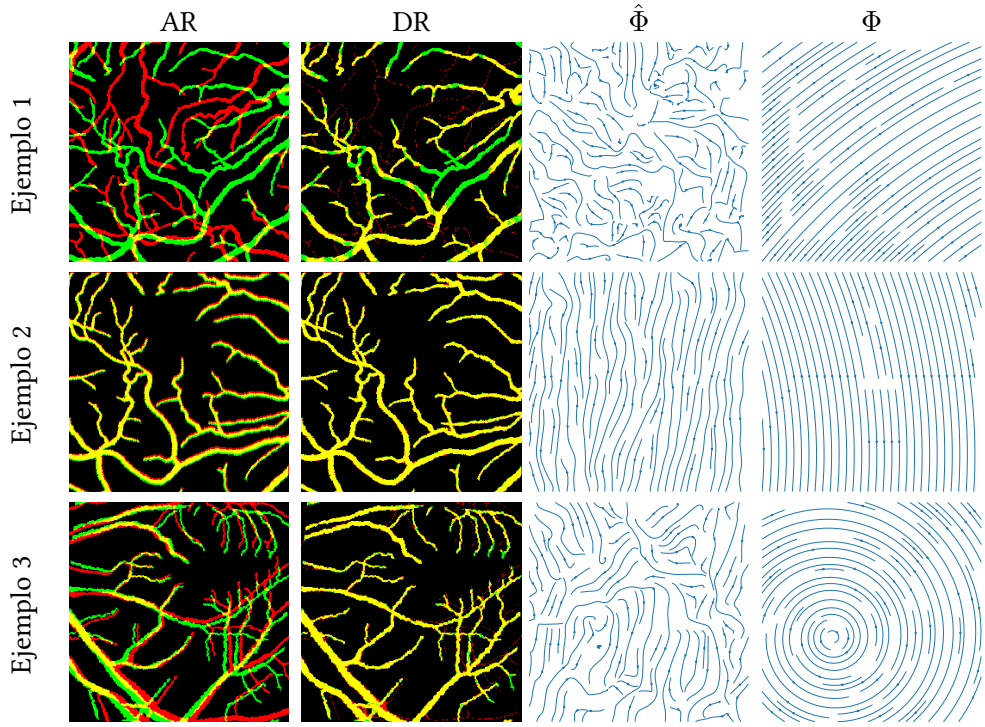


Figura 6.4: Ejemplo de registro con modelo no supervisado con GtaBa.

Conclusiones

La reducción de la amplitud de las transformaciones aplicadas a la hora de sintetizar pares, parece ayudar en hacer que la red devuelva flujos más similares el deseado, como se ha podido observar en el ejemplo 2 al compararlo con el primer experimento.

Sin embargo, parece no ser suficiente, un motivo podría ser que la cantidad de píxeles útil para inferir el flujo es pequeña, ya que la mayor parte de la superficie de la imagen es fondo (i.e. píxeles sin información, con valor 0).

6.5.3 Registro no supervisado de *vessels* con GtaBa aplicando SDT

El identificador de este experimento es UVtaBa-dt, el generador es compuesto e implementa la estrategia GenParTAso con la variante básica, es decir, el generador es GtaBa. Adicionalmente, se aplica la operación de *Signed Distance Transform* (SDT).

Configuración experimental

Las imágenes de los vasos sanguíneos en la mayor parte de su superficie es fondo, es decir partes que no contribuyen en la inferencia del flujo desplazamiento. Por lo tanto se plantea realizar una variante del generador empleado en el experimento anterior, añadiendo

una operación de SDT para expandir la superficie con información. Esta operación consiste en que para cada píxel le asigna un valor que es la distancia entre ese píxel y el píxel más cercano cuyo valor es distinto a 0, el signo indica si está por dentro o por fuera. Adicionalmente se añade un rango de saturación de -20 a 20, en la que si la distancia supera esos valores se sustituyen por el mínimo o máximo establecido en el rango. La red es entrenada con las imágenes obtenidas tras la SDT, pero la función de pérdida se aplica sobre la fija binaria y la movida de aplicar el flujo obtenido sobre la móvil binaria.

Los hiperparámetros del entrenamiento son los mismos que el experimento anterior, que se pueden ver en la Tabla 6.6 y por tanto la cantidad de pares observados por la red también es igual, es decir 4000 pares.

Resultados

En la Tabla 6.9 aparecen los resultados del experimento, se observa que la TE vuelve a aumentar en comparación con el anterior experimento, indicando que esta expansión de la superficie con información de la imagen puede haber contribuido en una mejor inferencia de flujos.

Dice	TE
0.8085 ($\pm$ 0.0502)	1.30%

Tabla 6.9: Resultados cuantitativos del registro no supervisado de *vessels* con GtaBa aplicando SDT.

En la Figura 6.5 están los ejemplos de registro utilizando este modelo, y en la Tabla 6.10 sus métricas.

Ejemplo	Dice AR	Dice	$EF(\hat{\Phi})$
1	0.1846	0.7247	9381.7441
2	0.7131	0.9260	0.4630
3	0.3865	0.8020	30.7356

Tabla 6.10: Métricas de los ejemplos de la Figura 6.5.

Conclusiones

En el ejemplo 3 se puede ver como en los extremos de la imagen aparecen artefactos que pueden ser debidos a que el flujo se obtiene por las transformadas de distancia, es decir que la expansión de la información ayuda a generar flujos con menos deformaciones pero también puede hacer que aparezcan artefactos en los extremos por el mismo motivo.

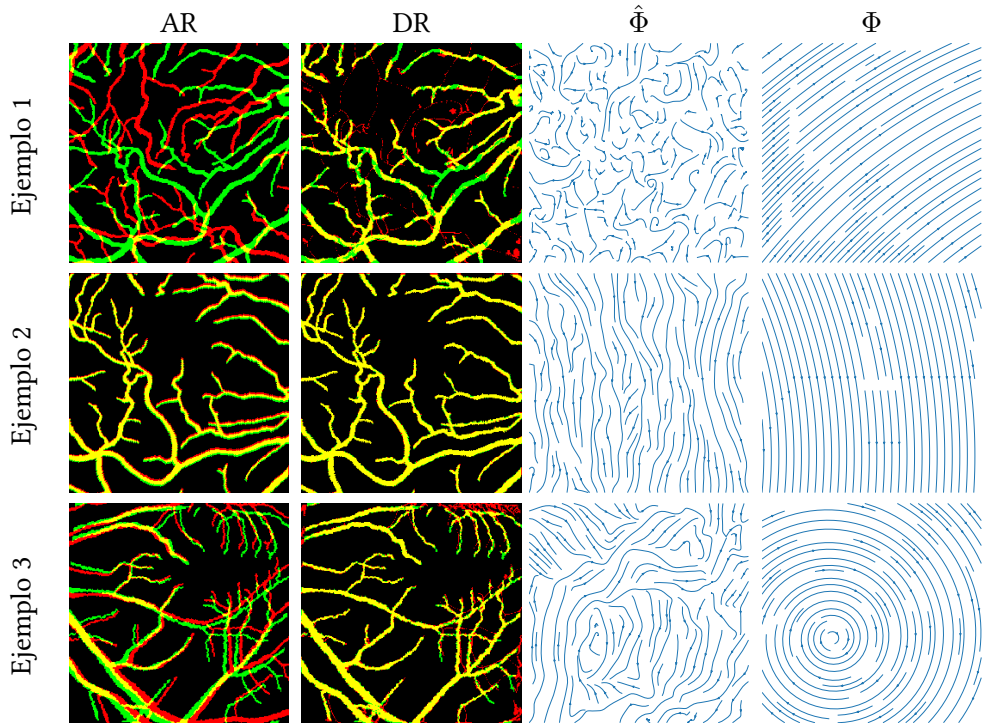


Figura 6.5: Ejemplo de registro con modelo no supervisado con GtaBa aplicando SDT.

6.5.4 Registro supervisado de *vessels* con GtaBa

El identificador de este experimento es SVtaBa, el generador es simple e implementa la estrategia GenParTAso con la variante básica, es decir, el generador es GtaBa.

Configuración experimental

Los resultados de los experimentos no supervisados no son muy buenos en cuanto a la calidad del flujo de desplazamiento que se obtiene. Una alternativa para que la red entrenada genere mejores flujos es guiar su entrenamiento de una manera supervisada, sin necesidad de cambiar la arquitectura del modelo, ya que otra opción sería adaptar la arquitectura para que devuelva los parámetros que conforman una transformación afín.

En la Tabla 6.11 está la relación de los hiperparámetros utilizados en el entrenamiento, la red vio 384000 pares.

Learning Rate	Batch Size	Epochs	Steps per Epoch
3e-4	64	200	30

Tabla 6.11: Hiperparámetros de experimentos supervisados.

Resultados

La Tabla 6.12 muestra los resultados del experimento, como se puede ver en el mismo, la media obtenida de Dice es muy similar al homólogo no supervisado, sin embargo la TE es muy superior, alcanzando casi un 90%.

Dice	TE
0.8776 ($\pm$ 0.0389)	88.06%

Tabla 6.12: Resultados cuantitativos del registro supervisado de *vessels* con GtaBa.

En los ejemplos de registro, que se pueden ver en la Figura 6.6, se puede observar como el flujo obtenido por la red es bastante más similar al esperado. Aunque en el ejemplo 1 se ve que para casos muy diferentes a los vistos en el entrenamiento la red lleva a cabo el registro bastante mal. En la Tabla 6.13 están las métricas de estos ejemplos.

Ejemplo	Dice AR	Dice	$EF(\hat{\Phi})$
1	0.1846	0.2013	9793.7509
2	0.7131	0.9517	0.1833
3	0.3865	0.8486	0.8925

Tabla 6.13: Métricas de los ejemplos de la Figura 6.6.

Conclusiones

Parece que con un entrenamiento supervisado, la red consigue generar buenos flujos de desplazamiento, siempre y cuando los nuevos pares que se quieran registrar tengan transformaciones dentro del rango entrenado o muy similar.

6.5.5 Conclusiones generales

En los experimentos que se han realizado, se ha podido ver que al entrenar de forma no supervisada, la red consigue buenos resultados en cuanto al alineamiento del contenido de las imágenes. Sin embargo, el flujo que infiere la red no son realistas con respecto a la transformación afín que se le aplica.

Al cambiar el planteamiento a un modelo supervisado, se logra conseguir un modelo que aporta buenas soluciones, tanto en alineamiento del contenido como en el realismo de las transformaciones (se asemejan más a la transformación afín aplicada).

