



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería Industrial

Diseño y desarrollo de un sistema para predecir el  
potencial de implantación embrionaria en blastocistos  
desvitrificados.

Trabajo Fin de Grado

Grado en Ingeniería Biomédica

AUTOR/A: Hidalgo Blasco, Ayelén

Tutor/a: Naranjo Ornedo, Valeriana

Cotutor/a externo: PAYA BOSCH, ELENA

CURSO ACADÉMICO: 2022/2023

# Agradecimientos

“ A mi familia por siempre estar ahí apoyándome  
A mis amigos, compañeros y tutoras  
por guiarme en este camino.  
Gracias. ”



# Resumen

La baja natalidad y el aumento de la esperanza de vida se presentan como un problema social y económico a largo plazo a nivel global. La baja natalidad está relacionada, en gran medida, con el incremento de la infertilidad en la población debido a diferentes factores sociales, ambientales y económicos, lo que lleva cada día a más parejas a recurrir a tratamientos de reproducción asistida. En este contexto, las transferencias de embriones congelados son cada vez más frecuentes, lo que conduce a la necesidad de optimizar su evaluación. Muchos de estos embriones no son capaces de recuperar sus características cuando vuelven a la temperatura ambiente, de manera que el análisis del periodo comprendido entre la descongelación y la transferencia podría resultar clave. La aplicación de inteligencia artificial sobre la información morfocinética de dicho periodo aportada por los incubadores *time-lapse* se plantea como una herramienta objetiva para identificar características que se relacionen con el potencial de implantación de estos embriones. El objetivo de este TFG es el desarrollo de modelos de *deep learning* capaces de caracterizar los embriones tras la desvitrificación. Esta información es necesaria de cara a evaluar la calidad y predecir el potencial de implantación del blastocisto. Para ello, se hará uso de una base de datos aportada por el Instituto Valenciano de Infertilidad que consta de imágenes de 1 a 6 horas post-desvitrificación y la información clínica asociada. Metodológicamente, se quiere hacer uso de redes neuronales convolucionales que caractericen la información presente en las imágenes. Además, se construirán modelos para la inclusión de los datos clínicos. El trabajo conllevará el desarrollo de algoritmos basados en técnicas de *deep learning* y la validación de los modelos obtenidos.

**Palabras Clave:** *deep learning*, clasificación, embriología, blastocisto, desvitrificación.



# Resum

La baixa natalitat i l'increment de l'esperança de vida es presenten com un problema social i econòmic a llarg termini a nivell global. La baixa natalitat està relacionada, en gran mesura, amb l'augment de la infertilitat en la població a causa de diferents factors socials, ambientals i econòmics, la qual cosa porta cada dia a més parelles a recórrer a tractaments de reproducció assistida. En aquest context, les transferències d'embrions congelats són cada vegada més freqüents, la qual cosa porta a la necessitat d'optimitzar-ne l'avaluació. Molts d'aquests embrions no són capaços de recuperar les seves característiques quan tornen a la temperatura ambient, de manera que l'anàlisi del període comprès entre el descongelament i la transferència podria resultar clau. L'aplicació d'intel·ligència artificial sobre la informació morfocinètica d'aquest període aportada pels incubadors *time-lapse* es planteja com una eina objectiva per identificar característiques que es relacionin amb el potencial d'implantació d'aquests embrions. L'objectiu d'aquest TFG és el desenvolupament de models de *deep learning* capaços de caracteritzar els embrions després de la desvitrificació. Aquesta informació és necessària per avaluar la qualitat i predir el potencial d'implantació del blastocist. Per a això, s'utilitzarà una base de dades aportada per l'Institut Valencià d'Infertilitat que consta d'imatges de 1 a 6 hores post-desvitrificació i la informació clínica associada. Metodològicament, es vol fer ús de xarxes neuronals convolucionals que caracteritzin la informació present a les imatges. A més, es construiran models per a la inclusió de les dades clíniques. El treball comportarà el desenvolupament d'algoritmes basats en tècniques de deep learning i la validació dels models obtinguts.

**Paraules Clau:** *deep learning*, classificació, embriologia, blastocist, desvitrificació.



# Abstract

The low birth rate and the increase in life expectancy are presented as a long-term social and economic problem at a global level. The low birth rate is largely related to the increase of infertility in the population due to different social, environmental and economic factors, which leads more and more couples to resort to assisted reproduction treatments. In this context, frozen embryo transfers are becoming more and more frequent, leading to the need to optimise their evaluation. Many of these embryos are not able to recover their characteristics when they return to room temperature, so analysis of the period between thawing and transfer could be key. The application of artificial intelligence on the morphokinetic information provided by time-lapse incubators during this period is proposed as an objective tool to identify characteristics related to the implantation potential of these embryos. The aim of this TFG is the development of deep learning models capable of characterising embryos after devitrification. This information is necessary in order to assess the quality and predict the implantation potential of the blastocyst. For this purpose, we will make use of a database provided by the Instituto Valenciano de Infertilidad that consists of images from 1 to 6 hours post-devitrification and the associated clinical information. Methodologically, convolutional neural networks will be used to characterise the information present in the images. In addition, models will be built for the inclusion of clinical data. The work will involve the development of algorithms based on deep learning techniques and the validation of the models obtained.

**Key words:** deep learning, classification, embryology, blastocyst, desvitrification.





# Índice general

Resumen	III
Índice general	IX
I Memoria	1
1 Introducción	3
1.1 Descripción de del problema	3
1.1.1 Reproducción asistida.	3
1.1.2 Técnicas de reproducción asistida: IA y FIV.	4
1.1.3 Blastocisto: formación y características	6
1.1.4 Vitricación y desvitrificación.	8
1.2 Motivación	9
1.3 Marco del proyecto: IVI	9
2 Objetivos	11
3 Estado del arte	13
4 Marco teórico	15
4.1 Estructura de las redes neuronales	16
4.1.1 Perceptrón	16
4.1.2 Perceptrón multicapa	17
4.1.3 Funciones de activación.	18
4.1.4 Algoritmo <i>forward-backward propagation</i>	20
4.1.5 Funciones de pérdidas	21
4.2 Redes neuronales convolucionales (CNN)	22
4.2.1 Capa convolucional	22
4.2.2 Capa <i>pooling</i>	24
4.2.3 Capa <i>fully-connected</i>	24

5	Materiales	25
5.1	Hardware y software	25
5.1.1	Hardware	25
5.1.2	Software	26
5.2	Base de datos	27
6	Métodos	29
6.1	Diagrama del flujo de trabajo	29
6.2	Creación y preparación de la base de datos	30
6.2.1	Selección de datos	30
6.2.2	Análisis exploratorio y estadístico de las variables	31
6.2.3	Base de datos de imágenes ( <i>ImageDatastore</i> , IMDS)	33
6.2.4	Base de datos de características ( <i>ArrayDatastore</i> , FTDS)	33
6.2.5	Partición en los subconjuntos de entrenamiento, validación y test	33
6.2.6	Balanceo y aumento de datos	34
6.2.7	Combinación de las dos bases de datos	36
6.3	Entrenamiento	36
6.3.1	Arquitectura	36
6.3.2	Hiperparámetros	39
6.4	Validación cruzada: <i>k-fold cross-validation</i>	41
7	Resultados	43
8	Conclusiones, limitación y líneas futuras	49
II	Presupuesto	51
9	Presupuesto	53
9.1	Presupuestos parciales	53
9.1.1	Mano de obra	53
9.1.2	Maquinaria	54
9.1.3	Materiales	54
9.2	Presupuestos totales	55
	Bibliografía	57

Parte I

Memoria



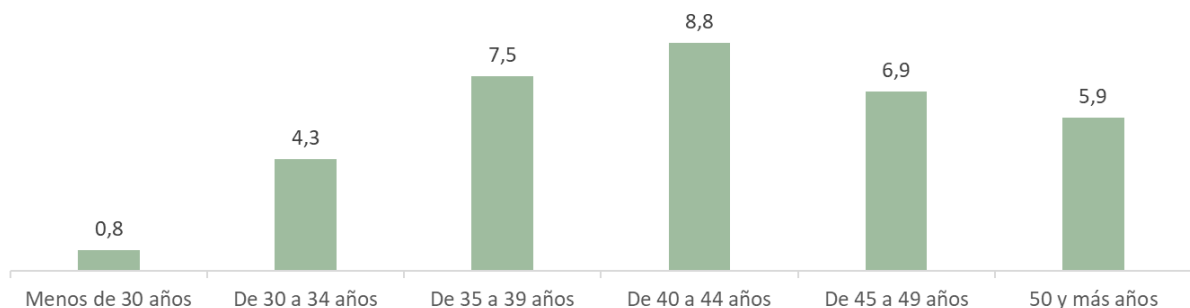
## Introducción

### 1.1 Descripción de del problema

#### 1.1.1 Reproducción asistida

La reproducción asistida (RA) es el conjunto de técnicas y tratamientos médicos que buscan lograr el embarazo en parejas o personas que tienen dificultades para concebir a un hijo de manera natural [1]. Según la Organización Mundial de la Salud (OMS) la infertilidad se define como “una enfermedad del sistema reproductor masculino o femenino consistente en la imposibilidad de conseguir un embarazo después de 12 meses o más de relaciones sexuales habituales sin protección” [2].

En el año 2018 el Instituto Nacional de Estadística refleja en su encuesta de fecundidad que el 5,4 % de las mujeres entre 18 y 55 años que residen en España se han sometido a algún tipo de tratamiento de reproducción asistida [3]. Este porcentaje aumenta con la edad, alcanzando su máximo del 8,8 % en las mujeres entre 40 y 44 años, y luego vuelve a disminuir [3].



**Figura 1.1:** Porcentaje de utilización de técnicas de reproducción asistida por rangos de edad. Extraída de [3].

Durante los últimos años se ha registrado un aumento creciente del uso de estas técnicas. Una de las principales causas está relacionada con el estilo de vida de la sociedad moderna. Las preocu-

paciones principales de las mujeres a edades tempranas han cambiado debido a su incorporación en el mundo laboral. En los últimos años el número de hijos por mujer se está viendo disminuido y retrasado a edades más lejanas [3]. Debido al retraso de la edad de maternidad y paternidad la fecundidad disminuye y la esterilidad aumenta.

Es preciso resaltar también que existen muchos otros factores genéticos, físicos y de estilo de vida que pueden impedir el embarazo. Además, el incremento en la demanda de estas técnicas de reproducción asistida se debe a un mayor conocimiento de su existencia, así como la creciente confianza en su eficacia [4].

Cabe destacar que además, estas técnicas se proponen no solo para lograr el embarazo en parejas infértiles, sino que también para conseguir el embarazo en mujeres que deciden ser madres en solitario, para parejas homosexuales femeninas, para evitar la transmisión de enfermedades graves al niño o de un miembro de la pareja al otro.

### 1.1.2 Técnicas de reproducción asistida: IA y FIV

Las técnicas que se engloban dentro de la RA son la inseminación artificial y la fecundación in vitro (FIV). Además existen variedad de posibilidades, se pueden realizar con los propios gametos de la pareja o con donantes de gametos, ya sean de óvulos o espermatozoides.

La inseminación artificial (IA) es una técnica en la que se depositan los espermatozoides directamente en el aparato genital de la mujer, a menudo dentro del útero (intrauterina) [4]. La fecundación se produce dentro de los genitales femeninos, es decir, in vivo. Es un tratamiento ampliamente utilizado debido a su sencillez, sin embargo, las probabilidades de embarazo no suelen ser muy elevadas. Actualmente, esta técnica se ha ido reemplazando por otras con más probabilidades de éxito.

Hoy en día el tratamiento más utilizado es la fecundación in vitro (FIV). Como su propio nombre indica se produce “in vitro”, fuera de la mujer [1]. Esta técnica consiste en fecundar un óvulo con un espermatozoide en el laboratorio para crear un embrión, que posteriormente se transferirá al útero de la mujer. A diferencia de la IA, la FIV es una técnica de alta complejidad, y se trata del tratamiento de reproducción asistida con mayor posibilidad de éxito [5]. Además, la FIV permite controlar exactamente qué número de embriones se transfieren. En contraposición, en la IA se debe tener cuidado en controlar la estimulación ovárica, ya que con una estimulación excesiva existe riesgo de embarazo múltiple, mientras que con una estimulación insuficiente es probable que no llegue a concebirse el embarazo [5].

En la Tabla 1.1 se resumen las principales diferencias entre ambas técnicas.

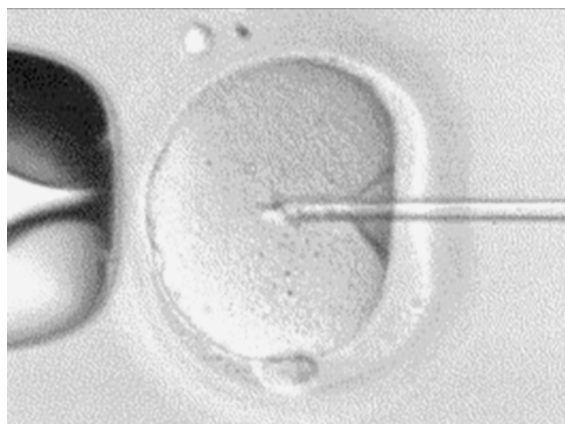
	<b>Inseminación artificial (IA)</b>	<b>Fecundación in vitro (FIV)</b>
Fecundación	<i>in vivo</i>	<i>in vitro</i>
Complejidad	Baja	Alta
Estimulación ovárica	Debe ser mínima y controlada	Se busca obtener un número de óvulos adecuado para usar en el laboratorio
Éxito	Menores posibilidades de embarazo	Mayores posibilidades de embarazo

**Tabla 1.1:** Principales diferencias entre la Inseminación Artificial y la Fecundación In Vitro.

### *Procedimiento FIV*

Para entender mejor cómo funciona la fecundación in vitro, a continuación se describe su procedimiento:

- Estimulación ovárica. Consiste en el empleo de fármacos que promueven el desarrollo de múltiples folículos ováricos, de forma que se produzca la maduración y ovulación de múltiples óvulos en vez de uno solo como suele ocurrir de forma natural cada mes [5]. La razón de esto es obtener varios óvulos para tener varias oportunidades, ya que algunos de ellos no se fertilizan o desarrollan adecuadamente tras la fertilización.
- Seguimiento de la maduración de los óvulos. Se hacen múltiples comprobaciones mediante ecografías y concentración hormonal para saber cuando los óvulos del interior de los folículos están en el estado madurativo adecuado. Se considera que un folículo está maduro cuando, y por tanto contiene un óvulo maduro, cuando su diámetro es superior a los 16-18 mm [6].
- Punción folicular. Se extraen los óvulos maduros mediante una aguja de aspiración a través de la vagina y guiada por ultrasonidos [6]. Para este procedimiento se utiliza anestesia.
- FIV. Se pretende unir los gametos en el laboratorio para conseguir el embrión. Existen dos técnicas para conseguir la fecundación in vitro: [6]
  - Fecundación in vitro convencional. En el método clásico de fecundación in vitro, el embriólogo coloca un óvulo y una gota de semen con miles de espermatozoides en la misma placa de laboratorio, permitiendo que uno de los espermatozoides entre en el óvulo por sí solo sin necesidad de ayuda. Para ello, el espermatozoide debe tener cierta vitalidad y movilidad porque debe acercarse al óvulo y poder atravesar la membrana transparente de este.
  - FIV-ISCI (Inyección intracitoplasmática de espermatozoides). En este caso, el especialista en embriología selecciona espermatozoides con mejor apariencia y luego los inyecta en los óvulos con una microaguja. La ventaja que ofrece este método es que da muy buenos resultados en casos de mal pronóstico en hombres, es decir, cuando los espermatozoides son de mala calidad con poca movilidad o inadecuada morfología.



**Figura 1.2:** Inyección intracitoplasmática de espermatozoides (ISCI) donde se observa la inyección del espermatozoide dentro del óvulo. Extraída de [1].



- Comprobación de la fecundación. Tras 16 a 20 horas post-inseminación (hpi), el especialista en embriología comprueba si se ha formado un cigoto [7]. Se trata de la primera fase embrionaria donde el embrión está compuesto por solo una célula que es consecuencia de la unión de los dos pronúcleos.
- Cultivo de embriones. Los cigotos se dejan en cultivo en las condiciones idóneas para que sigan desarrollándose por las diferentes etapas embrionarias hasta llegar al blastocisto. Durante su estancia en el incubador se analiza su calidad para ver cuál o cuáles son los más adecuados para su implantación.
- Transferencia embrionaria. Se realiza en la etapa de blastocisto, normalmente de los 3 a 5 días de la fecundación [8]. De forma general, solo se transfiere uno o dos embriones. El procedimiento de introducción de los blastocitos en el útero es sencillo e indoloro.
- Criopreservación. Los embriones de buena calidad que no han sido transferidos se congelan o vitrifican para ser utilizados en ciclos posteriores si fallará el primer intento.

### 1.1.3 Blastocisto: formación y características

En los procedimientos de reproducción asistida no todos los embriones generados son aptos para transferencia o vitrificación ya que su viabilidad puede verse comprometida. El objetivo principal de la selección de embriones es identificar el embrión con el mayor potencial de implantación para transferirlo al útero de la futura madre. Para ello es necesario, que se desarrolle correctamente y que tenga una serie de características.

El desarrollo del blastocisto comienza con la formación del cigoto. Esta es la primera fase del desarrollo embrionario y se produce cuando se consiguen unir el pronúcleo femenino y masculino procedentes de los gametos. Posteriormente, aproximadamente tras 30h post-fertilización [9], el cigoto empieza a experimentar una serie de divisiones mitóticas, formándose unas células de menor tamaño llamadas blastómeros. Las células se dividen generando más células cada vez más pequeñas y poco a poco van formando una masa compacta y redondeada. Aproximadamente 3 días después de la fecundación se tiene un embrión de 12 a 32 blastómeros, denominado mórula [9]. Finalmente, de los 4 a 7 días post-fertilización, los blastómeros forman dos estructuras: una capa celular que se sitúa en la periferia (trofotodermo) y será la encargada de originar la placenta; y un grupo de blastómeros internos forma una masa celular interna (MCI) que será la encargada de formar el bebé. Ambas estructuras se diferencian porque entre ellas aparece el blastocele, una cavidad llena de líquido. En este punto, al embrión se le denomina blastocito y ya está listo para ser implantado. El desarrollo desde el cigoto al blastocito se ilustra en la Figura 1.3.

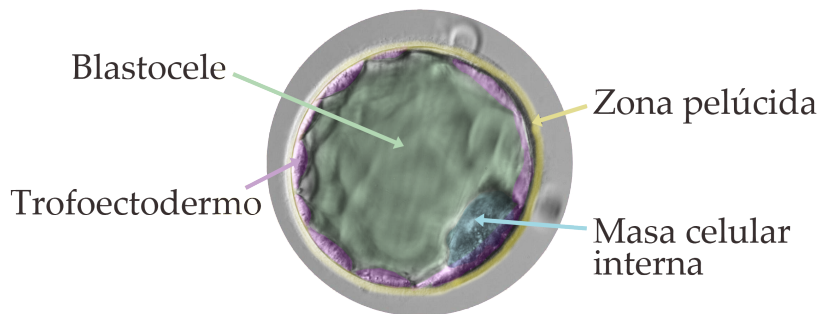


**Figura 1.3:** Desarrollo embrionario del cigoto al blastocisto. Modificada de [10]

Por tanto, en el blastocisto será necesario diferenciar entre:

- Blastocèle
- Zona pelúcida
- Masa celular interna (MCI)
- Trofoectodermo (TE)

Las diferentes zonas se pueden observar en la Figura 1.4.



**Figura 1.4:** Desarrollo embrionario del cigoto al blastocisto. Modificada de [11]

Llegados a este punto, existen ciertos criterios morfológicos que ayudan a los embriólogos a seleccionar los embriones de mayor calidad. Uno de los criterios más utilizados es el propuesto por la Asociación para el Estudio de la Biología de la Reproducción (ASEBIR) en “Criterios ASEBIR de valoración Morfológica de Oocitos, Embriones Tempranos y Blastocistos Humanos” [12]. Los parámetros que se tienen en cuenta son:

- Trofoectodermo. Su estructura y número de células son importantes para la clasificación.
- Masa celular interna. Forma y grado de compactación. Una forma ovalada y compacta parece tener buenas cualidades para su implantación.
- Grado de expansión del blastocèle. La expansión del blastocèle se ha visto que está relacionado con buenas tasas de implantación [12].

Con esto se hace una clasificación en 4 categorías: A (excelente calidad), B (buena calidad), C (calidad intermedia) y D (mala calidad).

En la Tabla 1.2 se recoge un resumen de las relaciones entre las características morfológicas y los grados de calidad.

CATEGORÍA	MCI		TE
	Tamaño ( $\mu\text{m}^2$ )	Cohesión	Descripción
A	3800-1900	Compacta	Homogéneo, cohesionado, muchas células
B	3800-1900	No compacta	Homogéneo, menos células
C	1900	Indiferente	Pocas células
D	Signos de degeneración	Indiferente	Signos de degeneración
Excluído	Degenerada	Indiferente	Degenerado

**Tabla 1.2:** Características morfológicas de la MCI y del TR y su categoría correspondiente. Basado en “Criterios ASEBIR de valoración Morfológica de Oocitos, Embriones Tempranos y Blastocistos Humanos” [12].

Cabe destacar que aunque estos criterios ayudan en la elección un tipo A, no asegura un 100 % de éxito ni un tipo D de fracaso. La transferencia de embriones es un proceso complejo en el que además influyen muchos otros factores.

#### 1.1.4 *Vitrificación y desvitrificación*

Cuando se aplica una técnica FIV se fecundan varios ovocitos y se criopreservan los embriones que no han sido implantados y se consideran de buena calidad. La criopreservación de los embriones permite que los siguientes ciclos del procedimiento sean sencillos y menos invasivos, ya que no es necesario recurrir otra vez a la estimulación ovárica y la punción. Existen dos métodos principales para conservar los embriones: congelarlos de forma convencional o por vitrificación (congelación rápida). Hoy en día, el más utilizado es el segundo ya que presentan mayor tasa de éxito tras el descongelamiento.

La vitrificación es un método de congelación ultrarrápida que ha mejorado las tasas de regeneración de embriones después de la congelación-descongelación. Se basa en congelar los ovocitos a una temperatura muy baja en un periodo de tiempo muy corto. Para ello se utiliza nitrógeno líquido y agentes crioprotectores. El nitrógeno líquido permite conseguir una velocidad de enfriamiento elevada, mientras que los agentes crioprotectores son sustancias que protegen a las células de la formación de cristales de hielo mediante su sustitución por el agua. Es importante que no se formen cristales de hielo, ya que pueden dañar a las células.

Por otro lado, cuando se necesite utilizar los embriones será necesario descongelarlos, a este proceso se le llama desvitrificación. La desvitrificación es el proceso inverso a la vitrificación, se basa en volver a la temperatura natural de los embriones mediante la sustitución de las moléculas crioprotectoras por agua, pero de una forma más lenta. El embrión en su proceso de descongelación va expulsando los crioprotectores y reabsorbiendo el agua hasta recuperar su forma.

Durante la vitrificación y la desvitrificación, los blastocistos sufren una serie de problemas morfológicos, como el colapso del blastocisto, la deshidratación celular y rehidratación posterior. Estos cambios pueden causar daño y pérdida celular, alterando la integridad morfológica. Tras su descongelación es importante volver a evaluar la morfología y características del blastocisto. Esta tarea es muy importante y está todavía a día de hoy en desarrollo. El presente trabajo se centra en este ámbito.

## 1.2 Motivación

Las técnicas de reproducción asistida están en creciente aumento de uso en la actualidad. Debido a las complicaciones que puede acarrear la implantación de varios embriones, a día de hoy solo se transfieren 1 o 2 simultáneamente. De aquí desencadena la importancia de la selección embrionaria, es decir, saber qué embriones son óptimos para la implantación, de forma que se puedan minimizar los costes y los ciclos a los que hay que someter a la mujer para conseguir el embarazo. En este proceso entra también en juego la vitrificación y desvitrificación de los embriones, ya que es otro condicionante de la calidad de los mismos.

Por todos estos motivos, la tarea de elección del embrión es importante y a la vez compleja. Los embriólogos analizan los embriones manualmente y utilizan criterios de clasificación de calidad, pero estos no garantizan la totalidad del éxito. El método de clasificación manual requiere tiempo y es sensible a la variabilidad inter- e intra- observador, lo que lo hace impreciso y subjetivo.

En este contexto, nace la idea de utilizar métodos de inteligencia artificial y aprendizaje automático para clasificar imágenes de blastocistos. El uso de algoritmos de redes neuronales puede ayudar a aumentar la precisión de la selección de embriones viables y de alta calidad para mejorar las tasas de éxito en el embarazo y el parto.

En el presente trabajo de fin de grado, nos centraremos en entrenar una red neuronal para la clasificación de imágenes de blastocistos desvitrificados o en proceso de desvitrificación, con el objetivo de encontrar características conocidas o desconocidas que permitan relacionar las imágenes con su potencial de implantación.

## 1.3 Marco del proyecto: IVI

Este trabajo se enmarca en un proyecto llevado a cabo por el Instituto Valenciano de Fertilidad (IVI) y CVBLab (Computer Vision and Behaviour Analysis Lab), un grupo de investigación de la Universitat Politècnica de València (UPV).

Se busca la introducción de nuevas tecnologías como el *deep learning* para objetivos clínicos. En concreto, en este caso, para la preselección de embriones humanos antes de su implantación.



## Capítulo 2

# Objetivos

El objetivo último de este trabajo final de grado es conseguir la clasificación automática de blastocitos desvitrificados en potencialmente viables o no para su implantación en técnicas de reproducción asistida. Se pretende desarrollar un sistema de inteligencia artificial basado en redes neuronales convolucionales capaz de realizar esta tarea, de forma que sirva de apoyo a la decisión clínica y aumente las probabilidades de éxito, además de ser un método no invasivo.

Para lograr esto, se plantean una serie de objetivos secundarios:

1. Realizar una revisión de la literatura para obtener información acerca del estado del arte de las técnicas de inteligencia artificial y obtener conocimiento acerca del procedimiento de las técnicas de reproducción asistida y de las dificultades o problemas que se presentan.
2. Construir una base de datos de imágenes y características a partir de imágenes clasificadas manualmente por expertos para poder entrenar la red neuronal.
3. Análisis estadístico de las variables numéricas para evaluar si añaden información utilizable en la determinación de las diferentes clases.
4. Partición de los datos en diferentes subconjuntos y balanceo de las clases.
5. Desarrollo una arquitectura basada en redes neuronales convolucionales adecuada para la clasificación y búsqueda de los hiperparámetros óptimos para su entrenamiento.
6. Evaluar los resultados y compararlos con los previamente reportados en la literatura. Analizar la precisión, la exactitud y otras métricas relevantes para determinar la eficacia de la clasificación lograda por el modelo obtenido.
7. Analizar los problemas obtenidos, extraer conclusiones y plantear posibles mejoras de cara a futuros trabajos.



## Capítulo 3

# Estado del arte

Como se comentó en el Capítulo 1, el proceso de elección del embrión a implantar es una tarea compleja que todavía a día de hoy se hace de forma manual. El especialista en embriología responsable sigue una serie de criterios morfológicos y en base a esto toma una decisión. Este método conlleva tiempo, ya que se debe visualizar uno a uno de forma manual y, además, se trata de un método altamente subjetivo y dependiente del observador.

Con intención de mejorar esto, en los últimos años se ha apostado por la introducción de la tecnología *time-lapse* en el campo de la reproducción asistida. La introducción de pequeñas cámaras fotográficas en las incubadoras que mantienen los embriones ha permitido evitar que estos se tengan que extraer de las incubadoras para analizarlos en momentos concretos de su desarrollo. Así que la principal ventaja que proporciona esta tecnología es la posibilidad de analizar el embrión sin perturbar las condiciones de cultivo, ya que se evita daños en los mismos al exponerlos al exterior.

Estas cámaras funcionan tomando fotografías en lapsos de tiempo aproximadamente de 5 a 20 minutos, y luego las imágenes resultantes se unen para formar un vídeo donde se puede observar de forma rápida todo el desarrollo embrionario hasta que el embrión es sacado de la incubadora. Uno de los problemas del método tradicional es que los embriones solo pueden evaluarse en puntos específicos del proceso de desarrollo, de forma que se pierde mucha información que podría ser de utilidad. Así que otro punto a favor de esta tecnología *time-lapse* es que proporciona información morfocinética, relacionada con la dinámica de las células embrionarias. Por tanto, permite disponer de un vídeo del completo desarrollo.

Son varios los estudios que han demostrado la eficacia y mejora de las incubadoras *time-lapse* respecto a las tradicionales, por ejemplo en Rubio et al. [13] se evaluó EmbryoScope® frente a una incubadora estándar. La utilización de la primera mostró mejoras en las tasas de éxito, las justificaciones propuestas fueron que las condiciones de cultivo más rigurosas y estables perturban menos el embrión y existe una menor manipulación de ellos.

Por otro lado, el aumento de información y que esta sea digital abre la puerta a la introducción de algoritmos que puedan analizar las imágenes y datos para obtener características comunes que



permitan diferenciar entre clases. Es en este punto donde se unen la embriología y la inteligencia artificial.

En los últimos años, se han diseñado diferentes algoritmos para realizar esta tarea basados en los marcadores morfocinéticos. Aunque muchos de ellos han presentado buenos resultados, ninguno ha llegado a aceptarse globalmente puesto que la mayoría solo son exitosos de forma interna, cuando se utiliza para datos de otras clínicas o laboratorios los resultados de precisión disminuyen significativamente [14].

En el estudio Meseguer et al. [15] se propuso un modelo basado en árboles de decisión para clasificar los embriones según su probabilidad de éxito. Esto se hizo registrando características morfológicas y parámetros morfocinéticos, como por ejemplo los tiempos específicos de escisión. Mediante análisis estadísticos se determinó que características eran las más relevantes y se utilizaron para crear el modelo de predicción.

El desarrollo de algoritmos de aprendizaje profundo también ha sido explorado en este campo. En el estudio de Zhang et al. [16] mediante el uso de técnicas como DenseNet, LSTM y clasificadores de aumento de gradiente, se creó un modelo complejo llamado STEM+ capaz de predecir la formación de blastocistos y evaluar su calidad. Este modelo se validó comparando sus resultados con los de 4 embriólogos y demostró bastante buenos resultados.

Otro modelo predictivo fue desarrollado por Milewski et al. [17]. Se utilizaron variables tanto cualitativas como cuantitativas, donde destaca la introducción de la edad de la paciente, ya que se observó que tenía bastante impacto en la probabilidad de éxito. El modelo se construyó a partir de un análisis de componentes principales y una posterior clasificación introduciendo las variables relevantes en una red neuronal artificial [17].

En definitiva, desde el comienzo de la visión artificial, se están explorando alternativas automáticas para estandarizar las anotaciones de los sistemas *time-lapse*. A pesar de esto, muy pocos investigadores han optado por métodos de *deep learning* para analizar directamente las imágenes o videos de *time-lapse*. Entrenar un sistema a base de imágenes etiquetadas podría permitir encontrar características para diferenciar entre las clases. Es una tarea compleja que requerirá de muchos datos de entrenamiento, pero que se espera que con ello en algún momento reúna el conocimiento equivalente a cientos de embriólogos y permita decidir con mejor precisión los resultados de la implantación de embriones.

## Capítulo 4

# Marco teórico

En los últimos años, el campo del aprendizaje automático o *machine learning* se ha visto impulsado por el desarrollo y la aplicación de las redes neuronales artificiales o profundas, también conocido como *deep learning*. Este proceso de aprendizaje se llama *profundo* porque las arquitecturas de las redes neuronales se componen de varias capas interconectadas entre sí. Estos métodos han revolucionado muchas áreas, incluida la clasificación de imágenes.

La base del *deep learning* o la unidad mínima funcional es la neurona. Está inspirado en el funcionamiento del cerebro humano y se busca imitar la inteligencia humana a través de la adquisición de conocimiento a través de entrenamiento. A diferencia de los métodos tradicionales de *machine learning*, que dependen en gran parte de la extracción manual de características, las redes neuronales profundas pueden aprender automáticamente patrones abstractos y significativos de los datos. Esto se consigue mediante la combinación de múltiples capas de procesamiento, a su vez compuestas por múltiples neuronas, que combinan las entradas y las transforman para formar una salida.

Una de las arquitecturas más utilizadas en el campo de la clasificación de imágenes es la Red Neuronal Convolutiva (CNN - del inglés, *Convolutional Neural Network*). Las CNN se distinguen por su capacidad para capturar y extraer características esenciales de las imágenes mediante el uso de operaciones de convolución y *pooling*. Estas capas convolucionales permiten la detección de patrones locales y características específicas, mientras que las capas de *pooling* reducen el tamaño y mejoran la invariancia espacial. Además, estas arquitecturas utilizan en su última fase para conseguir la clasificación un perceptrón multicapa. Estos están formados por múltiples neuronas conectadas entre sí mediante conexiones ponderadas. Estas neuronas reciben entradas, realizan unas operaciones y generan unas salidas. El objetivo es que durante el entrenamiento, los pesos de estas conexiones se ajusten para minimizar la diferencia entre la salida de la red y las etiquetas de imagen de entrenamiento válidas.