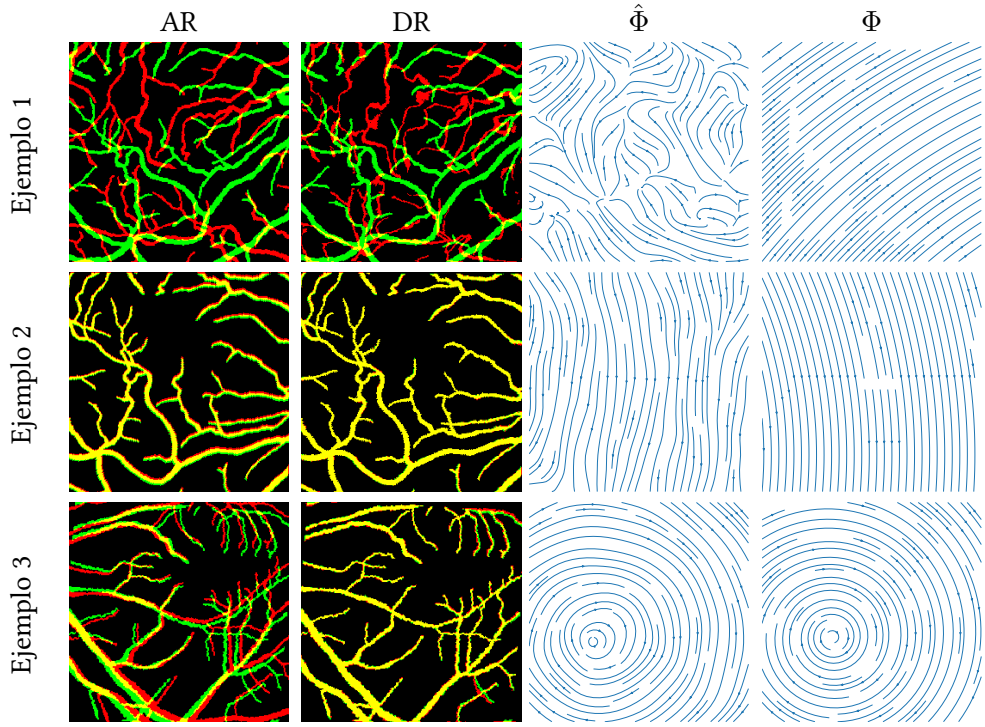


Figura 6.6: Ejemplo de registro con modelo supervisado con GtaBa.

6.6 Eficacia usando otras modalidades de imagen

Ahora que se ha visto que con un entrenamiento supervisado se consigue que la red devuelva flujos aceptables, se experimentará con otras modalidades de imagen para ver si funciona o no el modelo al cambiar estos datos.

Para ello, se van a usar las retinografías de DRIVE y las angiografías de Retino-OCTA.

Listado de experimentos

En esta propuesta experimental se realizan dos experimentos. Siguiendo el planteamiento del último experimento que se ha visto, pero cambiando la modalidad de la imagen usada para entrenar, se tienen los siguientes experimentos:

- **Registro supervisado de *retinal* con GtaBa (SRtaBa)**: El objetivo es comprobar si el planteamiento experimental logrado en el último experimento es efectivo con las retinografías.

- **Registro supervisado de *angiography* con GtaBa (SAtaBa)**: *Ídem* que el anterior, pero con angiografías.

6.6.1 Registro supervisado de *retinal* con GtaBa

El identificador de este experimento es SRtaBa, el generador es compuesto e implementa la estrategia GenParTAso con la variante básica, es decir, el generador es GtaBa.

Configuración experimental

La red recibe como entrada las retinografías, pero la evaluación del coeficiente de Dice se hace sobre los vasos segmentados correspondientes. Los hiperparámetros del entrenamiento son los presentes en la Tabla 6.11 y por tanto la red se entrenó sobre 384000 pares.

Resultados

En la Tabla 6.14 aparecen los resultados del experimento, se percibe una TE aceptable con la que se puede deducir que la red está funcionando.

Dice	TE
0.8184 ($\pm$ 0.0533)	64.26%

Tabla 6.14: Resultados cuantitativos del registro supervisado de *retinal* con GtaBa.

Conclusiones

En comparación a los resultados de usar solamente los vasos sanguíneos, se puede apreciar que a pesar de tener una media de Dice similar, la TE en este caso es más baja, probablemente debido a que las imágenes usadas para inferir el flujo son más complejas.

6.6.2 Registro supervisado de *angiography* con GtaBa

El identificador de este experimento es SAtaBa, el generador es simple e implementa la estrategia GenParTAso con la variante básica, es decir, el generador es GtaBa.

Configuración experimental

Los hiperparámetros están en la Tabla 6.11, la red se ha entrenado sobre 384000 pares.

Resultados

Con angiografías no es posible utilizar el coeficiente de Dice, pues las imágenes no son binarias, por lo tanto se utiliza la otra métrica que se introdujo con anterioridad, la *Structural Similarity Index* (SSIM). Uno de los parámetros que se pueden elegir en la implementación de

esta función es σ, influye en el valor medido ya que tendrá en cuenta más o menos detalles según el σ utilizado. En este trabajo se establece que $\sigma = 1.5$.

Como se puede ver en la Tabla 6.15, el TE obtenido indica que la red funciona.

SSIM (σ = 1.5)	TE
0.6281 ($\pm$ 0.1015)	84.86%

Tabla 6.15: Resultados cuantitativos del registro supervisado de *angiography* con GtaBa.

6.6.3 Conclusiones generales

El cambio en las modalidades de imagen no parecen afectar, de forma significativa, en la capacidad del modelo para aprender y registrar imágenes con estas modalidades.

6.7 Extensión del rango de transformación

Los modelos entrenados hasta ahora funcionan relativamente bien cuando se les pide realizar el registro sobre pares cuya transformación es similar al rango que se ha utilizado en el entrenamiento, sin embargo al extender este rango. la red comienza a distorsionar partes del flujo que genera.

A continuación se muestran ejemplos de esto y como el refinamiento puede ayudar en mejorar la efectividad de la red en estos rangos extendidos sin perder mucho en los rangos en los que ya funcionaba. Para ello se va a comparar los resultados de la red al entrenar desde el principio con el rango extendido, con una red entrenada usando de base el que ya se tiene de los anteriores experimentos supervisados. Esto último se conoce como refinamiento de un modelo, que principalmente consiste en realizar ajustes sobre un modelo con un propósito similar al nuevo objetivo, que en este caso es aumentar el rango de las transformaciones.

En los siguientes experimentos, se utilizan en todos los casos los mismos hiperparámetros que están en la Tabla 6.11 y por lo tanto son entrenados con 384000 pares.

Listado de experimentos

Se ha realizado un experimento por cada modalidad de imagen oftalmológica.

- Registro con rango extendido en vasos sanguíneos.
  - Registro supervisado de *vessels* con GtaAm (SVtaAm)
  - Registro supervisado de *vessels* con GtaAm aplicando refinamiento (SVtaAm-rf)
- Registro con rango extendido en retinografías.

  - Registro supervisado de *vessels* con GtaAm (SRtaAm)
  - Registro supervisado de *retinal* con GtaAm aplicando refinamiento (SRtaAm-rf)

- Registro con rango extendido en angiografías.
  - Registro supervisado de *vessels* con GtaAm (SAtaAm)
  - Registro supervisado de *angiography* con GtaAm aplicando refinamiento (SAtaAm-rf)

En cada una de ellas, se entrenan dos modelos con las imágenes generadas con transformadas en un rango extendido. Un modelo es entrenado partiendo de cero (es decir, un entrenamiento habitual, como los que se han estado viendo durante el trabajo), y el otro modelo se entrena utilizando como base el modelo obtenido en los experimentos anteriores, es decir, realiza un refinamiento a partir de éste. El objetivo es realizar una comparativa y ver si el refinamiento ayuda a tener mejores resultados, dando indicios de que un entrenamiento incremental podría ser efectivo para aumentar la capacidad del modelo.

6.7.1 Registro con rango extendido en vasos sanguíneos

El generador es simple e implementa la estrategia GenParTAso con la variante ampliada, es decir, el generador es GtaAm.

Configuración experimental

En este experimento se entrenan dos redes, una sin partir de otro modelo (SVtaAm) y otra en la que se refina desde el modelo obtenido en SVtaBa (SVtaAm-rf).

Resultados

En la Tabla 6.16 se muestran los diferentes resultados obtenidos por estas redes sobre diferentes conjuntos de test. En negrita se marcan los resultados donde el generador usado en entrenamiento y test coinciden. Observando estos resultados se aprecia que en GtaEx la mejor media de Dice es el del modelo refinado y que el modelo sin refinamiento en GtaBa cae drásticamente su TE, sin embargo la TE de SVtaBa sigue superando en todos los casos, algo que es extraño, pues se supone que los modelos entrenados en el rango extendido funcionarían mejor al evaluar en pares generados con dicho rango. Aunque en GtaAm hay que tener en cuenta que parte de los pares son con transformaciones dentro del rango base.

La Figura 6.7 muestra la distribución del coeficiente de Dice de los diferentes modelos con cada generador planteado. A su vez, se observa su valor Antes de Registro (AR). El punto

Exp.	GtaBa		GtaAm		GtaEx	
	Dice	TE	Dice	TE	Dice	TE
AR	0.3613 (± 0.1153)	-	0.2556 (± 0.0858)	-	0.2276 (± 0.0573)	-
SVtaBa	**0.8776 (± 0.0389)**	**88.06%**	0.7381 (± 0.1306)	30.66%	0.6093 (± 0.1002)	0.02%
SVtaAm	0.7866 (± 0.0412)	19.20%	**0.7101 (± 0.0808)**	**3.16%**	0.6425 (± 0.0746)	0.00%
SVtaAm-rf	0.8128 (± 0.0383)	60.32%	**0.7477 (± 0.0732)**	**16.28%**	0.6876 (± 0.0710)	0.00%

Tabla 6.16: Resultados de los experimentos supervisados de *vessels* con generadores en diferentes rangos.

blanco central representa la mediana, los extremos del rectángulo negro con el primer y tercer cuartil, y en color la distribución de los valores obtenidos.

En la Figura 6.8, se puede ver con mayor claridad el rendimiento general de las redes, se puede ver como las que están entrenadas sobre le rango extendido tiene mayor TE a medida que se aumenta el umbral para considerarlo como exitoso.

Por otro lado en la Figura 6.9 se puede ver la diferencia en el error del flujo al compararlo con SVtaBa (mediante la Ecuación 5.11). Se observa cómo en GtaBa el grado de error es menor en la red refinada y cómo en GtaEx ambas redes mejoran el flujo inferido.

En la Figura 6.10 aparecen ejemplos de registro usando estos tres modelos, se puede ver cómo el modelo base presenta deformaciones exageradas en el ejemplo 2 que los modelos extendidos subsanan.

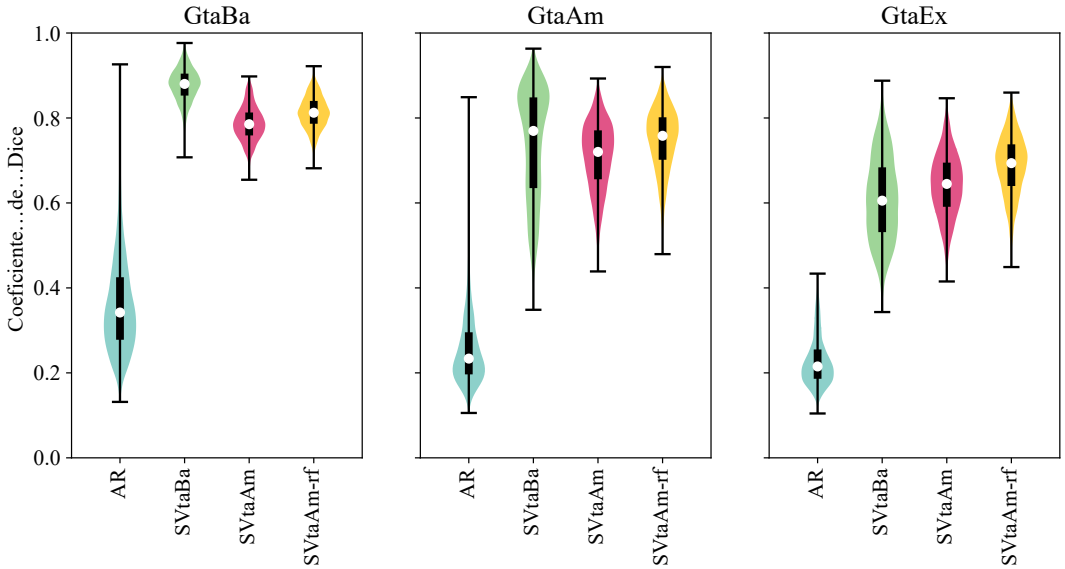


Figura 6.7: Resultado del coeficiente de Dice en los distintos modelos planteados en el experimento con el rango extendido en vasos sanguíneos.

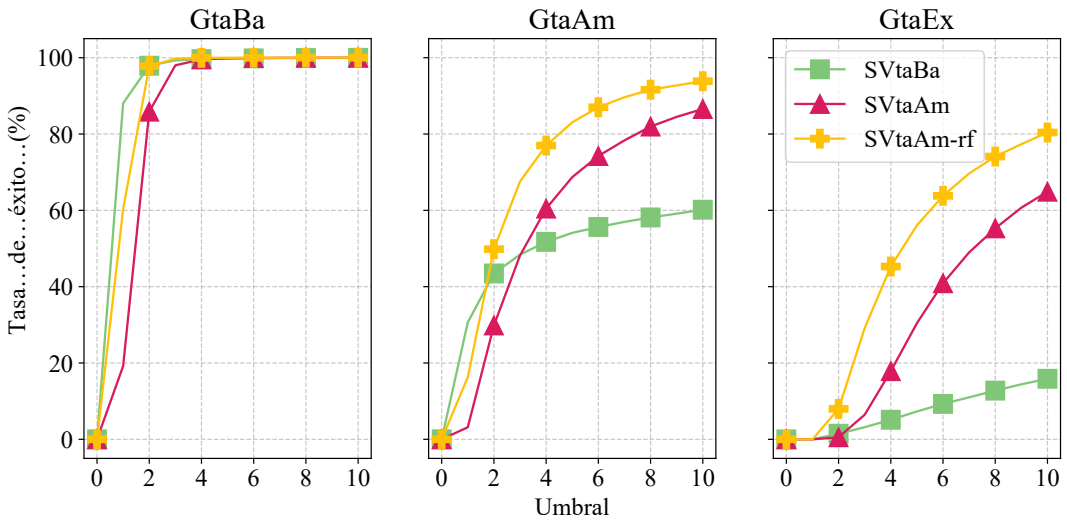


Figura 6.8: Variación de la TE, de los distintos modelos del experimento con el rango extendido en vasos sanguíneos, en función del umbral elegido.

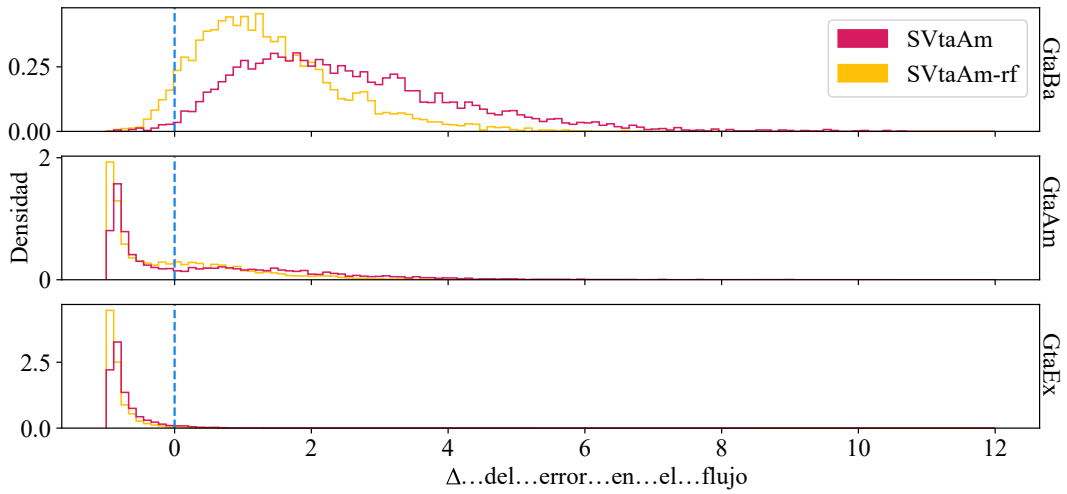


Figura 6.9: Diferencia en el EF de los modelos SVtaAm y SVtaAm-rf comparados con SVtaBa.

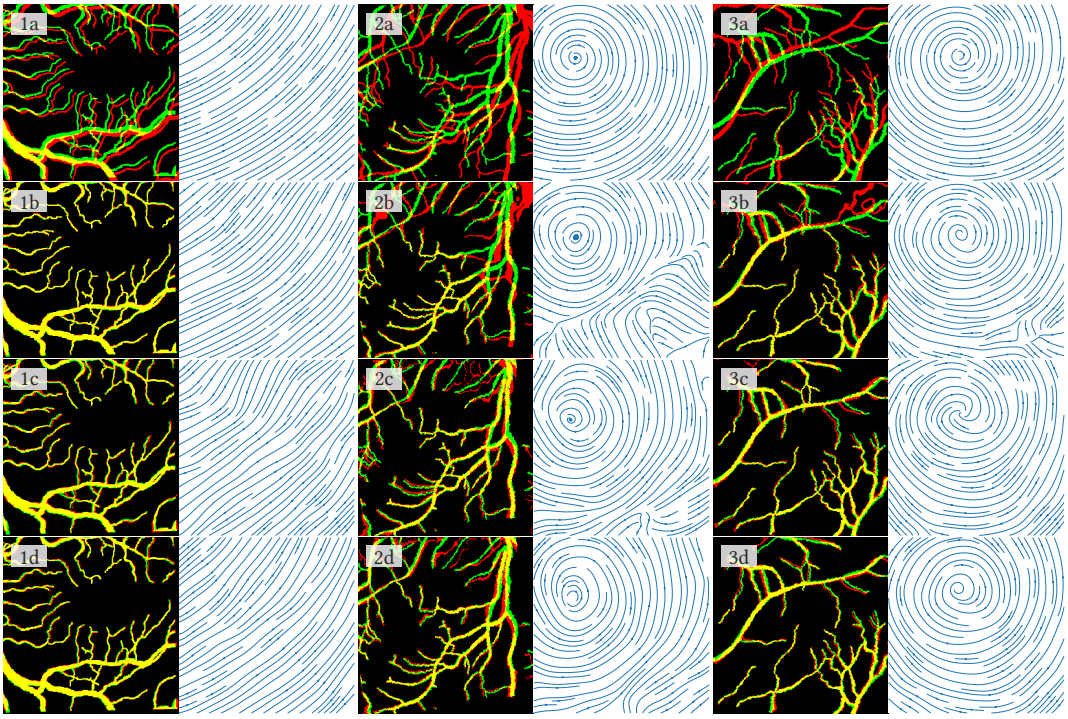


Figura 6.10: Ejemplos de registro en *vessels*. El número representa el ejemplo, en la primera fila se muestra el solapamiento previo al registro y el flujo sintético real. De la *b* a la *d* son los resultados obtenidos por los modelos SVtaBa, SVtaAm y SVtaAm-rf, respectivamente.

6.7.2 Registro con rango extendido en retinografías

El generador es compuesto e implementa la estrategia GenParTAso con la variante ampliada, es decir, el generador es GtaAm.

Configuración experimental

Se entrena una red desde el principio con el rango extendido (SRtaAm) y otra refinando a partir de SRtaBa (SRtaAm-rf).

Resultados

En la Tabla 6.17 se muestran los resultados de los diferentes modelos con distintos generadores, es similar a los resultados en el anterior experimento. En ese caso se ve que usando una red refinada el grado de empeoramiento en el rango base es algo menor en comparación al anterior experimento, mientras que el Dice medido en los otros generadores tiene una mejora más grande. Los resultados siguen indicando que partir de una red refinada se obtienen mejores resultados en general.