En este capítulo, se estudia en detalle los principales componentes de las redes neuronales convolucionales, desde perceptrones, capas y grupos convolucionales hasta funciones de activación. Además, se añaden aspectos clave del entrenamiento de CNN, como funciones de pérdida, al-

goritmos de optimización y regularización, que juegan un papel clave en el rendimiento y la generalización de la red.

## 4.1 Estructura de las redes neuronales

### 4.1.1 Perceptrón

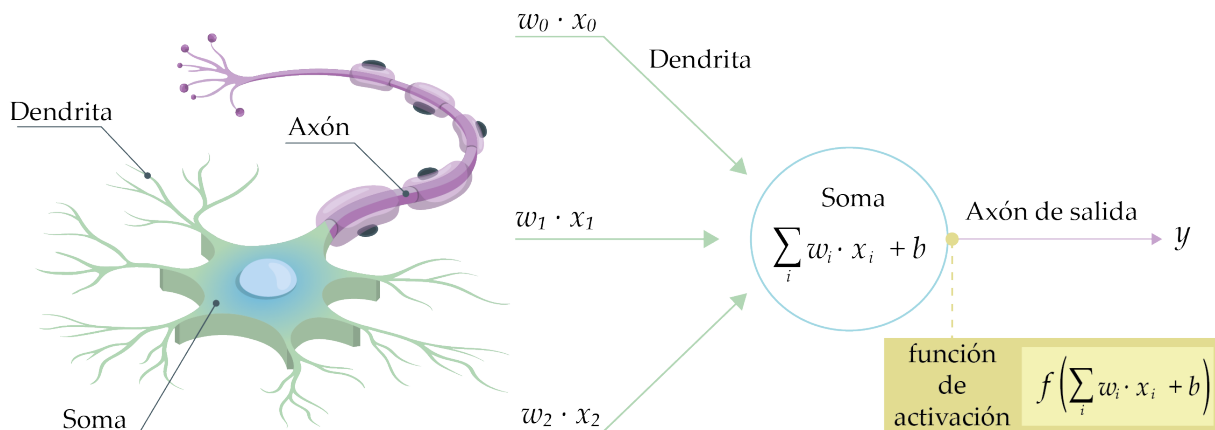
El perceptrón es el modelo más simple de red neuronal. Se trata de la unidad mínima funcional o neurona artificial. Es habitual su comparación con la neurona biológica, pues se trata de una abstracción simplificada de esta. Así pues, en la neurona nerviosa se distinguen tres partes principales:

- Dendritas. Canales por donde se recibe la información.
- Núcleo o soma. Encargado de realizar la operación o procesado de la información.
- Axón. Canal por donde se manda la información de salida o respuesta.

Por otro lado, el perceptrón se puede definir de forma análoga, ya que se compone de:

- Una entrada de características.
- Una parte central donde se realizan operaciones a través de funciones.
- Una salida o respuesta.

En la Figura 4.1 se puede observar las similitudes entre neurona y perceptrón.



**Figura 4.1:** Comparación entre la estructura de una neurona nerviosa y la de un perceptrón.

En el perceptrón la capa de entrada contiene  $n$  nodos uno para cada característica de entrada  $X = [x_1, x_2, \dots, x_n]$ . Cada una de ellas está asociada con un peso  $W = [w_1, w_2, \dots, w_n]$  que representa la importancia de esa entrada para la salida.

En la capa de salida se computa la función de suma ponderada que es una combinación lineal de las entradas multiplicadas por sus pesos respectivos. En algunos casos se puede introducir una parte invariante para la predicción que es el *bias* o sesgo. Matemáticamente se expresa de la

siguiente manera:

$$W \cdot X + b = \sum_{i=1}^n w_i x_i + b \quad (4.1)$$

El resultado obtenido se pasa a la función de activación. La función de activación introduce no linealidad en el modelo y determina si el perceptrón se activa o no. La salida de la función de activación es la salida del perceptrón. Posteriormente, en el apartado 4.1.3 se explican algunas de las funciones de activación más comunes.

Finalmente, la salida del perceptrón se utiliza para realizar una clasificación o tomar una decisión. Cabe destacar que se considera que el perceptrón consta solo de una capa, ya que solo se contabilizan las capas que realizan operaciones.

#### 4.1.2 Perceptrón multicapa

El perceptrón multicapa es una extensión del perceptrón simple. Se introducen nuevas capas de neuronas para añadir complejidad al modelo.

El perceptrón multicapa está formado por tres tipos de capas:

- Capa entrada. Cada neurona de esta capa representa una de las características iniciales y no realizan ninguna operación.
- Capas ocultas. Perceptrones simples o neuronas que siguen el procedimiento y aplican las funciones vistas en el apartado anterior 4.1.1. Estas son las capas que permiten al modelo obtener conocimiento complejo y abstracto de los datos a partir de la modificación de los pesos  $W$ .
- Capa de salida. Realiza la tarea de clasificación final. Requiere de una función de activación que varía de acuerdo al tipo de problema que se contemple: regresión, clasificación, etc.

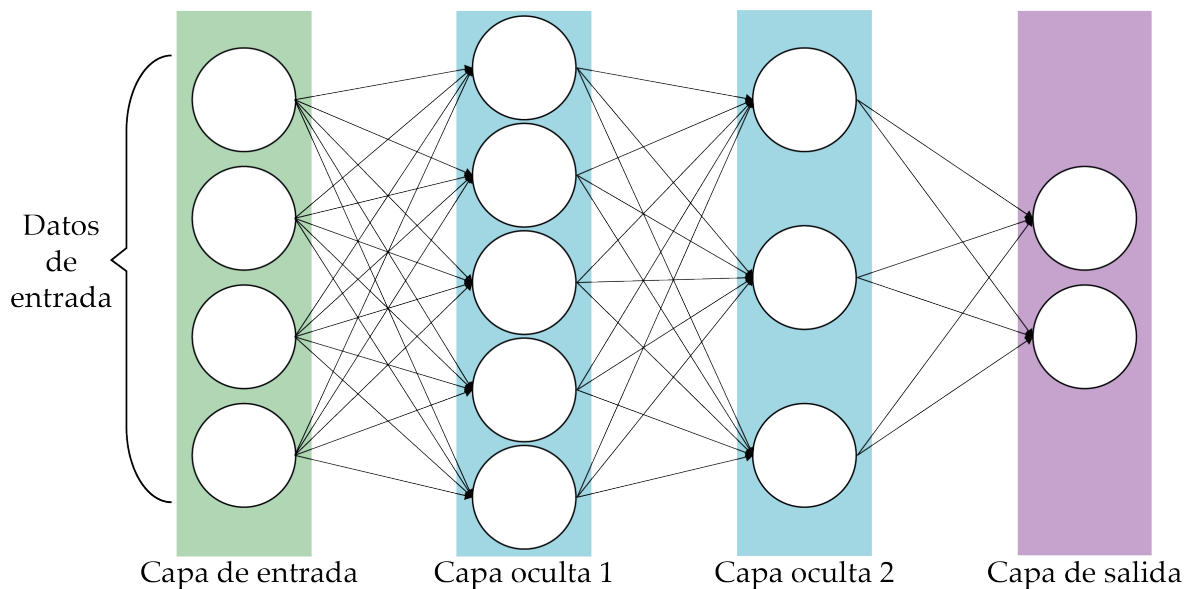


Figura 4.2: Arquitectura básica de un perceptrón multicapa *fully-connected* de tres capas.

En muchos casos las capas son *fully-connected*, significa que están totalmente conectadas entre sí, todas las neuronas conectan con todas las de la capa anterior y las de la siguiente. Además, como se ha visto, cada una de esas conexiones entre neuronas tiene un coeficiente o peso  $w_i$ , de modo que se van a tener tantos coeficientes para pasar de una capa a otra como posibles pares de neuronas.

### 4.1.3 Funciones de activación

Cada una de las capas que conforman la red neuronal tiene una función de activación. Esta función decide cuando una neurona debe activarse o no, es decir, si el *input* que se le ha dado a la neurona es importante o no para el proceso de predicción. Además, añaden no linealidad al modelo, permitiéndolo generalizar y adaptarse a una gran variedad de datos.

#### *Función escalón*

La función escalón, depende de un valor umbral a partir del cual se activa la neurona. Funciona comparándose la entrada a la función con este valor, si la entrada es mayor el resultado es 1 y se activa la neurona. En caso contrario, el valor de salida es 0 y no se activa, lo que significa que su salida no pasa a la siguiente capa oculta.

$$f(x) = \begin{cases} 0 & \text{si } x < 0 \\ 1 & \text{si } x \geq 0 \end{cases} \quad (4.2)$$

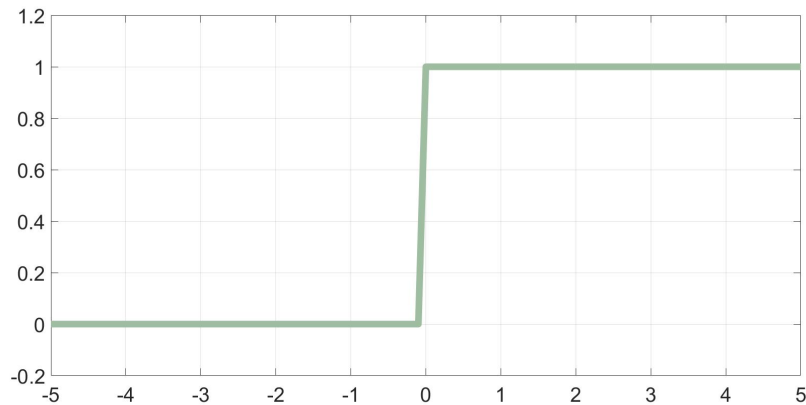


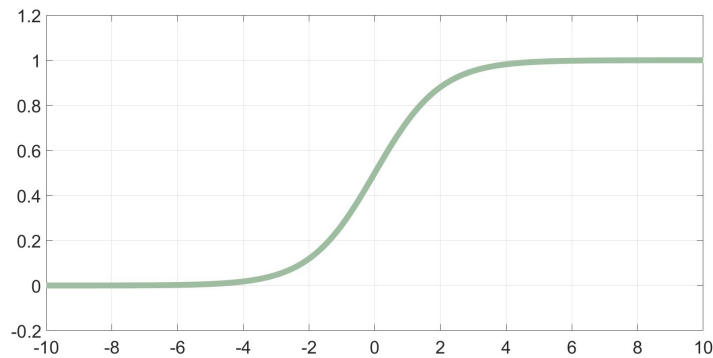
Figura 4.3: Función escalón.

### *Función sigmoide*

La función sigmoide convierte la entrada a la salida en un rango de valores entre 0 y 1. De este modo, permite interpretar la salida como una probabilidad. Cuanto mayor sea el valor de entrada más se acerca la salida a 1 y cuanto menor más se acerca a 0.

Es especialmente útil en problemas de clasificación binaria, donde se puede interpretar como la probabilidad de que una instancia pertenezca a la clase positiva.

$$\sigma(x) = \frac{1}{1 + e^{-x}} \quad (4.3)$$



**Figura 4.4:** Función sigmoide.

### *Función softmax*

De forma similar a la sigmoide, la función softmax toma un vector de valores de entrada y produce un vector de salida en el rango de 0 a 1, donde la suma de todas las salidas es igual a 1. La función softmax transforma los valores de entrada en probabilidades, representando la distribución de probabilidad de pertenecer a cada una de las clases. Por este motivo, esta función se utiliza más en problemas de clasificación multiclase.

$$f(x) = \frac{e^{x_i}}{\sum_{j=1}^N e^{x_j}} \quad (4.4)$$

### *Función ReLU*

La *rectified Linear Unit* (ReLU) es una de las funciones más utilizadas porque permite un aprendizaje bastante rápido en las redes neuronales. Convierte todos los valores negativos a 0 y los positivos los deja igual. Como solo activa ciertas neuronas, la función ReLU es computacionalmente eficiente y acelera la convergencia.

$$\text{ReLU}(x) = \max(0, x) \quad (4.5)$$

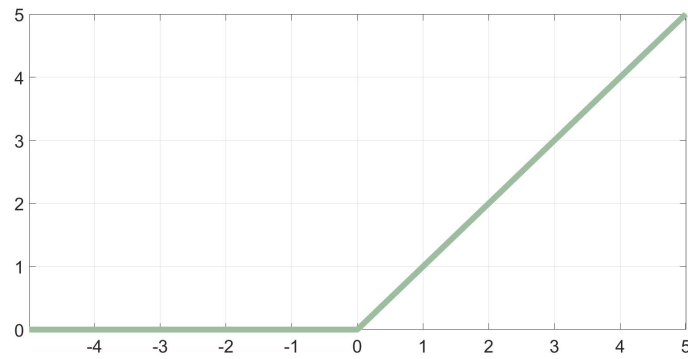


Figura 4.5: Función ReLU.

#### 4.1.4 Algoritmo *forward-backward propagation*

La complejidad de las redes neuronales se encuentra en obtener esos coeficientes  $W$  y  $b$  óptimos, que determinen cuáles son las características importantes y cómo de importantes lo son, para poder realizar una buena clasificación.

Para lograr esto se utiliza el algoritmo de *forward-backward propagation* descrito por Rumelhart, Hinton y Williams [18]. El proceso es el siguiente:

1. Propagación hacia delante o *forward propagation*:
  - I. Inicialmente,  $W$  y  $b$  tendrán un valor aleatorio.
  - II. Se introduce el vector de características de una instancia de entrenamiento en la entrada de la red y se pasa de neurona a neurona calculando la combinación lineal de los pesos y aplicando una función de activación en cada una de ellas.
  - III. En la capa de salida se aplica la función de activación final que permita hacer una predicción de la clase de esa muestra.
  - IV. Se compara la etiqueta real con la predicha por la red y se obtiene un error mediante la utilización de una función de pérdidas.

El objetivo principal es minimizar la función de pérdidas, lo que significaría que la predicción obtenida y el valor real son lo más parecidos posible. La forma de conseguirlo es encontrar los coeficientes  $W$  y  $b$  óptimos que minimicen esta función.

2. Propagación hacia atrás o *backward propagation*:
  - I. Se calcula el gradiente de la función de pérdidas con respecto a los coeficientes  $W$  y  $b$  utilizando la regla de la cadena. Este gradiente indica cómo afecta cada coeficiente al valor de la función de pérdidas.
  - II. Se actualizan los coeficientes  $W$  y  $b$  en dirección opuesta al gradiente, utilizando un algoritmo de optimización para encontrar los valores que minimizan la función de pérdidas. En este punto, se pueden utilizar diferentes optimizadores para actualizar los coeficientes de manera eficiente. Algunos de los más comunes son:

- *Stochastic Gradient Descent* (SGD) con momento (SGDM). Este optimizador actualiza los coeficientes utilizando el gradiente instantáneo junto con una componente de momento que ayuda a acelerar la convergencia del algoritmo. El momento se basa en la acumulación de promedios de gradientes anteriores.
- Adaptive Moment Estimation (Adam). Es un optimizador que combina las ventajas del algoritmo de momentum y del algoritmo de RMSprop. Adapta los coeficientes según la magnitud del gradiente y también según la magnitud del momento acumulado.
- *Root Mean Square Propagation* (RMSprop). Se trata de otro optimizador que también adapta la tasa de aprendizaje en función de los gradientes calculados en cada iteración. Sin embargo, a diferencia de Adam, utiliza un promedio en ejecución de los cuadrados de los gradientes anteriores para actualizar los parámetros.

Llegados a este punto es importante definir dos conceptos:

- *Batch* o lote datos. Conjunto de datos o paquete de datos de entrenamiento. En lugar de pasar muestra por muestra, se pasa un *batch* y se actualizan los pesos tras cada lote. El error que se utiliza es la media de los errores del lote. Esto se realiza para evitar introducir ruido en el modelo, en caso por ejemplo que una imagen estuviese mal etiquetada movería los pesos  $W$  hacia el lado incorrecto. Por tanto, el *backpropagation* se hace para cada *batch*.
- Época. Una época termina cuando se han pasado todos los datos de entrenamiento por la red. Si los datos se encuentran distribuidos en *batches*, en una época se actualizarán los pesos tantas veces como número de *batches* esté dividido el conjunto de entrenamiento.