Exp.	GtaBa		GtaAm		GtaEx	
	Dice	TE	Dice	TE	Dice	TE
AR	0.3613 (± 0.1153)	-	0.2556 (± 0.0858)	-	0.2276 (± 0.0573)	-
SRtaBa	**0.8184 (± 0.0533)**	**64.26%**	0.6341 (± 0.1487)	13.38%	0.4842 (± 0.0943)	0.00%
SRtaAm	0.6913 (± 0.0583)	0.00%	**0.5969 (± 0.0956)**	**0.02%**	0.5151 (± 0.0870)	0.00%
SRtaAm-rf	0.7898 (± 0.0444)	45.64%	**0.7019 (± 0.0881)**	**10.56%**	0.6313 (± 0.0798)	0.00%

Tabla 6.17: Resultados de los experimentos supervisados de *retinal* con generadores en diferentes rangos.

En la Figura 6.11 se muestra la distribución del coeficiente de Dice en los diferentes modelos con los distintos generadores.

En la Figura 6.12, se muestra el cambio de la TE en función del umbral utilizado. Se puede apreciar claramente que el modelo refinado al evaluarlo sobre imágenes exclusivamente del rango extendido funciona globalmente mejor que el que no partió de un modelo base. A diferencia del anterior experimento, la mejora es más clara, posiblemente debido a las características de las imágenes usadas, ya que en este caso tienen más complejidad que en el anterior experimento, donde eran binarias.

En la Figura 6.13 está la diferencia en el error del flujo, como se puede ver sigue una tendencia similar a lo visto anteriormente, se puede ver cómo el grado de error cometido en el GtaBa por el modelo refinado se concentra en torno al 0, a diferencia del modelo entrenado con vasos, donde se concentraba en torno al 1.

En la Figura 6.14 aparecen ejemplos de registro, donde en vez de mostrar el solapamiento de los vasos segmentados, se visualiza un tablero de ajedrez donde el primer cuadro corresponde a la imagen móvil y el segundo a la imagen fija, de forma alterna. Ambas imágenes tienen el brillo modificado ligeramente, para obtener una visualización más clara de este tablero. En esta ocasión no se observan tantas deformaciones, a parte del ejemplo 2, posiblemente debido al borde exterior que se forma en la imagen fija, donde se diferencia claramente el fondo del contenido de la imagen. En la Figura 6.15 aparece una imagen comparando los resultados obtenidos de esta red con la de *vessels*, en horizontal están los tres ejemplos agrupados en pares, donde el primer par corresponde a un modelo *retinal* y el otro a un modelo *vessels*. Se puede apreciar en el ejemplo 2, cómo los bordes aparecen menos distorsionados en el caso de *retinal*, debido posiblemente a lo que se comentaba anteriormente, la presencia de bordes marcados.

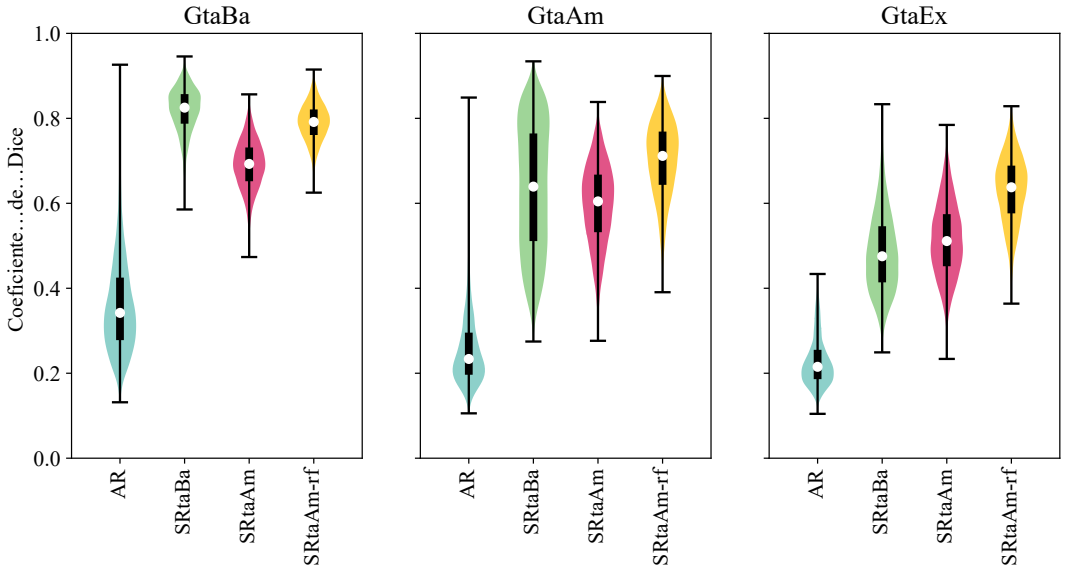


Figura 6.11: Resultado del coeficiente de Dice en los distintos modelos planteados en el experimento con el rango extendido en retinografías.

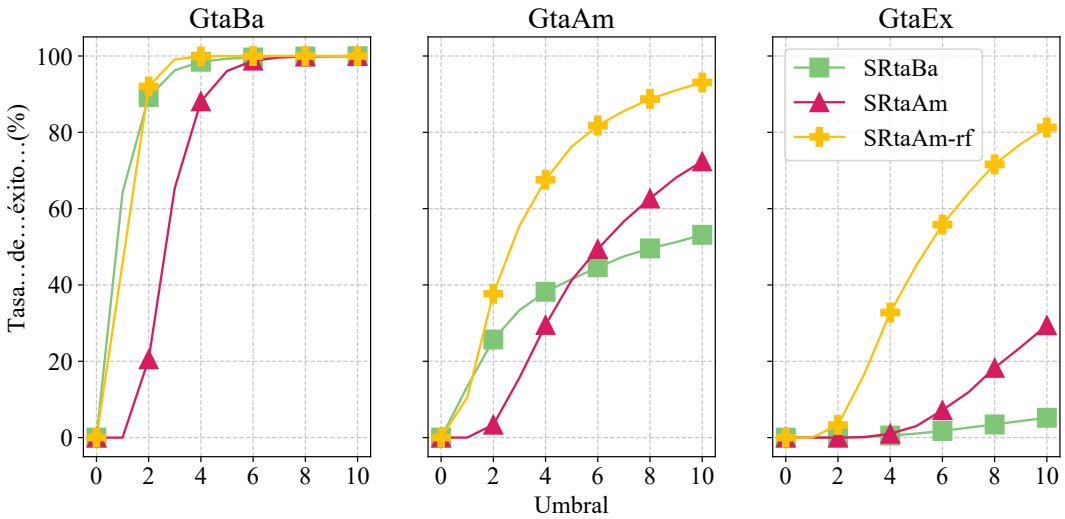


Figura 6.12: Variación de la TE, de los distintos modelos del experimento con el rango extendido en retinografías, en función del umbral elegido.

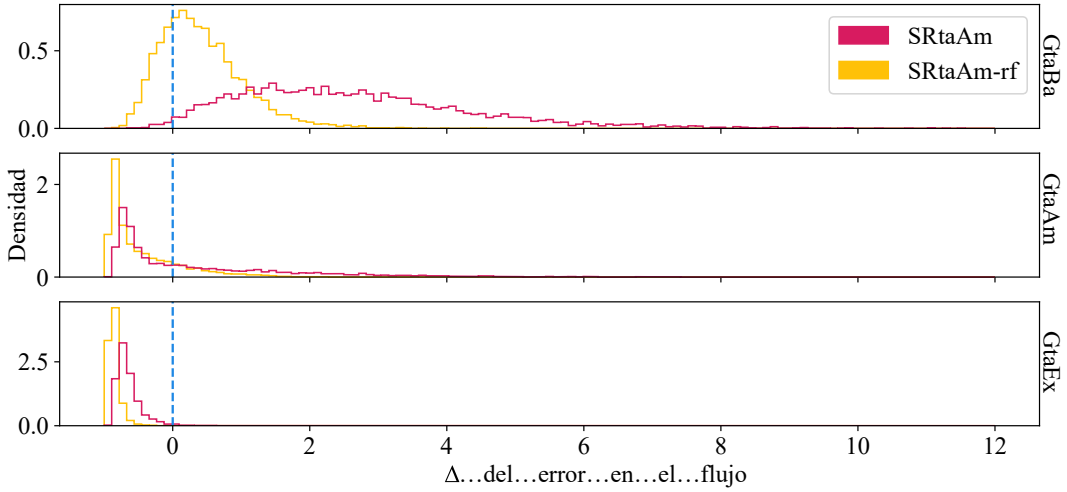


Figura 6.13: Diferencia en el EF de los modelos SRtaAm y SRtaAm-rf comparados con SRtaBa.

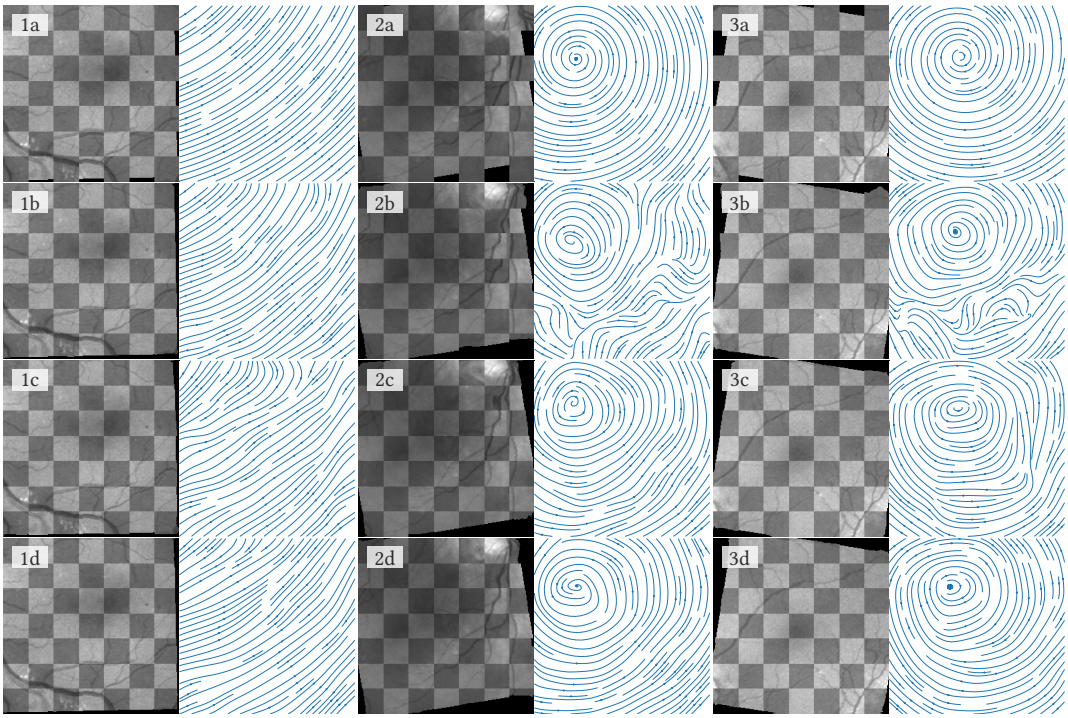


Figura 6.14: Ejemplos de registro en *retinal*. El número representa el ejemplo, en la primera fila se muestra el tablero de ajedrez previo al registro y el flujo sintético real. De la *b* a la *d* son los resultados obtenidos por los modelos SRtaBa, SRtaAm y SRtaAm-rf, respectivamente.