#### 4.1.5 Funciones de pérdidas

Las funciones de pérdidas  $L$ , también conocidas como funciones de costo, son una parte fundamental del entrenamiento de redes neuronales. Se utilizan para medir el error entre el valor obtenido en la predicción por la red y el valor real. Cuanto menor es el resultado de esta función, mejor es la predicción y por tanto, más eficiente es la red neuronal.

Existen muchas funciones de pérdidas y la elección de una u otra dependerá de la tarea a realizar. En concreto, para clasificación, las más comunes en *deep learning* son:

- Regresión logística. Se utiliza cuando se tienen solo dos clases, es decir, para clasificación binaria. Cuanto mayor sea la correlación entre la predicción  $\hat{y}$  con la etiqueta real  $y$  menor será el resultado exponencial y, por tanto, la función da un resultado de pérdida menor.

$$L = \log(1 + \exp(-y \cdot \hat{y})) \quad (4.6)$$

- Entropía cruzada (*cross-entropy*). Es la más común para los problemas de clasificación multiclase. Esta función compara las distribuciones de probabilidad estimadas ( $\hat{y}$ ) con las verdaderas ( $y$ ). Se calcula la pérdida para todas las clases y la suma de todas ellas proporciona una medida global de las discrepancias entre las distribuciones de probabilidad.

$$L(y, \hat{y}) = -\sum_{i=1}^N y_i \log(\hat{y}_i) \quad (4.7)$$



## 4.2 Redes neuronales convolucionales (CNN)

La resolución de imágenes actuales hace que sean imposible de tratar con perceptrones multicapa, ya que cada píxel contiene información y equivaldría a una variable de entrada, por lo que el vector de entrada sería excesivamente largo. Aquí es donde entran en juego las redes neuronales convolucionales, o *Convolutional Neural Networks* (CNN). Estas redes están específicamente diseñadas para trabajar con datos estructurados en una cuadrícula, como las imágenes, que presentan fuertes dependencias espaciales en regiones locales de la cuadrícula. Con cada capa, la CNN se vuelve más capaz de reconocer características más abstractas y de mayor nivel. Las capas anteriores se enfocan en características simples como colores y contornos, mientras que las capas posteriores se encargan de identificar formas más grandes y complejas, hasta llegar a reconocer el objeto deseado.

Las CNN utilizan diferentes tipos de capas para lograr este proceso de reconocimiento. La capa más importante y característica es la capa convolucional. Esta capa aplica filtros o *kernels* a la imagen de entrada, capturando características locales mediante operaciones de convolución. Estas características se combinan y se pasan a través de capas de agrupación o *pooling*, que reducen la dimensionalidad y conservan las características más importantes. Finalmente, las características extraídas se utilizan en capas completamente conectadas, donde se realiza la clasificación final del objeto. En la figura 4.6 se puede ver un esquema de la arquitectura de las CNN.

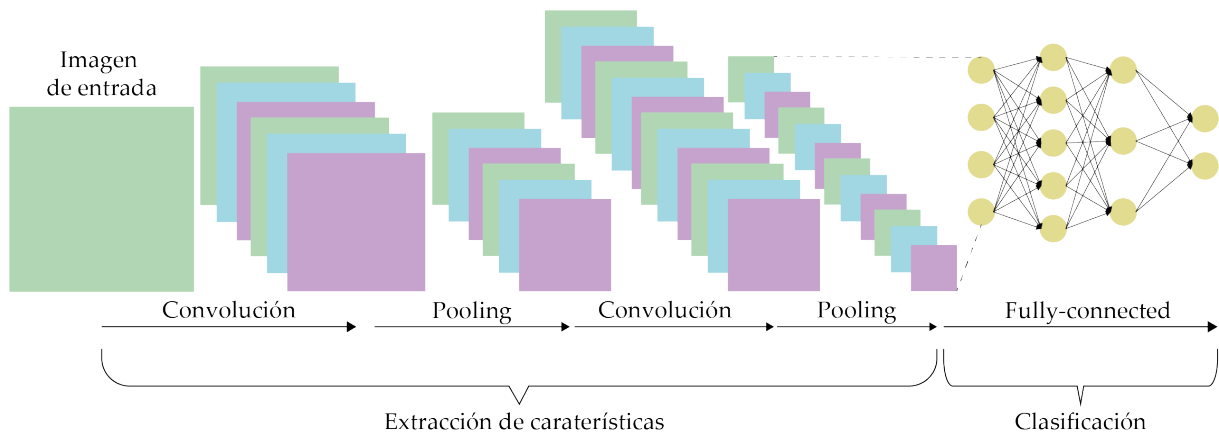


Figura 4.6: Arquitectura de una red neuronal convolucional.

En los siguientes apartados se explicarán con mayor detenimiento las diferentes capas.

### 4.2.1 Capa convolucional

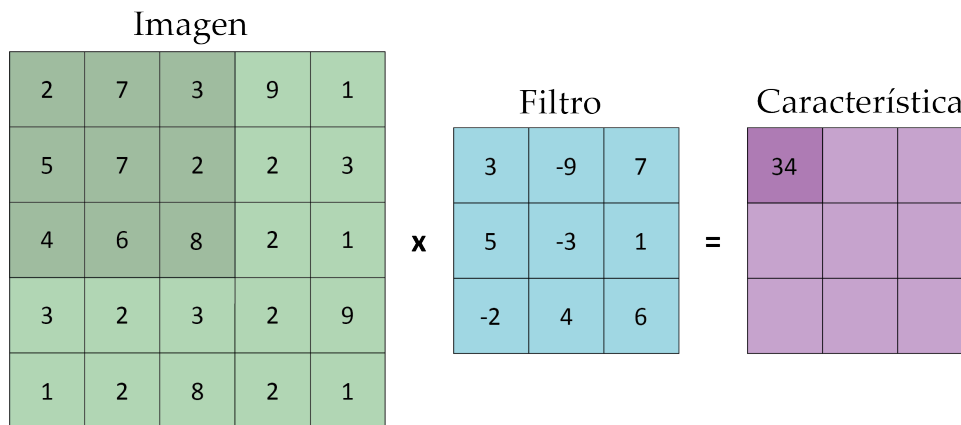
La capa convolucional es la base de las redes neuronales convolucionales. Esta capa se basa en la operación matemática de la convolución.

El objetivo es extraer características relevantes de la imagen. Esto se consigue a través de un conjunto de filtros, también llamados *kernels*, que se deslizan sobre la imagen de entrada, realizando operaciones de convolución en diferentes regiones. Cada filtro captura características específicas de la imagen, como bordes, texturas o formas, mediante la detección de patrones locales. La aplicación de múltiples filtros en paralelo en la capa convolucional permite extraer diversas ca-

racterísticas simultáneamente. A medida que se desliza el filtro sobre el ancho y la altura del volumen de entrada, se produce un mapa de activación en 2 dimensiones que da las respuestas de ese filtro en cada posición espacial.

La red neuronal lo que aprende durante el entrenamiento es a ajustar los pesos de los filtros, de modo que aprende a identificar las características más relevantes para la clasificación. Es decir, intuitivamente la red aprenderá a construir filtros que se activan cuando ven algún tipo de característica visual. Cada uno de ellos producirá un mapa de activación 2-dimensional separado. Estos mapas se apilan a lo largo de la dimensión de profundidad resultando el volumen de salida.

Cabe destacar que los *kernels* son matrices cuadradas y que pueden ser de cualquier profundidad. En caso de tener una imagen a color en la entrada se tienen *kernels* tridimensionales, una dimensión por cada mapa de color RGB. Finalmente, una vez se tiene el mapa de activación se suele aplicar la función de activación ReLU, vista en el apartado 4.1.3, a cada elemento para introducir no linealidad en el modelo.



**Figura 4.7:** Convolución con *kernels* en las imágenes.

En resumen, el procedimiento seguido en una capa convolucional es el siguiente:

1. Se define el tamaño de la ventana de filtro.
2. La ventana de filtro se desplaza progresivamente de izquierda a derecha un determinado número de píxeles definido previamente (el paso) hasta llegar al final de la imagen.
3. En cada trozo de imagen se efectúa el cálculo de convolución permitiendo obtener en la salida un mapa de activación o *feature map* que indica dónde están localizadas las características en la imagen.
4. Finalmente se aplica la función de activación ReLU.

### 4.2.2 Capa pooling

Las capas de pooling se utilizan para ir reduciendo el tamaño de nuestros mapas de activaciones e intentar retener la información más relevante. Se basa en la utilización de ventanas de cierto tamaño, que dividen la imagen en submatrices más pequeñas. Existen dos tipos principales de pooling:

- *Average-pooling*. Calcula la media de los elementos de la ventana.
- *Max-pooling*. Se elige el valor máximo de los elementos de la ventana.

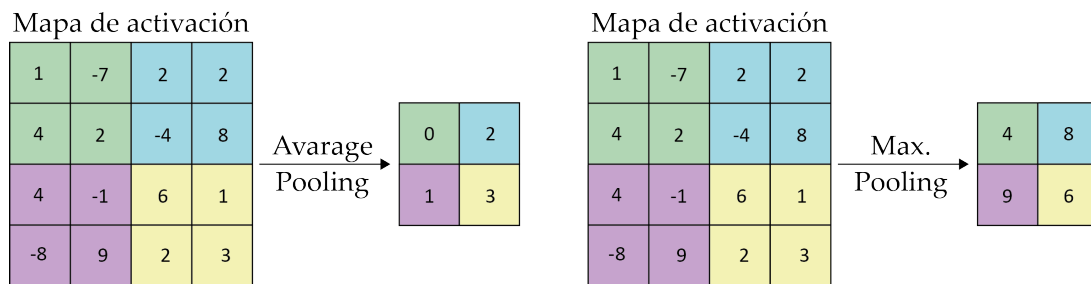


Figura 4.8: Representación de *Average-pooling* (izquierda) y *Max-pooling* (derecha) .

### 4.2.3 Capa fully-connected

Una vez se han conseguido extraer las características más relevantes y reducirlas a un número manejable, estas se pasan a una capa fully connected. La estructura de esta capa y su finalidad aquí son las mismas que las vistas en la sección 4.1.2. Se tiene un conjunto de características y se busca una predicción de clase final mediante el ajuste de los pesos calculando funciones de pérdidas.

En conclusión, las CNN son redes neuronales especiales que buscan la extracción de características con el objetivo de encontrar patrones en los píxeles que permitan una buena clasificación por la capa final *fully-connected*.

## Capítulo 5

# Materiales

### 5.1 Hardware y software

#### 5.1.1 *Hardware*

Antes de realizar un proyecto como el que se lleva a cabo en el presente TFG es necesario conocer las características internas del hardware sobre el que se ejecuta el mismo, pues de él depende tanto el rendimiento, la velocidad y la eficiencia que se puede lograr al compilar y ejecutar los diferentes algoritmos.

En este caso, este trabajo se ha realizado en un ordenador MSI PS63 Modern 8RC con las especificaciones que se muestran en la Tabla 5.1.

Procesador	Intel Core i7 de 8th Generation de 4 núcleos a 4.6GHz
Memoria RAM	16 GB
Disco dura	512 GB SSD
Sistema operativo	Windows 11 Home de 64 bits
Tarjeta gráfica	NVIDIA GeForce GTX 1050

**Tabla 5.1:** Especificaciones del ordenador empleado para el proyecto.

Por otro lado, para realizar el entrenamiento de los algoritmos de *deep learning* se requiere una gran complejidad computacional para evitar largos tiempos de espera, por ello, además del dispositivo especificado en la Tabla 5.1, se ha hecho uso de un computador de elevadas prestaciones con una tarjeta gráfica NVIDIA TITAN Xp.

Finalmente, también se dispone de un servidor Synology, donde se han alojado los datos necesarios para llevar a cabo el proyecto.

### 5.1.2 Software

El principal programa utilizado fue MATLAB®v.R2022b de The MathWorks, Inc. (Natic, Massachusetts, EE. UU). Se utilizó para desarrollar e implementar la creación y organización de las bases de datos de imágenes, la implementación de las arquitecturas de inteligencia artificial, así como para calcular los resultados de este TFG. Este software es una plataforma de programación y cálculo numérico con capacidad para analizar datos, desarrollar algoritmos y crear modelos [19].

Además, se ha hecho uso de la aplicación interna de MATLAB llamada Deep Network Designer. Esta app permite crear, visualizar, editar y entrenar redes de *deep learning* de forma interactiva [20].

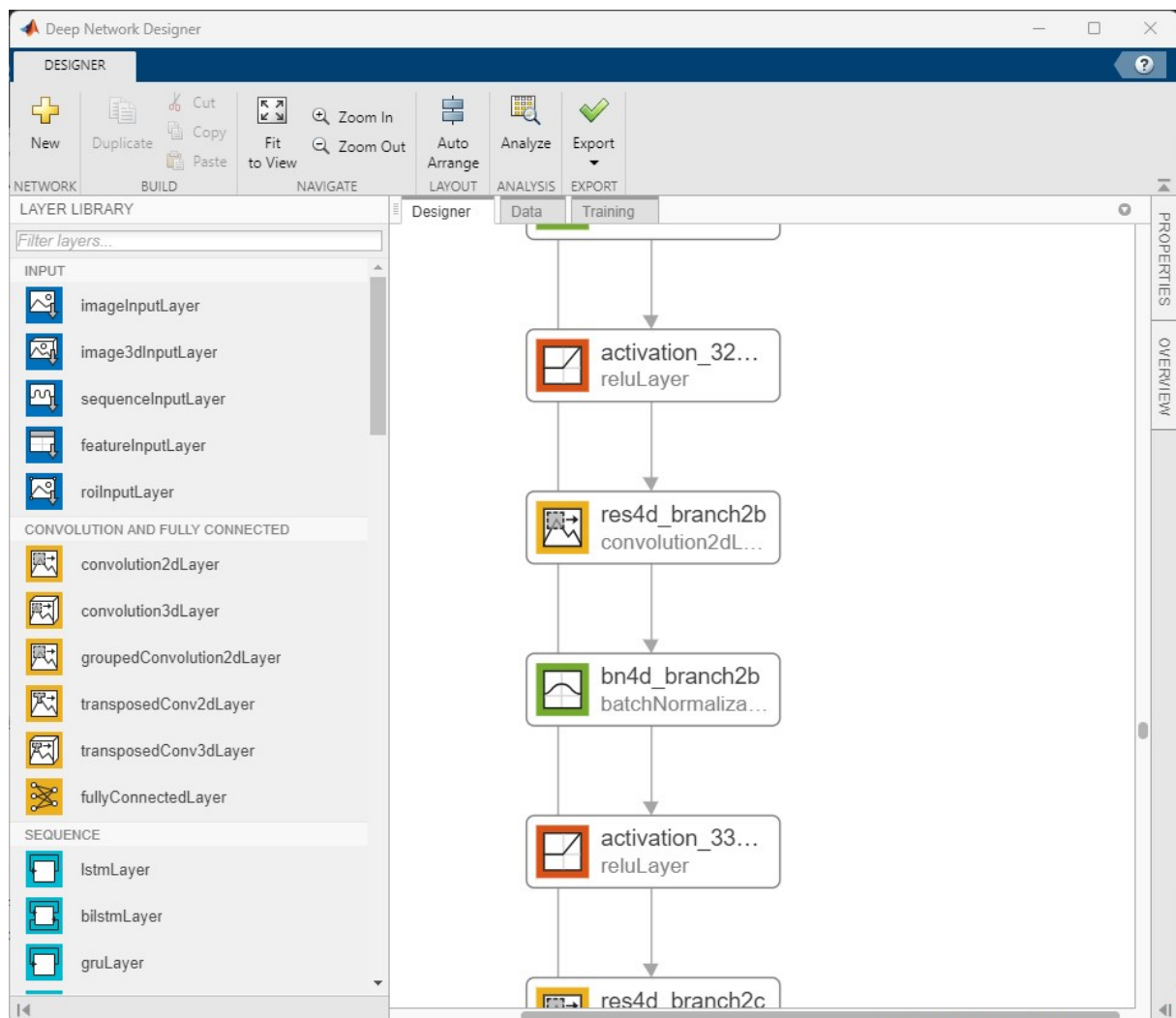


Figura 5.1: Interfaz de la aplicación Deep Network Designer con un fragmento de la red preentrenada ResNet-50.

## 5.2 Base de datos

La base de datos utilizada para el desarrollo de este proyecto ha sido proporcionada por el Instituto Valenciano de Fertilidad (IVI) y está constituida por dos partes principales:

- Un conjunto de 825 imágenes iniciales obtenidas del último fotograma de los vídeos *time-lapse* de las incubadoras de ovocitos en su etapa de blastocisto. Las imágenes son de tamaño 500 x 500 x 3 píxeles.
- Un archivo Excel donde están registrados los identificadores de cada imagen, la clase a la que pertenecen y 4 variables adicionales.

Las imágenes han sido clasificadas en dos grupos atendiendo a si han sido implantados los embriones observados en la imagen o no. Cuando se han implantado se identifican con el valor “100” y en caso contrario con “0”. Por tanto, la clasificación es binaria, solo existen dos posibilidades.

Tras hacer una primera revisión a los datos e imágenes se observó que había algunas de ellas que no tenían la clase definida, por lo que se eliminaron del proyecto. Además, en el archivo Excel había registradas algunas imágenes que no se encontraron, por lo que estos valores también se descartaron para este trabajo. En consecuencia, tras reorganizar la información se obtuvo un resultado final de 788 imágenes con las que trabajar. De este total, 472 son de la clase no implantadas (“0”) y 316 son de embriones que fueron considerados para implantación (“100”). En la Tabla 5.2 se presenta la proporción de un grupo frente al otro.

	<b>Muestras</b>	<b>Porcentaje respecto al total</b>
<b>Implantados</b>	316	40.1 %
<b>No implantados</b>	472	59.9 %
<b>Total</b>	788	

**Tabla 5.2:** Número de imágenes de cada clase en la base de datos.

Por otro lado, en cuanto a las variables numéricas se tiene 4: edad de la paciente, ovodonación, manipulación de ovocitos y día de vitrificación. La edad es una variable cuantitativa discreta que contiene las edades de las mujeres que se han sometido a la técnica de reproducción asistida y se les va a implantar el embrión. En cuanto a la ovodonación, se refiere a si el embrión resultante se ha producido con un óvulo de una donante (ovodonación positiva o “1”) o si el óvulo era de la propia paciente (ovodonación negativa o “0”). Por otra parte, del mismo modo se define la manipulación de ovocitos, “1” si ha sido manipulado y “0” si no lo ha sido. Finalmente, el día de vitrificación se refiere a el día que se congelo el embrión, un embrión que llega a blastocisto en día 5 suele ser mejor que uno que llega en día 7. Los embriones siempre se suelen vitrificar en estadio de blastocisto. En la Tabla 5.3 se tiene se resumen las variables empleadas en el proyecto.

<b>Variables</b>	<b>Tipo de variable</b>	<b>Valores</b>
Edad de la paciente	Cuantitativa discreta	20,21,...,50
Ovodonación	Categórica binaria	0,1
Manipulación de ovocitos	Categórica binaria	0,1
Día de vitrificación	Categórica con 3 valores posibles	5,6,7

**Tabla 5.3:** Características de las variables.

6.1 Diagrama del flujo de trabajo

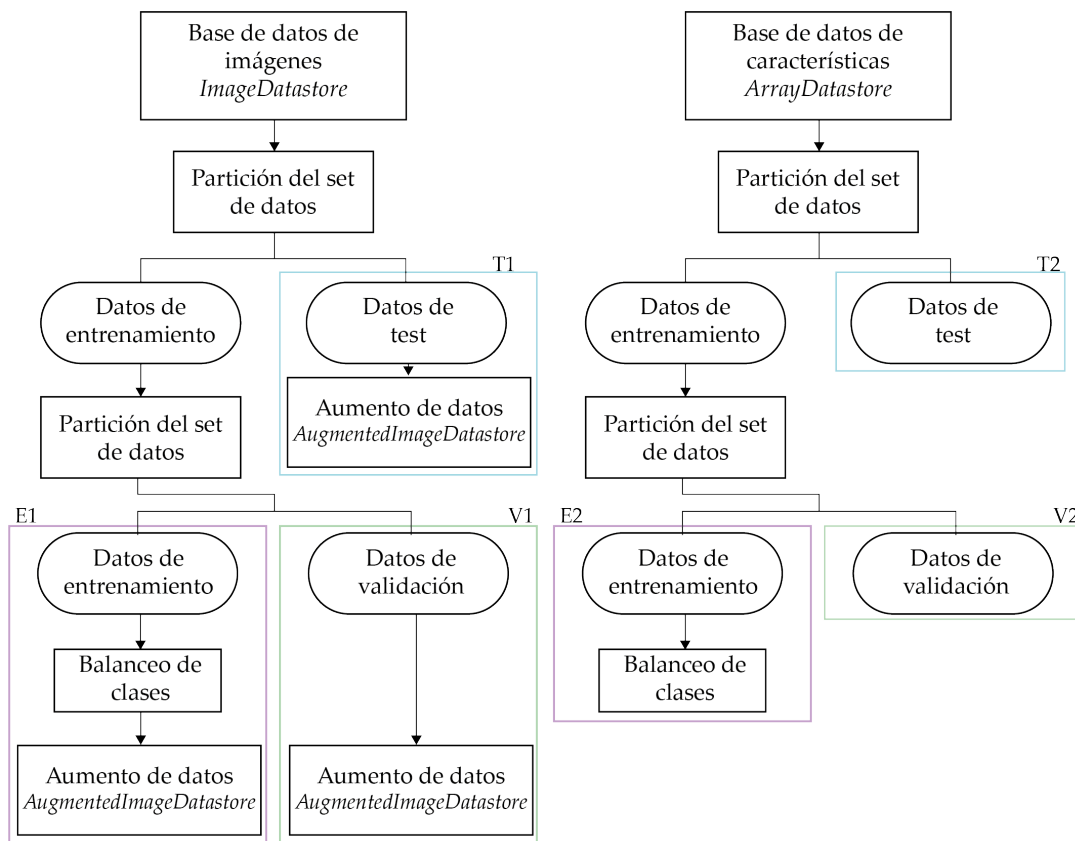
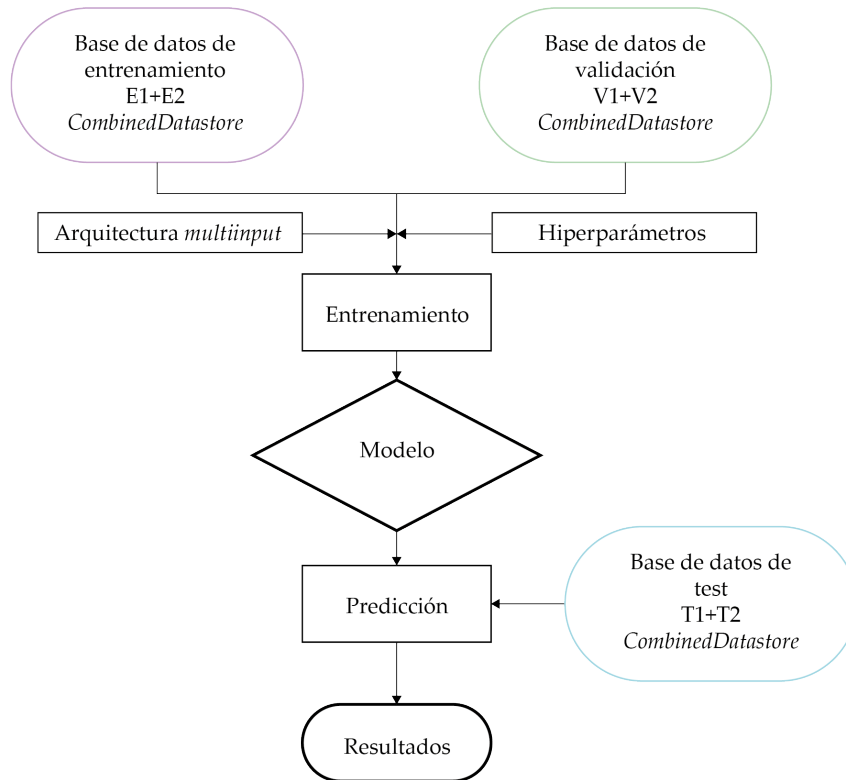


Figura 6.1: Diagrama de trabajo seguido en este TFG, siendo E1 y E2 conjunto de entrenamiento 1 y conjunto de entrenamiento 2, respectivamente. Del mismo modo con T1 y T2 (conjuntos de test 1 y 2;) y V1 y V2 (conjuntos de validación 1 y 2).





**Figura 6.2:** Segunda parte del diagrama de trabajo seguido en este TFG.

Para crear modelos de redes neuronales se necesitan una serie de elementos: tener una base de datos preparados, dividir estos datos en diferentes conjuntos de entrenamiento, validación y test; tener la arquitectura de la red que se va a utilizar y configurar las opciones de entrenamiento. En las Figuras 6.1 y 6.2 se representa el diagrama de flujo seguido en el presente TFG para realizar esta tarea.

En el presente capítulo se explica la metodología empleada hasta conseguir el modelo de clasificación.

## 6.2 Creación y preparación de la base de datos

### 6.2.1 Selección de datos

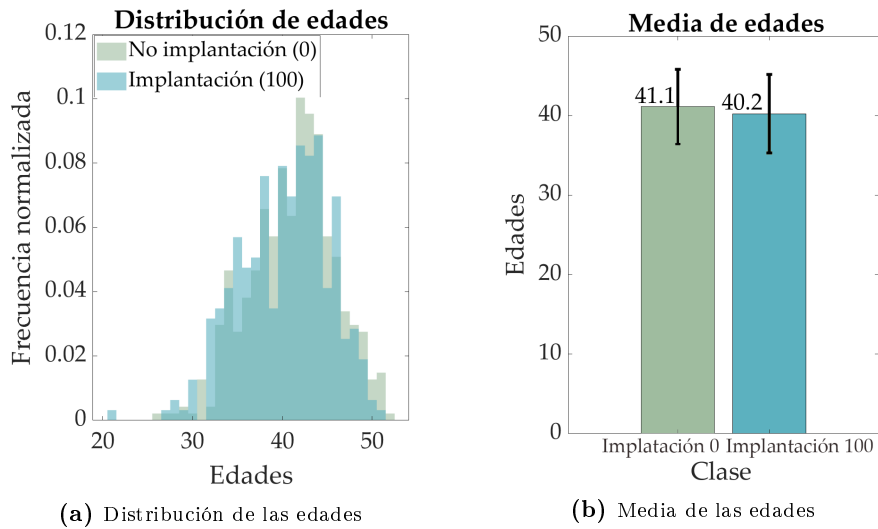
Tras realizar una primera revisión a los datos e imágenes proporcionadas por el Instituto Valenciano de Fertilidad se observó que algunas de las instancias no estaban etiquetadas, por lo que se descartaron para la utilización de este proyecto. En adición, se comparó el archivo Excel donde estaba el registro de todas las muestras con sus diferentes características y la carpeta de imágenes proporcionada, y se seleccionó para este proyecto solo los datos de los cuales se tenían las imágenes.

Tras esta organización inicial, como se comentó en el apartado 5.2, se tiene como resultado para trabajar en este TFG: 788 imágenes y 4 variables numéricas relacionadas con cada una.

### 6.2.2 Análisis exploratorio y estadístico de las variables

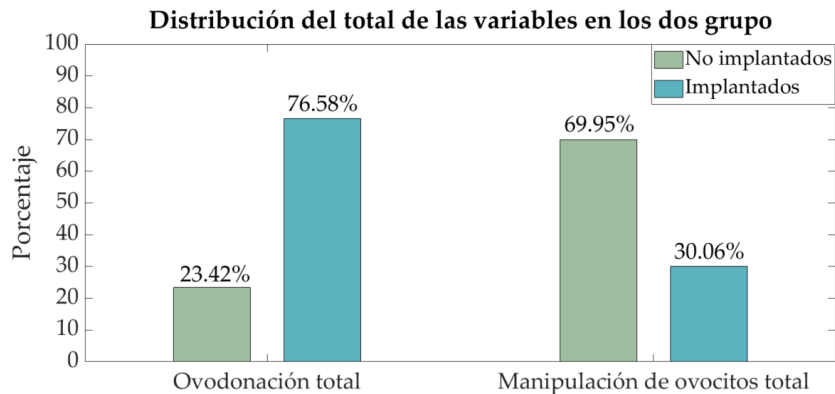
En primer lugar se analizó las variables numéricas para ver si podían ser relevantes en la predicción de la clase, para ello se realizó un análisis exploratorio de los datos y varias pruebas estadísticas.

En cuanto al análisis exploratorio, la primera variable analizada fue la edad de la paciente. En la Figura 6.3 se representa las distribuciones de edad para cada clase así como la media. Se observó que la media y desviaciones típicas de ambos grupos eran parecidas, así como la forma de las distribuciones, por lo que en un primer momento parece no haber diferencias significativas.



**Figura 6.3:** Distribución (6.3a) y medias (6.3b) de las edades de las pacientes sometidas a FIV en cada grupo (No implantación (0) e Implantación (100))

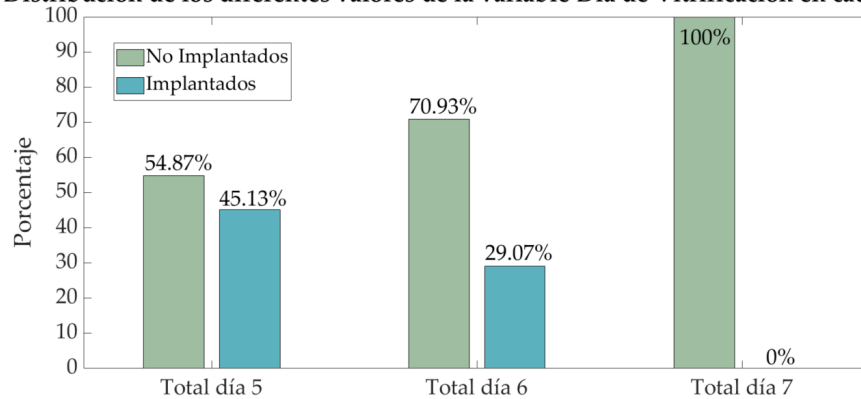
Seguidamente, en la variable Ovodonación se obtuvo que del conjunto de ovodonaciones totales el 76.58 % pertenecían al grupo de Implantados, mientras que el 23.42 % restante al grupo de No Implantación. Por otro lado, en cuanto a la Manipulación de Ovocitos, del total de los manipulados el 69.95 % fueron de la clase Implantación 0, mientras que el 30.06 % de la Implantación 100. Según estos datos, estas dos variables comentadas parecen tener una influencia notoria en la determinación de clase. En la Figura 6.4 se muestra gráficamente estos resultados.



**Figura 6.4:** Distribución del total de Ovodonación y Manejo de Ovocitos en ambos grupos (No implantado o 0 e implantados o 100).

Finalmente, si se distribuye el total de cada valor para la variable Día de Vitrificación entre las dos clases (Figura 6.5) también se observan diferencias significativas. Destaca que de todos los blastocistos vitrificados el día de 6 el 70.93 % pertenecen al grupo de los no implantados, mientras que el 29.07% restante a los implantados. Además, según estos datos, que los blastocistos se vitrifiquen el día 7 es determinante, ya que el 100 % de estas muestras no fueron implantadas.

**Distribución de los diferentes valores de la variable Día de Vitrificación en cada grupo**



**Figura 6.5:** Distribución de los diferentes valores de la variable Día de Vitrificación en cada grupo (No implantado o 0 e implantados o 100).

Tras este primer análisis inicial se aplicaron diferentes test estadísticos que permitiesen cuantificar la magnitud de la relevancia de las variables para la determinación de clases. Los test aplicados y los resultados se resumen en la Tabla 6.1.

Variable	Test aplicado	Hipótesis nula (H0)	p-valor
Edad de la paciente	Test de Wilcoxon-Mann-Whitney	No hay diferencia en las medias de las variables entre los grupos 0 y 100	0,0238
Ovodonación	Test Chi-Cuadrado	No hay asociación entre las variables binarias y los grupos 0 y 100	$6,5680 \times 10^{-161}$
	Test de Fisher		0,0049
Manejo de ovocitos	Test Chi-Cuadrado	No hay asociación entre las variables binarias y los grupos 0 y 100	$6,6235 \times 10^{-161}$
	Test de Fisher		0,0389
Día de vitrificación	ANOVA	No hay diferencia en los valores numéricos de la variable entre los dos grupos	$3,0317 \times 10^{-6}$

**Tabla 6.1:** Test estadísticos aplicados a las diferentes variables y los p-valores obtenidos.

Se tomó como referencia un umbral de significancia  $\alpha = 0,05$  , por lo que todos los p-valores obtenidos se encontraron por debajo de este umbral y, por tanto, se rechazó todas las hipótesis nulas. Con esto se concluye que todas las variables son importantes para la definición de las clases y se añadieron al modelo.