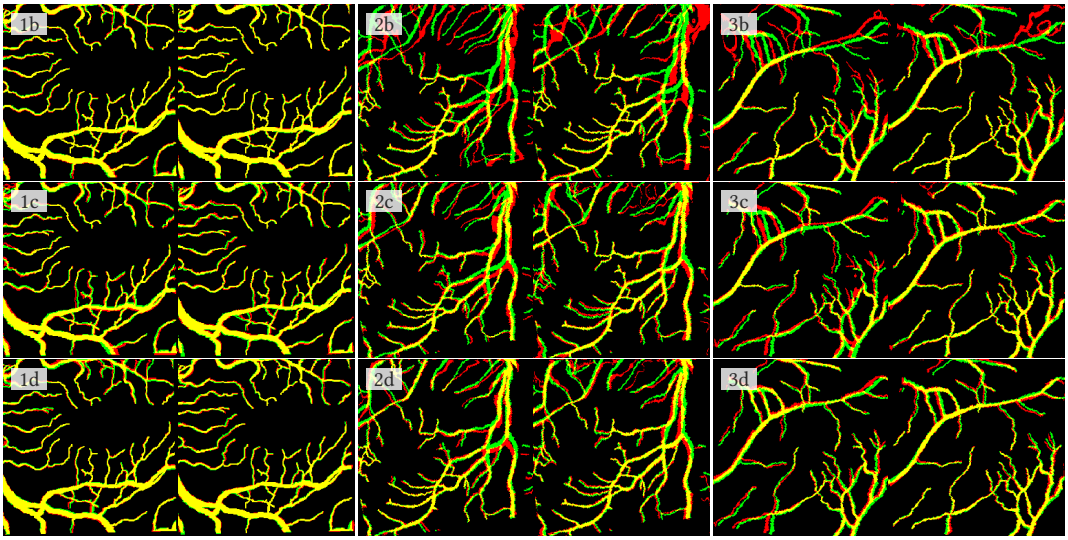


Figura 6.15: Comparativa resultados de registro entre *vessels* y *retinal*.

6.7.3 Registro con rango extendido en angiografías

El generador es simple e implementa la estrategia GenParTAso con la variante ampliada, es decir, el generador es GtaAm.

Configuración experimental

Una red es entrenada desde el inicio sobre el rango extendido (SAtaAm) y otra red usando como base SAtaBa para refinar (SAtaAm-rf).

Resultados

En la Tabla 6.18 se muestran los resultados de los diferentes modelos con distintos generadores, la estructura vuelve a ser similar, donde el modelo refinado tiene mejor valor en el SSIM en el rango exclusivamente aumentado.

Exp.	GtaBa		GtaAm		GtaEx	
	SSIM (σ = 1.5)	TE	SSIM (σ = 1.5)	TE	SSIM (σ = 1.5)	TE
AR	0.0608 ($\pm$ 0.0519)	-	0.0420 ($\pm$ 0.0240)	-	0.0394 ($\pm$ 0.0117)	-
SAtaBa	**0.6281 ($\pm$ 0.1015)**	**84.86%**	0.4045 ($\pm$ 0.1586)	26.38%	0.2630 ($\pm$ 0.0669)	0.00%
SAtaAm	0.4045 ($\pm$ 0.0551)	12.66%	**0.3146 ($\pm$ 0.0789)**	**2.02%**	0.2516 ($\pm$ 0.0500)	0.00%
SAtaAm-rf	0.5201 ($\pm$ 0.0647)	76.32%	**0.3944 ($\pm$ 0.1005)**	**21.58%**	0.3099 ($\pm$ 0.0642)	0.10%

Tabla 6.18: Resultados de los experimentos supervisados de *angiography* con generadores en diferentes rangos.

En la Figura 6.16 se puede observar la distribución del valor de SSIM con las diferentes configuraciones de generador y modelo.

En la Figura 6.17, se muestra el cambio de la TE en función del umbral, que es prácticamente igual al visto en *vessels*.

En la Figura 6.18 está la diferencia en el error del flujo, que sigue una vez más la estructura que se ha estado viendo en los otros experimentos.

En la Figura 6.19 aparecen ejemplos de registro, en este caso se tienen los bordes marcados en los extremos como el caso anterior, además de que las imágenes tienen prácticamente el mismo contraste de forma uniforme. Se puede ver una vez más, como el modelo refinado subsana las deformaciones del ejemplo 2 y 3, a parte de mantener un buen registro en el ejemplo 1.

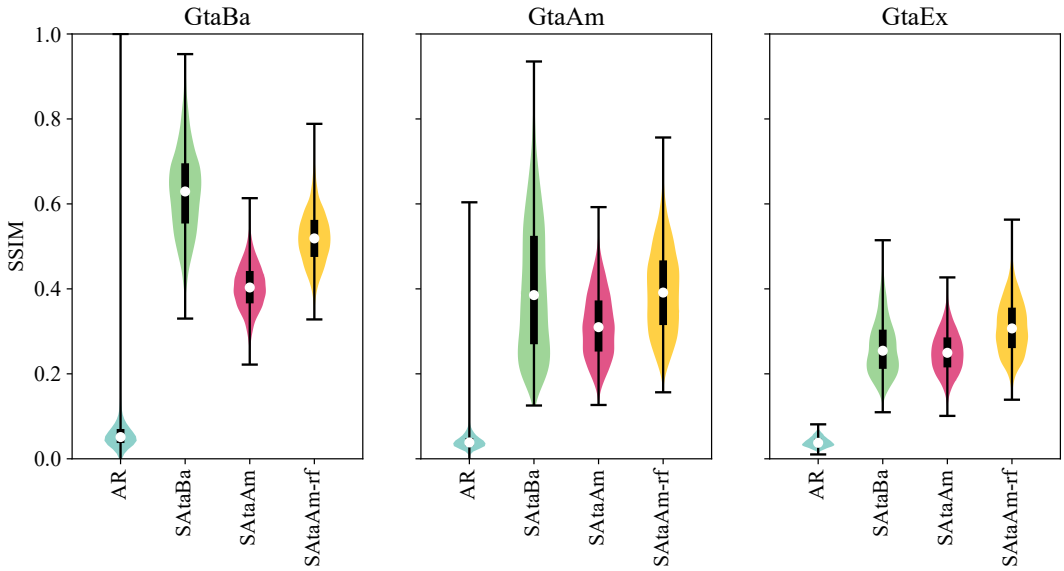


Figura 6.16: Resultado de SSIM en los distintos modelos planteados en el experimento con el rango extendido en angiografías.

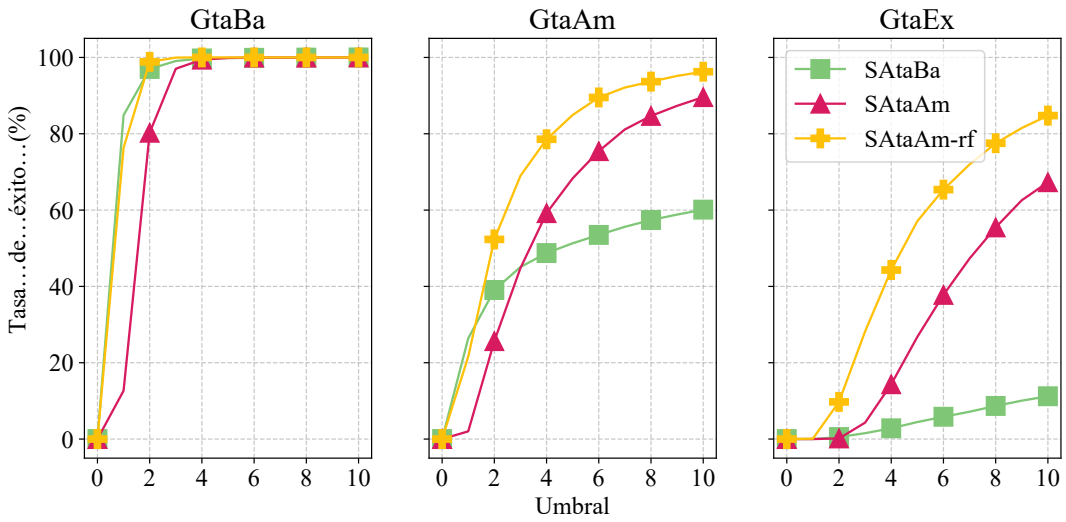


Figura 6.17: Variación de la TE, de los distintos modelos del experimento con el rango extendido en angiografías, en función del umbral elegido.

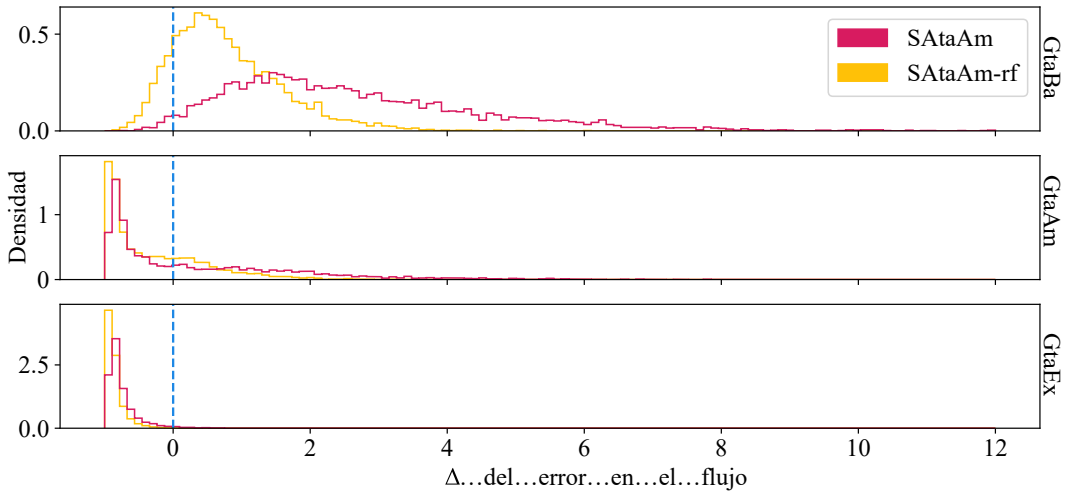


Figura 6.18: Diferencia en el EF de los modelos SAtaAm y SAtaAm-rf comparados con SAtaBa.