### 6.2.3 Base de datos de imágenes (*ImageDatastore*, *IMDS*)

Se crea una base de datos de imágenes (*imds*) mediante la creación de una *ImageDatastore*. Este objeto crea un almacén de datos que permite leer y procesar los mismos situados en un disco o una ubicación remota. Seguidamente, se añaden las etiquetas al objeto.

```
imds =
  ImageDatastore with properties:
    Files: {
      '/home/ahidbla/documents/data/D2014.11.25_S1089_I106_w1.png';
      '/home/ahidbla/documents/data/D2014.11.28_S1098_I106_w1.png';
      '/home/ahidbla/documents/data/D2014.11.28_S1098_I106_w2.png'
      ... and 785 more
    }
    Folders: {
      '/home/ahidbla/documents/data'
    }
    Labels: [0; 100; 100 ... and 785 more categorical]
    AlternateFileSystemRoots: {}
    ReadSize: 3
    SupportedOutputFormats: ["png" "jpg" "jpeg" "tif" "tiff"]
    DefaultOutputFormat: "png"
    ReadFcn: @readDatastoreImage
```

Figura 6.6: Propiedades del objeto *ImageDatastore* creado.

### 6.2.4 Base de datos de características (*ArrayDatastore*, *FTDS*)

Se obtienen las variables del archivo Excel y se construye una *ArrayDatastore*. Este objeto tiene el mismo funcionamiento que la *ImageDatastore* pero con la diferencia que es un almacén de arreglos o *arrays* en lugar de imágenes.

```
ftds =
  ArrayDatastore with properties:
    ReadSize: 3
    IterationDimension: 2
    OutputType: "cell"
```

Figura 6.7: Propiedades del *ArrayDatastore* creado.

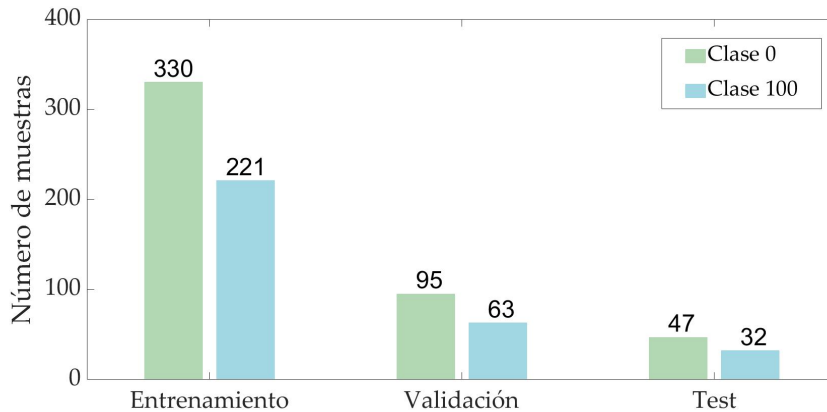
### 6.2.5 Partición en los subconjuntos de entrenamiento, validación y test

Una vez construidas las bases de datos, es necesario dividir las en diferentes conjuntos para poder entrenar las redes neuronales. Lo habitual es hacer tres divisiones, resultando en:

- Un conjunto de entrenamiento. Conjunto con el cual se entrena la red neuronal y se actualizan los pesos.
- Un conjunto de validación. Durante el entrenamiento se comprueba cada cierta iteración la exactitud con el conjunto de validación. Sirve para poder visualizar como está entrenando la red para saber si va por buen camino y ayuda a la optimización de los hiperparámetros.
- Conjunto de test. Una vez acabado el entrenamiento se clasificarán las imágenes del conjunto de test con el modelo obtenido y se obtendrá la exactitud o *accuracy* final.

En primer lugar, se hicieron las particiones de la *imageDatastore* mezclando las muestras aleatoriamente en los tres subconjuntos. Seguidamente, se partió la *arrayDatastore* procurando que se encontrasen las mismas muestras en el mismo orden en cada conjunto que en los de imágenes. En caso contrario, supondría no relacionar bien cada imagen con sus características y se traduciría en graves problemas para el entrenamiento.

La distribución utilizada para dividir los datos fue 70 % entrenamiento, 20 % validación y 10 % test.



**Figura 6.8:** Distribución de muestras en los diferentes conjuntos.

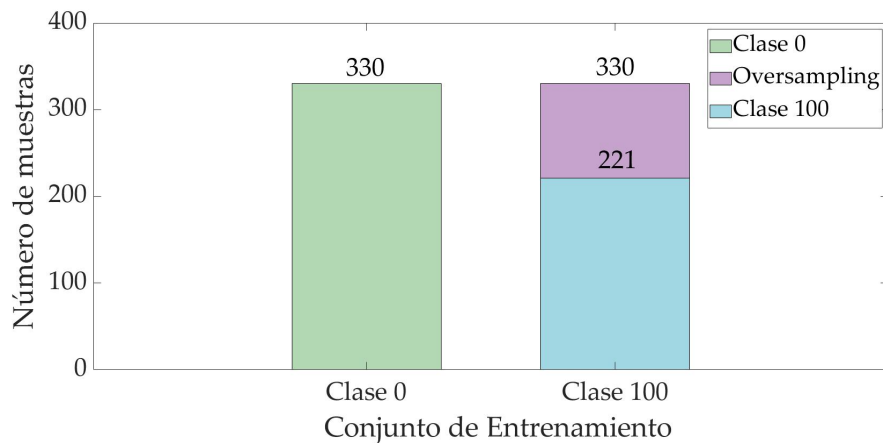
### 6.2.6 Balanceo y aumento de datos

Como se comentó en la sección 5.2, la base de datos tiene más instancias de la clase 0 (no implantado) que de la clase 100 (implantado). En concreto en la tabla 5.2 se puede observar que la proporción es aproximadamente del 60 % frente al 40 %. Esto puede afectar negativamente el rendimiento del modelo durante el entrenamiento por varias razones:

- Modelo sesgado. Un desequilibrio en la distribución de clases puede llevar a que el modelo se sesgue hacia la clase dominante. El modelo puede tener dificultades para aprender patrones y características de la clase minoritaria, ya que se encuentra expuesto a menos ejemplos de esa clase.
- Precisión engañosa. Un modelo puede tener una alta precisión global al predecir correctamente la clase dominante, pero un rendimiento deficiente en la clase minoritaria.
- Reducir el sobreajuste. En escenarios de desequilibrio de clases, es más probable que los modelos sobreajusten o se sobreadapten a la clase dominante. El modelo puede memorizar fácilmente los ejemplos de la clase dominante y no aprender patrones generales que diferencien entre clases.

Por estos motivos, se procede a balancear las clases. Este procedimiento se refiere a igualar la cantidad de muestras de cada clase en el conjunto de entrenamiento. Al proporcionar un conjunto de entrenamiento equilibrado, se permite que el modelo aprenda de manera más efectiva y se logre un mejor rendimiento en la clasificación de todas las clases.

Existen diferentes técnicas para llevar a cabo esta tarea. En el presente trabajo se ha optado por *oversampling*. El *oversampling* es una técnica sencilla que consiste en duplicar instancias de la clase minoritaria hasta igualar las de la clase mayoritaria. Por lo que finalmente el conjunto de entrenamiento queda formado por 330 muestras de la clase 0 y 330 muestras de la clase 100.



**Figura 6.9:** Distribución de muestras en el conjunto de entrenamiento una vez balanceadas las clases.

Por otro lado, un aumento de la base de datos se utiliza para transformar los datos con operaciones como cambio de tamaño, rotaciones, reflexiones en X o Y, cambios de brillo, entre otras. En este caso se ha utilizado con dos finalidades principales:

1. Adaptar el tamaño de las imágenes para que sean compatibles con el tamaño de entrada de la red neuronal. Por este motivo, los tres conjuntos (entrenamiento, validación y test) se han convertido a este tipo de base de datos.
2. Aumentar las imágenes con operaciones de procesamiento aleatorias, en concreto se han añadido rotaciones y reflexiones aleatorias. Esto se hace solo con el conjunto de entrenamiento y permite evitar el sobreajuste y que la red memorice los detalles exactos de las imágenes.

Además, cuando se utiliza un almacén de este tipo en el entrenamiento, en cada época se perturbaban aleatoriamente los datos, de forma que son ligeramente diferentes en cada periodo. Esto le da a la red una falsa sensación de tener más imágenes de las que realmente hay y permite conseguir un mejor entrenamiento con un número limitado de imágenes.

```
imageAugmenter = imageDataAugmenter( ...  
    'RandRotation', [0, 360], ...  
    'RandXReflection', true, ...  
    'RandYReflection', true);
```

**Figura 6.10:** Opciones de aumento de datos introducidas para el conjunto de entrenamiento.

### 6.2.7 Combinación de las dos bases de datos

Una vez llegados a este punto se tienen dos tipos de bases de datos con las mismas muestras para cada conjunto (entrenamiento, validación y test). A continuación, se procede a juntar ambas, con el objetivo de tener todos los datos de una instancia, así como su etiqueta, en una misma base de datos, de forma que todo esté relacionado y se pueda introducir en el entrenamiento.

Para ello se empleó la función *combine* de Matlab y se obtuvo como resultado unas *combinedDatastore*. Por lo que finalmente, se concluye la preparación de los datos listos para el entrenamiento con:

- Conjunto de entrenamiento. *CombinedDatastore* de la *augmentedImageDatastore* de entrenamiento balanceada (E1) con la *arrayDatastore* de entrenamiento balanceada (E2).
- Conjunto de validación. *CombinedDatastore* de la *augmentedImageDatastore* de validación (V1) con la *arrayDatastore* de validación (V2).
- Conjunto de test. *CombinedDatastore* de la *augmentedImageDatastore* de test (T1) con el *arrayDatastore* de test (T2).

	Var1	input	response
1	[37;1;1;6]	224×224×3 uint8	0
2	[30;1;1;5]	224×224×3 uint8	100
3	[41;1;1;6]	224×224×3 uint8	0

Figura 6.11: Primeras instancias de la *combinedDatastore* de entrenamiento.

## 6.3 Entrenamiento

Una vez se tienen los datos preparados, se procede al entrenamiento para conseguir el modelo de clasificación. Para ello es necesario, en primer lugar, disponer de la arquitectura adecuada, eligiendo correctamente las capas de red, cómo se van a conectar, qué funciones de activación tendrán y qué función de pérdidas final. Y en segundo lugar, es necesario buscar los parámetros y opciones de entrenamiento óptimos para obtener los mejores resultados.

### 6.3.1 Arquitectura

En este trabajo, se ha utilizado el enfoque del *transfer learning* como método para el desarrollo del modelo. El *transfer learning* es una técnica de aprendizaje automático que se basa en aprovechar un modelo preentrenado existente como punto de partida; en lugar de entrenar un modelo desde cero, se modifican y adaptan únicamente algunas partes del mismo para el nuevo problema de clasificación. Se parte entonces de un modelo que ha sido entrado con grandes conjuntos de datos y por tanto que ya tiene un cierto conocimiento previo, esto reduce significativamente el tiempo y recursos necesarios para entrenar.

Específicamente, para este proyecto, se han realizado cambios en las capas finales de las redes neuronales utilizadas. Al modificar estas capas, se ajustan a las necesidades y características de

nuestro problema de clasificación de imágenes. Para facilitar esta modificación y adaptación de las capas del modelo, se ha hecho uso de la herramienta Deep Network Designer proporcionada por MathWorks Inc. Las redes modificadas y probadas fueron las que se muestran en la Tabla 6.2.

<b>Red neuronal</b>	<b>Profundidad</b>	<b>Tamaño de entrada de imagen</b>
GoogLeNet	22	224 por 224
SqueezeNet	18	227 por 227
InceptionV3	48	299 por 299
ResNet-18	18	224 por 224
ResNet-50	50	224 por 224
Xception	71	299 por 299
AlexNet	8	227 por 277
VGG-19	19	224 por 224

**Tabla 6.2:** Redes neuronales convolucionales probadas con *transfer learning* para el presente TFG.

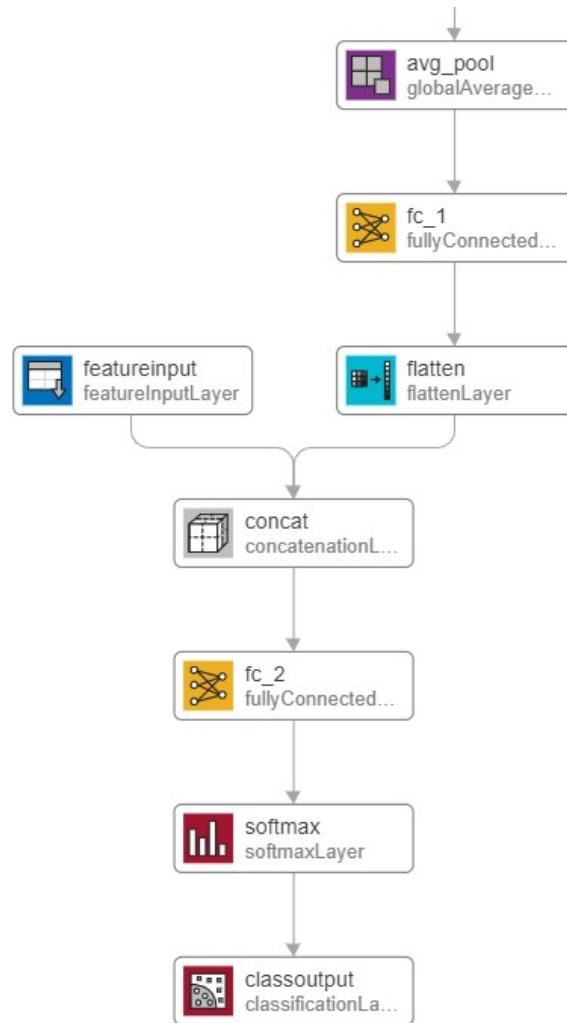
Tras realizar entrenamientos con todas las redes anteriores, solo con datos de imágenes, probando diferentes parámetros de entrenamiento y optimizadores, se decidió elegir ResNet-50 [21] como arquitectura final, puesto que fue la que dio mejores resultados.

Con esto, se volvió a modificar su arquitectura para crear una red neuronal *multiinput*, añadiendo una capa de entrada de características numéricas. Para poder unir estas características con las obtenidas de las imágenes a través de las capas de convolución y *pooling*, se introdujo también una *concatenationLayer*. Además, se necesitó dos capas adicionales previas a unir las características de las imágenes con las de las variables:

- Una capa *fully-connected* tras la última capa de *pooling* para reducir el número de características finales de las imágenes.
- Una *flattenLayer* para aplanar estas características en un vector de una dimensión.

En la Figura 6.12. Se puede ver de forma más visual la nueva organización de las capas finales explicada en el párrafo anterior.





**Figura 6.12:** Opciones de aumento de datos introducidas para el conjunto de entrenamiento.

### *Normalización de los datos de entrada*

La normalización de los datos consiste en reducir estos a rangos más pequeños, en general los más habituales son rangos  $[0,1]$  o  $[-1,-1]$ . De esta forma se consigue que los datos tengan una escala similar, ya que en caso contrario una característica simplemente por tener valores mayores podría dominar a otras y la red la interpretaría como más importante. Además, al utilizar optimizadores de gradiente se facilita que estos encuentren el mínimo global o uno local de forma más rápida.

Para esta tarea concreta se ha optado por la normalización en el rango  $[0,1]$  tanto de las imágenes como de las variables numéricas.

### *Funciones de activación*

En todas las capas intermedias la función de activación empleada es la *ReLU*, mientras que en la capa final se utiliza una *softmax* de modo que además de la predicción se puede saber también las probabilidades con las que la red ha tomado cada decisión. El funcionamiento de ambas funciones ya ha sido descrito en la sección 4.1.3.

### *Función de pérdidas*

La función de pérdidas es uno de los parámetros más importantes que va a indicar cómo de bien está clasificando la red neuronal. Como función de pérdidas para el modelo final se va a elegir la entropía cruzada (Ecuación 4.7), ya descrita en la sección 4.1.5.

### *Arquitectura final empleada*

Como se ha comentado finalmente se construye una red neuronal convolucional *multiinput* a partir de la arquitectura de ResNet-50. Tras todas las modificaciones explicadas anteriormente, las características de esta arquitectura final se resumen en la tabla 6.3.

<b>Nombre</b>	<b>ResNet-50 multiinput</b>
Número de capas	181
Número de conexiones	196
Tipo de entrada	Vector de datos, Imagen
Normalización de las entradas	[0,1], [0,1]
Tipo de salida	Clasificación
Función de pérdidas	Cross-entropy
Funciones de activación internas	ReLU
Función de activación final	Softmax

**Tabla 6.3:** Características principales arquitectura ResNet-50 multiinput creada.

### **6.3.2 Hiperparámetros**

Los hiperparámetros o opciones de entrenamiento son un conjunto de variables ajustables asociadas con el entrenamiento de la red. Estas se configuran por el programador antes de iniciar el proceso de entrenamiento y afectan al rendimiento y capacidad de aprendizaje de la red.

Los hiperparámetros influyen tanto en la arquitectura de la red como en el proceso de entrenamiento y aprendizaje. Por este motivo, se pueden dividir en estas dos categorías:

- Hiperparámetros relacionados con la arquitectura de la red neuronal convolucional:
  1. Número de filtros por capa. El número de filtros por capa determina el número de mapas de activación que se van a obtener en cada una. Permite controlar la cantidad de características que la red puede aprender y extraer, ya que cada filtro extrae características diferentes de la imagen de entrada. A mayor número de filtros la red se

vuelve más robusta para detectar características complejas, pero también aumenta el coste computacional.

2. Tamaño de los filtros. Tamaño de la matriz que se utiliza como filtro para realizar las operaciones de convolución sobre las imágenes. Define el área de muestreo local y la cantidad de información espacial que se toma en consideración. Un menor tamaño de *kernel* detecta características locales, mientras que a mayor tamaño se detectan características más globales de la imagen.
  3. Tamaño de paso. Número de píxeles que se desplaza el *kernel* en cada operación de convolución. Un paso mayor reduce el tamaño de salida y por tanto disminuye el coste computacional, sin embargo, se pierde mayor información espacial.
- Hiperparámetros relacionados con el proceso de aprendizaje:
    1. Optimizador. Algoritmo utilizado para ajustar los pesos de la red tras cada batch y conseguir minimizar la función de costes. Se ha probado con tres diferentes: SGDM, Adam y RMSprop.
    2. Tamaño de *batch*. Como se comentó en la sección 4.1.4, se trata del número de muestras utilizadas para cada actualización de los pesos.
    3. Número de épocas. Número de veces que se recorre todos los datos del conjunto de entrenamiento. Es necesario encontrar el número adecuado para conseguir que la red llegue a su eficiencia pero sin excederse.
    4. Tasa de aprendizaje o *learning rate*. Se trata del porcentaje de cambio que se actualizan los pesos cada vez. Valores elevados llevan a un aprendizaje rápido pero se corre el riesgo de que la red no llegue a converger, mientras que con valores pequeño el aprendizaje es lento y se corre el riesgo de que se quede estancado un en mínimo local.
    5. Frecuencia de validación. Frecuencia con la que se evalúa la precisión del modelo en un conjunto de datos de validación. Permite monitorear el progreso del modelo, así como su capacidad de generalización y ayuda a los programadores a ajustar los hiperparámetros en función de su desempeño.
    6. *Learn rate drop factor* y *learn rate drop period*. Se utilizan para ajustar la tasa de aprendizaje durante el entrenamiento de forma que esta no permanezca constante. Con *learn rate drop factor* se introduce por cuanto se va a multiplicar la *learning rate* inicial y con *learn rate drop period* se dice cada cuanto.
    7. Regularización L2. Ayuda a controlar y evitar el sobreajuste y mejora la generalización al añadir penalización en valores de ponderación extremos. Se trata de un término adicional que se suma en la función de pérdidas y que es la suma de los cuadrados de todos los pesos del modelo multiplicado por un factor de regularización. Este factor es el que se va a manipular.

La elección de estas opciones de entrenamiento es un proceso de prueba y error, donde se van probando diferentes valores y buscando los óptimos para conseguir el mejor entrenamiento.

## 6.4 Validación cruzada: *k-fold cross-validation*

Finalmente, para intentar mejorar la precisión del modelo obtenido previamente se utilizó la técnica de *k-fold cross-validation* o validación cruzada. Esta técnica consiste en dividir los datos de entrenamiento en  $k$  conjuntos diferentes. De estos  $k$  conjuntos,  $k-1$  se utilizan para entrenar y el restante para evaluar el entrenamiento (validación). Este proceso se repite  $k$  veces, eligiendo un subconjunto diferente para la evaluación cada vez y utilizando el resto de entrenamiento.

Las razones por las cuales se hace la validación cruzada en este momento del proceso son:

- Mejorar los resultados. Al variar el conjunto de validación y los de entrenamiento se busca la combinación óptima que mejor permite entrenar al modelo.
- Coste computacional. Esta técnica requiere entrenar  $k$  veces en lugar de una, lo que supone un mayor coste computacional que entrenar solo una vez. Por este motivo se ha decidido utilizar al final en lugar de entrenar desde el principio de esta forma, ya que en este punto se sabe por dónde se sitúan los parámetros óptimos y solo se necesita hacer algunas pruebas finales.

En concreto se ha de utilizado un *cross-validation* con  $k=4$ . Se ha dividido el conjunto de entrenamiento en 4 subconjuntos y se han obtenido 4 modelos.



## Capítulo 7

# Resultados

El proceso de entrenamiento de una red conlleva hacer múltiples pruebas en busca de los hiperparámetros que proporcionen los mejores resultados. Para la elección de la arquitectura base se adaptaron varias de las más comunes para clasificación de imágenes (Tabla 6.2) y se obtuvo unos primeros resultados que permitieron escoger una de ellas, la cual se utilizaría como punto de partida para la construcción de la red *multiinput*.

Para comparar los resultados, se evaluó las diferentes redes en condiciones parecidas y se calcularon métricas como el *accuracy* o exactitud de clasificación, las matrices de confusión y los mapas de sensibilidad de oclusión. Durante esta etapa, los mapas de sensibilidad resultaron especialmente relevantes para la toma de decisiones. En la Figura 7.1, se muestran los resultados comparativos entre Squeezenet y ResNet-50 para los mismos hiperparámetros. Los resultados obtenidos con diferentes conjuntos de parámetros mostraron una tendencia similar, llevando a la conclusión de que ResNet-50 parecía captar información más significativa acerca de las imágenes, en contraste con las otras redes que basaban sus predicciones en partes de la imagen no relevantes.

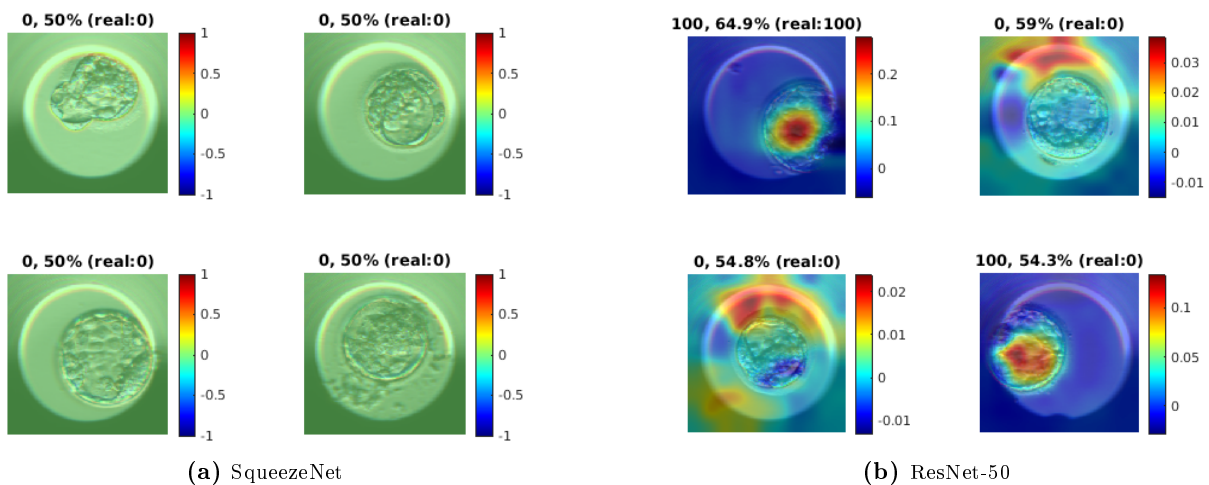
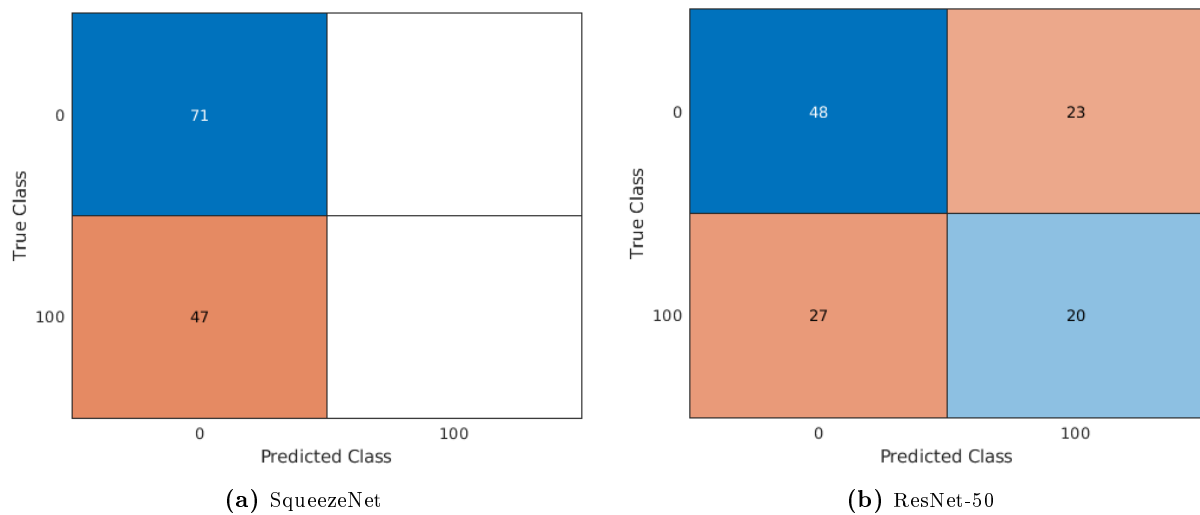


Figura 7.1: Comparativa entre *Occlusion sensitivity maps* de SqueezeNet y ResNet-50.

En las matrices de confusión también se puede observar estos resultados. Se observa en la Figura 7.2 lo que ocurre cuando la red no consigue aprender. La red está sesgada y etiqueta todas las muestras como 0, ya que se trata de la clase mayoritaria. Esto ocurre porque encontrar que clasificar todas las imágenes en la clase mayoritaria proporciona una precisión aparentemente alta (alrededor del 60 %). Esto puede generar problemas para identificar correctamente la clase minoritaria y, por tanto, a pesar de proporcionar estas redes los mejores resultados iniciales de *accuracy* se optó por descartarlas.



**Figura 7.2:** Comparativa entre los resultados de matrices de confusión de SqueezeNet y ResNet-50.

Tras estos resultados, se decidió optar por ResNet-50, como ya se comentó en el capítulo anterior. Sin embargo, a pesar de ser la mejor red, los resultados obtenidos hasta el momento no fueron demasiado satisfactorios, puesto que las *accuracys* eran bajas. En la tabla 7.1 se tienen algunos de los valores de *accuracy* obtenidos para diferentes hiperparámetros.

Optimizador	Tamaño de Batch	Épocas totales	Learning rate inicial	Learning rate drop period	Learning rate drop factor	Frecuencia de validación	Accuracy en validación	Accuracy en test
adam	110	20	0,0001	20	0,1	3	61,344	62,712
adam	54	10	0,01	-	-	7	21,849	61,017
adam	110	20	0,0001	5	0,1	3	59,664	59,322
sgdm	110	30	0,0001	20	0,1	3	57,143	58,474
sgdm	110	50	0,001	20	0,1	3	61,344	56,779

**Tabla 7.1:** Resultados de ResNet-50 para diferentes hiperparámetros.

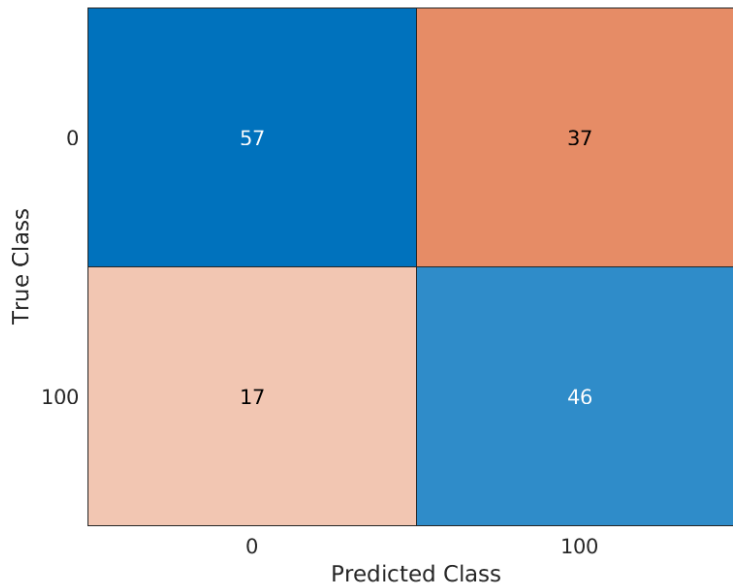
Es en este punto donde se decidió construir una red *multiinput* para añadir las características numéricas y ver si mejoraban el proceso de clasificación. Tras varios procesos de prueba los mejores resultados se encontraron en torno a los siguientes valores de parámetros:

Optimizador	Tamaño de Batch	Épocas totales	Learning rate inicial	Learning rate drop period	Learning rate drop factor	Frecuencia de validación
adam	110	40	0,001	30	0,1	6
<b>Accuracy en validación</b>			<b>61,392</b>	<b>Accuracy en test</b>		<b>64,557</b>

**Tabla 7.2:** Hiperparámetros y *accuracy* óptimos encontrados para la red *multiinput* desarrollada en el presente TFG.

Como se observa en la Tabla 7.2, el *accuracy* máximo conseguido es del 64,557%. Este resultado sigue lejos de lo que podría considerarse satisfactorio. Por este motivo, se optó a continuación, por utilizar la técnica de validación cruzada (*k-fold cross-validation*) para probar si distintos conjuntos de entrenamiento y validación podían mejorar los resultados. Esta técnica es de especial utilidad en bases de datos pequeñas, ya que permite aprovechar al máximo los datos disponibles y alivia la dependencia entre la separación de los conjuntos de entrenamiento y validación. Al pasar todos los datos por cada conjunto, se obtiene una evaluación más sólida del rendimiento del modelo, ya que se utiliza todo el conjunto de datos para entrenar y validar.

Finalmente, el mejor resultado obtenido fue del 65,605 % de *accuracy* en test. En la matriz de confusión se puede observar de forma visual la cantidad de acierto y error del modelo. Se puede ver que a pesar de clasificar bien la mayoría de las muestras todavía existe un porcentaje muy elevado de instancias mal clasificadas, sobre todo de la clase 0. En la Tabla 7.3 se detallan los resultados numéricos.