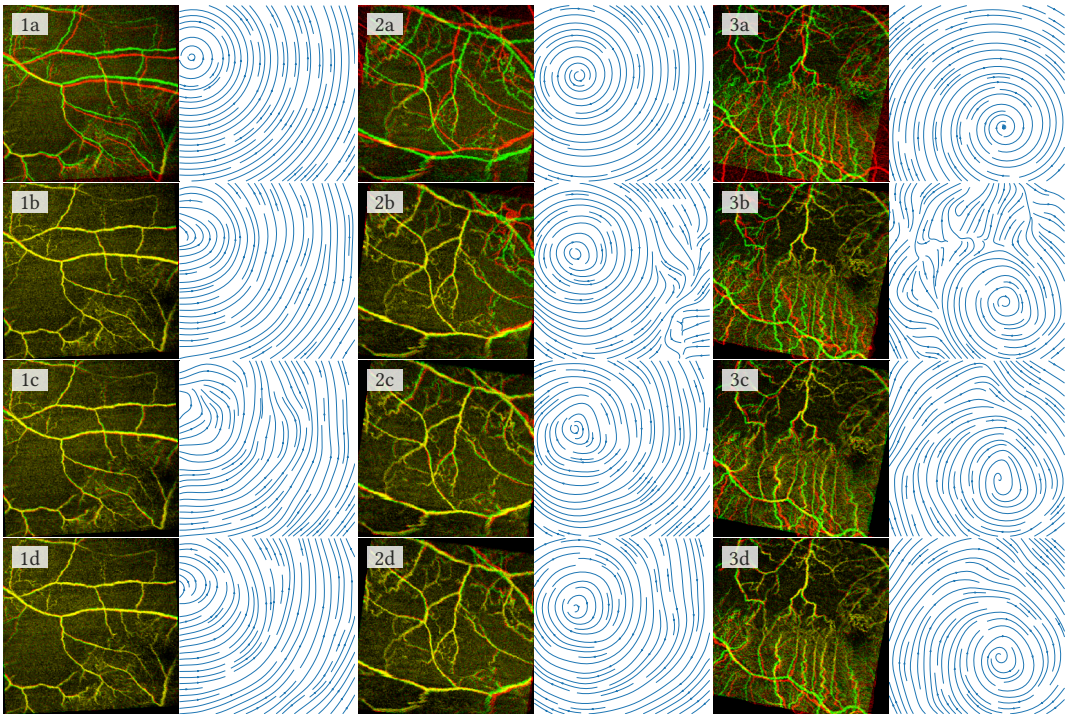


Figura 6.19: Ejemplos de registro en *angiography*. El número representa el ejemplo, en la primera fila se muestra el solapamiento previo al registro y el flujo sintético real. De la *b* a la *d* son los resultados obtenidos por los modelos SAtaBa, SAtaAm y SAtaAm-rf, respectivamente.

6.7.4 Conclusiones generales

El entrenamiento incremental surge de la necesidad de poder ampliar las capacidades del modelo, sin tener que partir de un entrenamiento sobre un conjunto de ejemplos con transformaciones en un rango amplio. La comparativa realizada con las tres modalidades de imagen oftalmológica, parece indicar que el modelo refinado consigue mejores resultados que el modelo no refinado, al menos en el caso de las retinografías. En el caso de los vasos sanguíneos y de las angiografías, esta mejora es menor.

No obstante, hay que tener en mente que los modelos refinados han sido entrenados con más iteraciones que los no refinados, aunque la cantidad de imágenes vistas en el rango extendido es la misma. Esto es debido a que los modelos refinados parten de un modelo que ya ha sido entrenado por un tiempo, por lo que tienen un mejor punto de partida.

Por lo que esta comparativa no es del todo concluyente, sería necesario realizar una comparativa más exhaustiva, en la que se incremente más el rango, la cantidad de iteraciones de entrenamiento sea equivalente y que los modelos sean lo más óptimos posibles, antes de usarse como base para refinar en la extensión.

6.8 Discusión

Durante el transcurso de este trabajo se han realizado diferentes experimentos con dos objetivos concretos. El primero, verificar que los modelos actuales que se están desarrollando para el registro de imagen pueden funcionar con imágenes oftalmológicas y segundo, que el entrenamiento realizado en rangos pequeños e ir aumentando a partir de ahí puede resultar efectivo. El modelo que se utiliza como base es el presentado en Voxelmorph [23].

Para el primer objetivo se han probado estrategias no supervisadas y supervisadas, donde se terminó determinando que con el modelo usado, debido a que el flujo desplazamiento es a nivel local, el entrenamiento supervisado daba resultados mientras que el no supervisado encontraba maneras de optimizar la función objetivo distorsionando el flujo de tal forma que no se obtenía lo que se esperaba. Realizar un entrenamiento supervisado implica tener ejemplos de registro, algo que normalmente es difícil de conseguir ya que habitualmente en imagen médica se hace registro deformable, pero en este trabajo al tratarse de transformaciones simples no resultó tan complicado. Sin embargo, esto hace que se precisen de ejemplos que abarquen los casos clínicos que se pudiera dar en la realidad, algo que en este trabajo no se ha tenido en cuenta, ya que se han generado ejemplos con transformaciones muy limitadas para conseguir que el modelo diese algún resultado.

Esto deriva en el segundo objetivo experimental, ver que al refinar con un rango aumentado la red es capaz de mantener de forma más o menos efectiva lo aprendido y poder ir incluyendo conocimiento nuevo, que son las transformaciones con el rango ampliado. En este

caso hay que hacer un inciso, pues se ha comparado el rendimiento de los modelos entrenados con una misma extensión (i.e. mismos epochs), sin embargo hay que tener en cuenta que el modelo refinado ya partía de base de un modelo entrenado con esa cantidad de ejemplos, esto es, el modelo refinado fue entrenado en realidad con el doble de epochs en comparación con el entrenado sin refinamiento (aunque ciertamente, la cantidad de ejemplos de registro con el rango ampliado fue el mismo). Con los rangos probados, parece que esta estrategia podría funcionar, pero para tener una respuesta más clara habría que optimizar los modelos y experimentar con rangos cada vez mayores, hasta llegar a un grado de transformación que sea acorde a la realidad clínica.

A diferencia del modelo para el registro afín propuesto en [22], donde la red se entrenaba de forma no supervisada y se predecían los parámetros de la matriz de transformación afín, con lo que se puede calcular posteriormente el flujo de desplazamiento; en este trabajo se deja que sea la red la que determine directamente este flujo de desplazamiento, imponiendo la restricción de que tiene que ser afín, o lo más semejante posible, mediante un entrenamiento supervisado. La red del trabajo citado devuelve siempre un flujo afín, mientras que el entrenado en este Trabajo Fin de Grado no lo hace, esto puede suponer tanto una ventaja como una desventaja. Ventaja porque como se ha visto en [24], en la imagen oftalmológica los vasos sanguíneos pueden tener deformaciones y por tanto necesitar de transformaciones no lineales para hacer el registro, algo que con el modelo del trabajo realizado sería posible realizar; en [22] también posibilitan realizar esto, aplicando un registro en múltiples pasos, pero implicaría entrenar estos componentes y por tanto requeriría un mayor tiempo de entrenamiento. La principal desventaja de la aproximación propuesta en este Trabajo Fin de Grado, es la necesidad de tener ejemplos anotados para poder hacer el entrenamiento supervisado y que aún no es claro que esta estrategia funcione cuando las transformaciones son más bruscas.

Capítulo 7

Conclusiones

En este último capítulo se exponen las conclusiones generales a las que se ha llegado tras el trabajo realizado, las posibles líneas futuras de desarrollo en base a los resultados obtenidos y finalmente se comentan las relaciones que pueden establecerse de este trabajo con la titulación.

7.1 Conclusiones

En este trabajo se ha experimentado con un modelo basado en *Deep Learning* (DL) para la realización de la tarea de registro de imagen oftalmológica. El modelo usado como base mapea un campo de deformación a partir de un par de imágenes, fija y móvil, que recibe como entrada. Se ha adaptado el entrenamiento del modelo para conseguir que infiera un campo de deformación afín, a través de ejemplos etiquetados con los flujos reales y, por tanto, a través de un entrenamiento supervisado.

Los resultados obtenidos sugieren que este tipo de registro es posible, incluso usando un modelo de transformación deformable, explotando el uso de ejemplos generados sintéticamente. También se ha hecho un experimento en el que se sigue una estrategia de entrenamiento incremental, en el que al inicio se le suministra al modelo ejemplos de registro con transformaciones pequeñas y posteriormente aumentando dichas transformaciones para abarcar un mayor rango. Dando resultados positivos, indicando que podría ser una buena metodología para aumentar la capacidad de registro del modelo de manera gradual. Ya que un entrenamiento de forma directa sobre un conjunto de transformaciones amplio no fue posible, debido a la naturaleza del modelo; en el que ante esas transformaciones fue más óptimo deformar un flujo para optimizar la similaridad de las imágenes (esto no significa que un entrenamiento no supervisado sea imposible de realizarse, en el siguiente apartado de Trabajo Futuro se dan algunas ideas que podría ser interesante explorar).

7.2 Trabajo futuro

El trabajo realizado tiene una serie de limitaciones y restricciones que se han ido encontrado durante su realización. El hecho de haber realizado un entrenamiento supervisado para conseguir un modelo capaz de realizar el registro de imagen de forma satisfactoria, está sujeto a que los nuevos pares de imágenes tengan que tener características similares, en cuanto a la transformación que se le tiene que aplicar. Por lo tanto, se considera conveniente realizar un estudio sobre la eficacia real del modelo en ámbitos clínicos reales, una forma de hacerlo sería mejorar los generadores implementados para generar ejemplos más fieles a la realidad. Además de que el generador utilizado para conseguir que el entrenamiento funcione, presenta partes que no tiene información (e.g. los bordes en las retinografías), por consiguiente este es otro punto a mejorar.

Por otro lado, sería interesante realizar una comparativa del modelo deformable planteado con un modelo afín. Ya que el modelo deformable necesita inferir una cantidad mayor de parámetros para obtener un resultado que en un modelo afín, donde infiere los parámetros de una matriz de transformación afín directamente, que en el caso de imágenes 2D es de 6 parámetros (para tener en cuenta translación y rotación, que son las transformaciones estudiadas en este trabajo). Por lo tanto se ve como una posible línea futura, la comparación del modelo planteado con un modelo que infiera la matriz de transformación a partir de un par de imágenes (como por ejemplo el modelo propuesto en [22], la parte de registro afín).

Otra posible línea futura, sería diseñar alguna función de pérdida que, en base a las propiedades del flujo obtenido, determine si se trata de una flujo afín o no. Para poder penalizar en consecuencia durante la optimización, para ver si así sería posible entrenar el modelo de forma no supervisada y que siga devolviendo un flujo deformable, pero que sea similar a una afín sin necesidad de guiar el entrenamiento mediante ejemplos.

También cabe destacar que la estrategia de entrenamiento incremental planteado no va muy lejos, pues las transformaciones planteadas siguen siendo muy pequeñas. En parte debido a que no se ha buscando encontrar un modelo óptimo, sino más ver si es viable o no entrenar con imágenes oftalmológicas. Se podría realizar un estudio con mayor profundidad sobre esta estrategia, en primer lugar habría que optimizar los modelos y posteriormente ir incrementando los rangos de manera gradual.

Siguiendo la idea del párrafo anterior, para intentar optimizar el modelo, se podría escoger otra métrica para calcular el error que se comete en la predicción del flujo. Ya que, como se ha podido ver en los resultados obtenidos, la MSE no parece una opción muy acertada. Se podría usar por ejemplo el *End Point Error* (EPE), una métrica utilizada para comparar flujos ópticos en los trabajos sobre *Optical Flow* [36–38].

A continuación se presenta el posible trabajo futuro de forma resumida:

- Mejorar los generadores utilizados para sintetizar pares de imágenes, para que sea lo más fiel posible a la problemática real. Es decir, añadir realismo al generador de forma acorde a las necesidades clínicas.
- Experimentar con otros modelos, en los que la salida sea una matriz de transformación afín, y posteriormente comparar con el modelo entrenado en este Trabajo Fin de Grado. Para ver si compensa realmente el uso de un modelo de transformación deformable para una tarea de registro afín.
- Diseñar una nueva función de pérdida que haga uso de las propiedades de una transformación afín, para permitir un entrenamiento no supervisado del actual modelo.
- Optimizar y continuar investigando en la estrategia del entrenamiento incremental para comprobar si puede ser una buena estrategia para conseguir un modelo capaz de registrar sobre transformaciones mayores a las vistas en este Trabajo Fin de Grado.

7.3 Relación con la titulación

En esta sección se enumeran las asignaturas que han tenido una relación estrecha con respecto a los conocimientos aplicados en el desarrollo de este trabajo.

Visión Artificial. Está relacionada de forma directa con este trabajo, por los temas que trata. Ha servido para adquirir conocimientos sobre preprocesado de imágenes tanto de forma teórica como de forma práctica a través de prácticas desarrolladas en Python.

Sistemas Inteligentes. De esta asignatura se estudiaron los fundamentos de la inteligencia artificial. La relación que tiene con este trabajo radica en los conocimientos sobre redes neuronales y su base teórica.

Aprendizaje Automático. En esta materia se aprenden las diferentes estilos de aprendizaje que hay para la inteligencia artificial, también se introduce el *Deep Learning*. Por otro lado, una de las prácticas llevadas a cabo en esta asignatura sirvió como introducción a lo que podría ser realizar un trabajo de investigación, con su análisis del estado del arte, su experimentación y la recopilación de resultados y su análisis en una memoria.

Computación Gráfica y Visualización. La relación que se establece de esta asignatura a este trabajo, se concreta en la aplicación de matrices de transformación y la importancia del orden en la que se componen estas transformaciones. Ha ayudado especialmente en la parte de los generadores con la obtención de la matriz de transformación que asocia las imágenes del par que se genera.

Gestión de Proyectos. Esta asignatura ha proporcionado conceptos, ideas y fundamentos en la planificación de un proyecto.

7.4 Lecciones aprendidas

Al completar este trabajo, haciendo una reflexión de todo el desarrollo que ha implicado, se llega a enumerar la siguiente lista de lecciones aprendidas:

- En un trabajo de investigación hay que ser flexibles a imprevistos que puedan suceder, y cambiar la perspectiva sobre la que se mira un problema para afrontarlo desde otro punto de vista.
- La organización y registro de los experimentos que se realizan en un trabajo de esta índole es fundamental para su posterior trazabilidad a la hora de recopilar los resultados que se obtienen.
- Hay que seguir leyendo e informándose de los últimos avances e investigaciones que se realizan en el mismo campo o derivados. Para mantenerse actualizado y añadir ideas interesantes que puedan influir en el desarrollo del trabajo propuesto.
- La reproducibilidad es un problema actual en la investigación relacionada con el *Deep Learning* (DL) y por tanto una buena organización, estructura y claridad del código desarrollado es muy importante.

Apéndices

Lista de acrónimos

AR Antes de Registro. 44, 54, 55, 58, 61

BBF *Best Bin First*. 9, 10

CNN *Convolutional Neural Network*. 11, 31

Dice coeficiente de Dice. 33, 44, 50, 52, 54, 55, 57–59

DL *Deep Learning*. VII, 1, 2, 9–11, 14, 16, 40, 67, 70, 77

DRIVE *Digital Retinal Images for Vessel Extraction*. 22, 24, 27, 33, 42, 43, 51

DS *Deep Similarity*. 10

EF Error en el Flujo. 34, 35, 56, 60, 63

EPE *End Point Error*. 68

GenParTAso Generación de Pares mediante Transformación Asociada. 28–30, 38, 43, 46, 47, 49, 52, 54, 57, 61, 73

GenParTInd Generación de Pares mediante Transformación Individual. 28, 29, 38, 43, 44, 73

Gta Generador que implementa la estrategia de GenParTAso. 38, 73

GtaAm Gta Ampliado. 38, 42, 53–55, 57, 58, 61, 74

GtaBa Gta Base. 38, 42, 43, 46–55, 58, 61, 74, 75

GtaEx Gta Exclusivo. 38, 54, 55, 58, 61

Gti Generador que implementa la estrategia de GenParTInd. 38, 42–45, 75

LT Listado de Transformaciones. 26–28

MFSP *Mixture Feature and Structure Preservation.* 9, 10

MSE *Mean Squared Error*. 33, 34, 46, 68

OCT *Optical Coherence Tomography*. 1, 23

OCTA *Optical Coherence Tomography Angiograhy*. 4, 9, 24, 26, 51

PIIFD *Partial Intensity Invariant Feature Descriptor*. 9, 10

RANSAC *Random Sampling Consensus.* 9

ROI *Region Of Interest*. 22, 24

SAtaAm Registro supervisado de *vessels* con GtaAm. 54, 61, 63, 64

SAtaAm-rf Registro supervisado de *angiography* con GtaAm aplicando refinamiento. 54, 61, 63, 64

SAtaBa Registro supervisado de *angiography* con GtaBa. 51, 52, 61, 63, 64

SDT *Signed Distance Transform.* 42, 43, 47–49, 75

SIFT *Scale-Invariant Feature Transform.* 9

SRtaAm Registro supervisado de *vessels* con GtaAm. 54, 57, 58, 60

SRtaAm-rf Registro supervisado de *retinal* con GtaAm aplicando refinamiento. 54, 57, 58, 60

SRtaBa Registro supervisado de *retinal* con GtaBa. 51, 52, 57, 58, 60

SSIM *Structural Similarity Index*. 33, 52, 61, 62

STL *Spatial Transformer Layer*. 31, 33

STN *Spatial Transformer Network.* 11, 31

SURF *Speed-Up Robust-Features.* 9

SVtaAm Registro supervisado de *vessels* con GtaAm. 53–57

SVtaAm-rf Registro supervisado de *vessels* con GtaAm aplicando refinamiento. 53–57

SVtaBa Registro supervisado de *vessels* con GtaBa. 43, 49, 54–57

TE Tasa de Éxito. 35, 44, 46, 48, 50, 52–56, 58, 59, 62, 63

UVtaBa Registro no supervisado de *vessels* con GtaBa. 43, 46

UVtaBa-dt Registro no supervisado de *vessels* con GtaBa aplicando SDT. 43, 47

UVti Registro no supervisado de *vessels* con Gti. 43, 44

Glosario

dataset Es una colección de datos que son utilizados por ordenadores para realizar analíticas o predicciones. 22–24, 26, 42

disco óptico Es una zona circular situada en el centro de la retina, de él emergen los vasos sanguíneos. 11

framework En relación al software, es una abstracción que provee funcionalidades genéricas para el desarrollo de aplicaciones de forma estandarizada. 9, 11, 31

hiperparámetro Nombre que recibe un parámetro, en aprendizaje automático, cuyo valor controla algún aspecto en el proceso de entrenamiento de un modelo. 19, 25, 44, 46, 48, 49, 52, 53

refinamiento Dentro del campo del *Deep Learning* (DL), se refiere al concepto de utilizar los parámetros de un modelo de DL para realizar ajustes sobre estos con el propósito de realizar una tarea similar al modelo anteriormente mencionado. 42, 53, 54, 66, 74

supervisión débil En *Deep Learning* (DL), es un tipo de supervisión en el que se utilizan etiquetas de menor calidad o una cantidad limitada de ellas.. 11

Bibliografía

[1] M. D. Abramoff, M. K. Garvin, and M. Sonka, "Retinal imaging and image analysis," *IEEE Reviews in Biomedical Engineering*, vol. 3, pp. 169–208, 2010. [En línea]. Disponible en: https://doi.org/10.1109/rbme.2010.2084567

[2] P. J. Saine and M. E. Tyler, *Ophthalmic Photography: Retinal Photography, Angiography, and Electronic Imaging*, 2nd ed. Butterworth-Heinemann, 2002.

[3] Y. Li, X. Xia, and Y. Paulus, "Advances in retinal optical imaging," *Photonics*, vol. 5, no. 2, p. 9, Apr. 2018. [En línea]. Disponible en: https://doi.org/10.3390/photonics5020009

[4] T. J. MacGillivray, E. Trucco, J. R. Cameron, B. Dhillon, J. G. Houston, and E. J. R. van Beek, "Retinal imaging as a source of biomarkers for diagnosis, characterization and prognosis of chronic illness or long-term conditions," *The British Journal of Radiology*, vol. 87, no. 1040, p. 20130832, Aug. 2014. [En línea]. Disponible en: https://doi.org/10.1259/bjr.20130832

[5] D. Huang, A. Hagag, S. Gao, and Y. Jia, "Optical coherence tomography angiography: Technical principles and clinical applications in ophthalmology," *Taiwan Journal of Ophthalmology*, vol. 7, no. 3, p. 115, 2017. [En línea]. Disponible en: https://doi.org/10.4103/tjo.tjo_31_17

[6] B. Zitová and J. Flusser, "Image registration methods: a survey," *Image and Vision Computing*, vol. 21, no. 11, pp. 977–1000, Oct. 2003. [En línea]. Disponible en: https://doi.org/10.1016/s0262-8856(03)00137-9

[7] J. Maintz and M. A. Viergever, "A survey of medical image registration," *Medical Image Analysis*, vol. 2, no. 1, pp. 1–36, Mar. 1998. [En línea]. Disponible en: https://doi.org/10.1016/s1361-8415(01)80026-8

[8] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, Nov. 2004. [En línea]. Disponible en: https://doi.org/10.1023/b:visi.0000029664.99615.94

[9] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool, "Speeded-up robust features (SURF)," *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346–359, Jun. 2008. [En línea]. Disponible en: https://doi.org/10.1016/j.cviu.2007.09.014

[10] C. Harris and M. Stephens, "A combined corner and edge detector," in *Procedings of the Alvey Vision Conference 1988*. Alvey Vision Club, 1988. [En línea]. Disponible en: https://doi.org/10.5244/c.2.23

[11] J. Chen, J. Tian, N. Lee, J. Zheng, R. T. Smith, and A. F. Laine, "A partial intensity invariant feature descriptor for multimodal retinal image registration," *IEEE Transactions on Biomedical Engineering*, vol. 57, no. 7, pp. 1707–1718, Jul. 2010. [En línea]. Disponible en: https://doi.org/10.1109/tbme.2010.2042169

[12] M. A. Fischler and R. C. Bolles, "Random sample consensus," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, Jun. 1981. [En línea]. Disponible en: https://doi.org/10.1145/358669.358692

[13] J. Beis and D. Lowe, "Shape indexing using approximate nearest-neighbour search in high-dimensional spaces," in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE Comput. Soc, 1997, pp. 1000–1006. [En línea]. Disponible en: https://doi.org/10.1109/cvpr.1997.609451

[14] J. Wang, A. Camino, X. Hua, L. Liu, D. Huang, T. S. Hwang, and Y. Jia, "Invariant features-based automated registration and montage for wide-field OCT angiography," *Biomedical Optics Express*, vol. 10, no. 1, p. 120, Dec. 2018. [En línea]. Disponible en: https://doi.org/10.1364/boe.10.000120

[15] H. Tang, A. Pan, Y. Yang, K. Yang, Y. Luo, S. Zhang, and S. H. Ong, "Retinal image registration based on robust non-rigid point matching method," *Journal of Medical Imaging and Health Informatics*, vol. 8, no. 2, pp. 240–249, Feb. 2018. [En línea]. Disponible en: https://doi.org/10.1166/jmihi.2018.2283

[16] X. Chen, A. Diaz-Pinto, N. Ravikumar, and A. Frangi, "Deep learning in medical image registration," *Progress in Biomedical Engineering*, Dec. 2020. [En línea]. Disponible en: https://doi.org/10.1088/2516-1091/abd37c

[17] X. Cheng, L. Zhang, and Y. Zheng, "Deep similarity learning for multimodal medical images," *Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization*, vol. 6, no. 3, pp. 248–252, Apr. 2016. [En línea]. Disponible en: https://doi.org/10.1080/21681163.2015.1135299

[18] F. Gao, Y. Wang, P. Li, M. Tan, J. Yu, and Y. Zhu, "DeepSim: Deep similarity for image quality assessment," *Neurocomputing*, vol. 257, pp. 104–114, Sep. 2017. [En línea]. Disponible en: https://doi.org/10.1016/j.neucom.2017.01.054

[19] O. Eberle, J. Buttner, F. Krautli, K.-R. Muller, M. Valleriani, and G. Montavon, "Building and interpreting deep similarity models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2020. [En línea]. Disponible en: https://doi.org/10.1109/tpami.2020.3020738

[20] M. Zou, J. Hu, H. Zhang, X. Wu, J. He, Z. Xu, and Y. Zhong, "Rigid medical image registration using learning-based interest points and features," *Computers, Materials & Continua*, vol. 60, no. 2, pp. 511–525, 2019. [En línea]. Disponible en: https://doi.org/10.32604/cmc.2019.05912

[21] K. Kuppala, S. Banda, and T. R. Barige, "An overview of deep learning methods for image registration with focus on feature-based approaches," *International Journal of Image and Data Fusion*, vol. 11, no. 2, pp. 113–135, Jan. 2020. [En línea]. Disponible en: https://doi.org/10.1080/19479832.2019.1707720

[22] B. D. de Vos, F. F. Berendsen, M. A. Viergever, H. Sokooti, M. Staring, and I. Išgum, "A deep learning framework for unsupervised affine and deformable image registration," *Medical Image Analysis*, vol. 52, pp. 128–143, Feb. 2019. [En línea]. Disponible en: https://doi.org/10.1016/j.media.2018.11.010

[23] G. Balakrishnan, A. Zhao, M. R. Sabuncu, J. Guttag, and A. V. Dalca, "VoxelMorph: A learning framework for deformable medical image registration," *IEEE Transactions on Medical Imaging*, vol. 38, no. 8, pp. 1788–1800, Aug. 2019. [En línea]. Disponible en: https://doi.org/10.1109/tmi.2019.2897538

[24] B. Zou, Z. He, R. Zhao, C. Zhu, W. Liao, and S. Li, "Non-rigid retinal image registration using an unsupervised structure-driven regression network," *Neurocomputing*, vol. 404, pp. 14–25, Sep. 2020. [En línea]. Disponible en: https://doi.org/10.1016/j.neucom.2020.04.122

[25] J. Fan, X. Cao, Q. Wang, P.-T. Yap, and D. Shen, "Adversarial learning for mono- or multi-modal registration," *Medical Image Analysis*, vol. 58, p. 101545, Dec. 2019. [En línea]. Disponible en: https://doi.org/10.1016/j.media.2019.101545

[26] Y. Fu, Y. Lei, T. Wang, W. J. Curran, T. Liu, and X. Yang, "Deep learning in medical image registration: a review," *Physics in Medicine & Biology*, vol. 65, no. 20, p. 20TR01, Oct. 2020. [En línea]. Disponible en: https://doi.org/10.1088/1361-6560/ab843e

[27] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu, "Spatial transformer networks," in *Advances in Neural Information Processing Systems*, vol. 28. Curran Associates, Inc., Jan. 2015, pp. 2017–2025.

[28] J. Lee, P. Liu, J. Cheng, and H. Fu, "A deep step pattern representation for multimodal retinal image registration," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE, Oct. 2019. [En línea]. Disponible en: https://doi.org/10.1109/iccv.2019.00518

[29] G. Luo, X. Chen, F. Shi, Y. Peng, D. Xiang, Q. Chen, X. Xu, W. Zhu, and Y. Fan, "Multimodal affine registration for ICGA and MCSL fundus images of high myopia," *Biomedical Optics Express*, vol. 11, no. 8, p. 4443, Jul. 2020. [En línea]. Disponible en: https://doi.org/10.1364/boe.393178

[30] Y. Wang, J. Zhang, M. Cavichini, D.-U. G. Bartsch, W. R. Freeman, T. Q. Nguyen, and C. An, "Robust content-adaptive global registration for multimodal retinal images using weakly supervised deep-learning framework," *IEEE Transactions on Image Processing*, vol. 30, pp. 3167–3178, 2021. [En línea]. Disponible en: https://doi.org/10.1109/tip.2021.3058570

[31] M. Díaz, J. Novo, P. Cutrín, F. Gómez-Ulla, M. G. Penedo, and M. Ortega, "Automatic segmentation of the foveal avascular zone in ophthalmological OCT-a images," *PLOS ONE*, vol. 14, no. 2, p. e0212364, Feb. 2019. [En línea]. Disponible en: https://doi.org/10.1371/journal.pone.0212364

[32] A. Hoover, V. Kouznetsova, and M. Goldbaum, "Locating blood vessels in retinal images by piecewise threshold probing of a matched filter response," *IEEE Transactions on Medical Imaging*, vol. 19, no. 3, pp. 203–210, Mar. 2000. [En línea]. Disponible en: https://doi.org/10.1109/42.845178

[33] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*. Springer International Publishing, 2015, pp. 234–241. [En línea]. Disponible en: https://doi.org/10.1007/978-3-319-24574-4_28

[34] L. R. Dice, "Measures of the amount of ecologic association between species," *Ecology*, vol. 26, no. 3, pp. 297–302, Jul. 1945. [En línea]. Disponible en: https://doi.org/10.2307/1932409

[35] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Transactions on Image Processing*,

vol. 13, no. 4, pp. 600–612, Apr. 2004. [En línea]. Disponible en: https://doi.org/10.1109/tip.2003.819861

[36] C. Bailer, K. Varanasi, and D. Stricker, "CNN-based patch matching for optical flow with thresholded hinge embedding loss," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, Jul. 2017. [En línea]. Disponible en: https://doi.org/10.1109/cvpr.2017.290

[37] Z. Tu, W. Xie, D. Zhang, R. Poppe, R. C. Veltkamp, B. Li, and J. Yuan, "A survey of variational and CNN-based optical flow techniques," *Signal Processing: Image Communication*, vol. 72, pp. 9–24, Mar. 2019. [En línea]. Disponible en: https://doi.org/10.1016/j.image.2018.12.002

[38] L. Tian, Z. Tu, D. Zhang, J. Liu, B. Li, and J. Yuan, "Unsupervised learning of optical flow with CNN-based non-local filtering," *IEEE Transactions on Image Processing*, vol. 29, pp. 8429–8442, 2020. [En línea]. Disponible en: https://doi.org/10.1109/tip.2020.3013168