**Figura 7.3:** Matriz de confusión del modelo final.

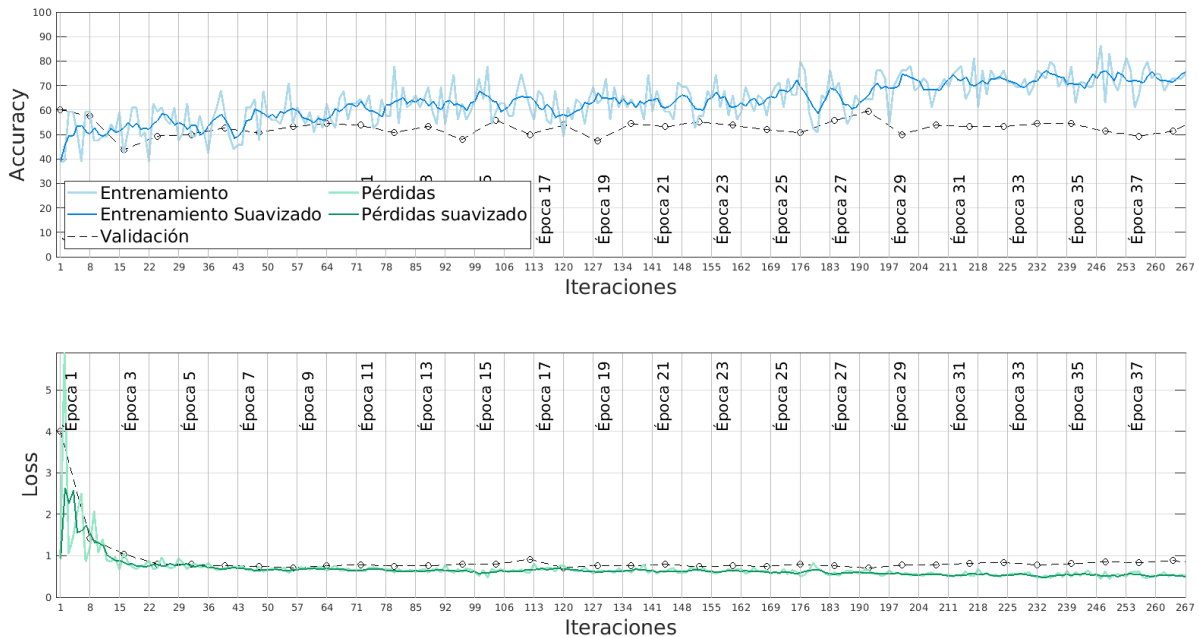


Clase	Sensibilidad	Especificidad	Precisión	F1-score	AUC	Accuracy
Clase 0	0,6064	0,7302	0,7703	0,6786	0,6272	0,6561
Clase 100	0,7302	0,6064	0,5542	0,6301		
<b>Media</b>	0,6683	0,6683	0,6622	0,6544		

**Tabla 7.3:** Métricas de la matriz de confusión del modelo final.

Se observa que el modelo es ligeramente más sensible en la detección de implantaciones positivas (clase 100) que en las negativas, esto sugiere que tiene una mayor capacidad para detectar correctamente las implantaciones positivas. La sensibilidad se enfoca en cómo de bueno es el modelo para encontrar los casos positivos. En cuanto a la precisión, esta se centra en qué tan preciso es el modelo al afirmar que una instancia es positiva. El modelo clasifica correctamente alrededor del 77,03 % de los casos de la clase 0, sin embargo, con la clase 100 lo hace correctamente alrededor del 55,42 % de los casos de esa clase. Finalmente, el área bajo la curva o *area under de curve* (AUC) del 0,6272 se encuentra lejos del 1 ideal e indica que el modelo tiene una capacidad moderada para discriminar entre las clases. Por tanto, de todo esto se puede concluir que el modelo detectar correctamente la mayoría de los casos implantados, sin embargo, al ser la precisión baja, se deduce que también clasifica incorrectamente muchos casos de no implantados como implantados (generando bastantes falsos positivos).

### CURVAS DE ENTRENAMIENTO Y PÉRDIDAS



**Figura 7.4:** Curvas de entrenamiento y pérdidas del proceso de entrenamiento del modelo final.

Por otro lado, las curvas de entrenamiento y pérdidas (Figura 7.4) denotan un poco de *overfitting*. Esto se puede deber a que el modelo se ajusta demasiado al entrenamiento y luego no es capaz de generalizar para nuevas muestras. De nuevo, en este punto tiene mucha importancia disponer de bases de datos grandes. Al tener de una base de datos pequeña, es posible que esta no contenga suficientes ejemplos como para representar todas las variaciones y casos posibles que pueden

surgir en el mundo real. Esto limita la capacidad del modelo para aprender patrones generales y se centra en aprender patrones más específicos, por lo que acaba sobreajustándose a los datos de los que dispone.

Los resultados parecen no ser muy satisfactorios, sin embargo, cabe destacar que muestran cierta esperanza a futuro, puesto que si se consigue aumentar la base de datos es muy probable que se consiga mejorar la exactitud del modelo y su performance; y de esta forma obtener un modelo que pueda aplicarse a la práctica clínica.



# Conclusiones, limitación y líneas futuras

En el presente trabajo de fin de grado, se ha construido un modelo de inteligencia artificial basado en redes neuronales convolucionales para la clasificación de embriones descongelados en su etapa de blastocisto. El principal objetivo fue determinar el potencial de implantabilidad de los mismos para su utilización en técnicas de reproducción asistida. Con esta finalidad se ha construido una arquitectura *multiinput* y se ha variado los parámetros de entrenamiento hasta conseguir el modelo con la mayor precisión y exactitud posible.

En primer lugar, se probó diferentes arquitecturas utilizando la técnica de *transfer learning* para clasificación de las imágenes y se determinó cuál de ellas proporcionaba mejores resultados. Con todo ello, se escogió una de ellas como base para crear un modelo de clasificación *multiinput*. Seguidamente, se probaron múltiples entrenamientos, determinando cuáles eran los parámetros óptimos que proporcionaban los mejores resultados y, finalmente, se aplicó la técnica de *k-fold cross-validation* para intentar mejorar los mismos probando con diferentes conjuntos de entrenamiento y validación.

Tras todo ello, el nivel de exactitud final en la clasificación obtenido fue de 65,605 %. Si bien este resultado puede considerarse bajo en comparación a otros trabajos similares donde se trabaja con inteligencia artificial para la clasificación de blastocistos, es importante destacar que la clasificación mediante imágenes es un desafío complejo y todavía poco explorado.

Sin embargo, se destaca que todavía hay un amplio margen de mejora. Durante el desarrollo del proyecto se han identificado algunas limitaciones que influyeron en el rendimiento del modelo, como el número de imágenes y datos disponibles o el desequilibrio entre la distribución de clases. Sin duda alguna, el número de imágenes utilizadas para el entrenamiento es crucial al emplear técnicas de inteligencia artificial. Los grandes modelos de clasificación más famosos han sido entrenados con miles de imágenes para obtener las exactitudes de las que disponen. La inteligencia artificial, de igual forma que la inteligencia humana, mejora su aprendizaje a través de la experiencia que conlleva la observación de múltiples ejemplos. Aumentando el número de imágenes

en futuros estudios se podría mejorar no solo la exactitud del modelo, sino otros aspectos como la precisión y robustez del mismo.

Además, cabe destacar que ya inicialmente se partía de una tarea complicada, puesto que las imágenes eran de embriones descongelados cuando normalmente partiendo de embriones sin congelar el proceso ya es complejo. Por otro lado, cabe destacar que no se han encontrado en el estado del arte estudios específicos como este para la selección embrionaria con imágenes post-desvitrificación. En adición, es de resaltar que en el IVI se probó con diferentes algoritmos de clasificación de las imágenes de embriones descongelados, como el algoritmo LifeWhisperer, y se obtuvo un AUC de 0,64 y de 0,66 cuando se añadían parámetros, resultados parecidos a los del presente TFG. Por tanto, a pesar de las limitaciones mencionadas, en este trabajo se presenta la base para futuros estudios en la clasificación de blastocistos desvitrificados por inteligencia artificial. Se proponen varias direcciones de investigación para mejorar el rendimiento del modelo, como expandir el conjunto de datos de entrenamiento con imágenes adicionales, utilización de otras técnicas de balanceo de clases, continuar en la búsqueda de los hiperparámetros más adecuados o probar con otras arquitecturas.

En conclusión, en este proyecto de trabajo de final de grado se ha demostrado el potencial de las CNNs para la clasificación de blastocistos y la predicción de su potencial de implantación. Aunque los resultados obtenidos hasta el momento presentan un nivel de precisión moderado, es fundamental continuar investigando y mejorando los enfoques utilizados para lograr una clasificación más precisa y confiable. La combinación de medicina reproductiva e inteligencia artificial tiene el potencial de transformar y optimizar la selección embrionaria, beneficiando tanto a los profesionales embriológicos, que necesitarían dedicar un menor tiempo a esta tarea, como a las pacientes de estos tratamientos.

Parte II

# Presupuesto



## Capítulo 9

# Presupuesto

El presente documento tiene como objetivo realizar una valoración económica del trabajo realizado en este proyecto TFG titulado “Diseño y desarrollo de un sistema para predecir el potencial de implantación embrionaria en blastocistos desvitrificados”. El presupuesto se ha estructurado en dos partes principales: los presupuestos parciales y el presupuesto final. A su vez, el informe de presupuestos parciales se ha dividido en tres cuadros de precios: mano de obra, maquinaria y materiales.

### 9.1 Presupuestos parciales

#### 9.1.1 *Mano de obra*

En esta sección se describen los recursos humanos necesarios para el desarrollo del presente TFG. Se realiza una estimación del tiempo dedicado por la tutora, cotutora y autora. Se considera la contribución de: D<sup>a</sup>. Valery Naranjo Ornedo, como directora del proyecto y tutora; D<sup>a</sup>. Elena Payá Bosch, como cotutora del proyecto; y D<sup>a</sup>. Ayelén Hidalgo Blasco, como estudiante del Grado de Ingeniería Biomédica y autora del proyecto.

Denominación	Uds	Precio Unitario (€)	Cantidad	Total (€)
Tutora (Catedrática)	h	42,00	16	672,00
Cotutora (Doctorando)	h	17,20	36	619,20
Autora (Estudiante GIB)	h	13,80	300	4.140,00
			<b>Total</b>	<b>5.431,00</b>

**Tabla 9.1:** Cuadro de precios de la mano de obra.



### 9.1.2 Maquinaria

En esta sección se detallan los costes de maquinaria, en concreto, en el caso de este proyecto corresponden con los recursos de *software* y *hardware* utilizados.

Como software se ha utilizado MATLAB® y su aplicación Deep Network Designer. Por otro lado, para redacción de la memoria se ha utilizado el lenguaje de código abierto LaTeX con el editor TeXstudio.

Denominación	Uds	Cantidad	Precio Unitario (€)	Periodo de amortización (años)	Intervalo amortizado (meses)	Total (€)
Licencia MATLAB®	u	1	800,00	1	6	400,00
Deep Network Desginer	u	1	400,00	1	6	200,00
TeXstudio	u	1	0,00	1	2	0,00
<b>Total</b>						<b>600,00</b>

En cuanto al hardware, el TFG se ha desarrollado haciendo uso de un ordenador personal MSI PS63 Modern 8RC, de un servidor como almacén de datos y de una GPU para realizar los cálculos de entrenamiento.

Denominación	Uds	Cantidad	Precio Unitario (€)	Periodo de amortización (años)	Intervalo amortizado (meses)	Total (€)
MSI PS63 Modern 8RC	u	1	1500,00	5	8	200,00
NVIDIA Titan Xp	u	1	1349,00	4	5	140,52
Servidor Synology CVBLab	u	1	3500,00	4	6	437,50
<b>Total</b>						<b>778,02</b>

### 9.1.3 Materiales

En esta sección se detallan los costes de materiales, en este caso, corresponden a los costes de obtención de las imágenes a través de las incubadoras. En concreto la incubadora utilizada es Embryoscope+® y también tiene asociados unos costes de mantenimiento.

Denominación	Uds	Cantidad	Precio Unitario (€)	Periodo de amortización (años)	Intervalo amortizado (meses)	Total (€)
Incubadora Embryoscope+®	u	1	120.000,00	10	1	1.000,00
Mantenimiento	u	1	6.000,00	1	1	500,00
<b>Total</b>						<b>1500,00</b>

## 9.2 Presupuestos totales

Para el cálculo del presupuesto total se debe tener en cuenta los presupuestos parciales calculados anteriormente. Además, es necesario añadir los gastos generales y el asociado al beneficio industrial, así como el impuesto del IVA. Con todo ello se obtiene el presupuesto total que supone la realización de este proyecto TFG.

<b>CAPÍTULOS</b>	<b>IMPORTE (€)</b>
1. Coste mano de obra	5.431,00
2. Coste maquinaria	1.378,02
3. Coste materiales	1500,00
<b>PRESUPUESTO DE EJECUCIÓN DE MATERIAL</b>	<b>8.309,02</b>
13 % de gastos generales	1.080,17
6 % de beneficio industrial	498,54
<b>PRESUPUESTO DE EJECUCIÓN POR CONTRATA</b>	<b>9.887,73</b>
21 % IVA	2.076,42
<b>PRESUPUESTO TOTAL</b>	<b>11.964,15</b>



# Bibliografía

- [1] American Society for Reproductive Medicine. *Assisted reproductive technologies: a guide for patients*. 2019. URL: <https://www.reproductivefacts.org/globalassets/rf/news-and-publications/bookletsfact-sheets/english-fact-sheets-and-info-booklets/art-booklet2.pdf> (vid. págs. 3-5).
- [2] Organización Mundial de la Salud. *Infertility*. 2021. URL: <https://www.who.int/es/health-topics/infertility> (visitado 11-05-2023) (vid. pág. 3).
- [3] Instituto Nacional de Estadística (INE). *Encuesta de Fecundidad de 2018*. [Último acceso: 9 mayo 2023]. 2018 (vid. págs. 3, 4).
- [4] Sociedad Española de Fertilidad. *Guía de la Sociedad Española de Fertilidad para pacientes con problemas de fertilidad*. [Accedido el 9 de mayo de 2023]. 2014 (vid. pág. 4).
- [5] Instituto Bernabeu. *Diferencias entre fecundación in vitro (FIV) e inseminación artificial (IA)*. 2021. URL: <https://www.institutobernabeu.com/es/foro/diferencias-fecundacion-in-vitro-fiv-e-inseminacion-artificial-ia/> (vid. págs. 4, 5).
- [6] Reproducción Asistida ORG. *Punción ovárica: ¿qué es y cómo se hace?* s.f. URL: <https://www.reproduccionasistida.org/puncion-ovarica/#que-es-la-puncion-ovarica> (vid. pág. 5).
- [7] Reproducción Asistida ORG. *El proceso de la fecundación in vitro paso a paso*. s.f. URL: <https://www.reproduccionasistida.org/el-proceso-de-la-fecundacion-in-vitro/#comprobar-la-fecundacion> (vid. pág. 6).
- [8] M Peigné y S Epelboin. “Reproducción asistida”. En: *EMC-Tratado de Medicina* 22.4 (2018), págs. 1-11 (vid. pág. 6).
- [9] Keith L Moore, Trivedi Vidhya Nandan Persaud y Mark G Torchia. *Embriología clínica*. Elsevier, 2020 (vid. pág. 6).

- [10] 2021. URL: <https://www.reproduccionasistida.org/desarrollo-embrionario/> (vid. pág. 6).
- [11] Reproducción Asistida ORG. *Estructura del embrioblasto*. Imagen. 2023. URL: <https://www.reproduccionasistida.org/clasificacion-embriones-calidad/estructura-del-blastocisto/> (vid. pág. 7).
- [12] M Hurtado de Mendoza et al. “II Criterios ASEBIR de valoración morfológica de oocitos, embriones tempranos y blastocistos humanos”. En: *Cuad. Embriol.* (2015) (vid. págs. 7, 8).
- [13] Irene Rubio et al. “Clinical validation of embryo culture and selection by morphokinetic analysis: a randomized, controlled trial of the EmbryoScope”. En: *Fertility and Sterility* 102.5 (2014), 1287-1294.e5. ISSN: 0015-0282. DOI: <https://doi.org/10.1016/j.fertnstert.2014.07.738>. URL: <https://www.sciencedirect.com/science/article/pii/S001502821401365X> (vid. pág. 13).
- [14] Raquel Del Gallego, José Remohí y Marcos Meseguer. “Time-lapse imaging: the state of the art†”. En: *Biology of Reproduction* 101.6 (feb. de 2019), págs. 1146-1154. ISSN: 0006-3363. DOI: 10.1093/biolre/iox035. eprint: <https://academic.oup.com/biolreprod/article-pdf/101/6/1146/31673666/iox035.pdf>. URL: <https://doi.org/10.1093/biolre/iox035> (vid. pág. 14).
- [15] Marcos Meseguer et al. “The use of morphokinetics as a predictor of embryo implantation†”. En: *Human Reproduction* 26.10 (ago. de 2011), págs. 2658-2671. ISSN: 0268-1161. DOI: 10.1093/humrep/der256. eprint: <https://academic.oup.com/humrep/article-pdf/26/10/2658/1680287/der256.pdf>. URL: <https://doi.org/10.1093/humrep/der256> (vid. pág. 14).
- [16] Qi Zhang et al. “Development of deep learning algorithms for predicting blastocyst formation and quality by time-lapse monitoring”. En: *Communications Biology* 4.1 (2021), pág. 415. DOI: 10.1038/s42003-021-01937-1 (vid. pág. 14).
- [17] Robert Milewski et al. “How much information about embryo implantation potential is included in morphokinetic data? A prediction model based on artificial neural networks and principal component analysis”. En: *Advances in Medical Sciences* 62.1 (2017), págs. 202-206. ISSN: 1896-1126. DOI: <https://doi.org/10.1016/j.advms.2017.02.001>. URL: <https://www.sciencedirect.com/science/article/pii/S1896112617300251> (vid. pág. 14).
- [18] David E Rumelhart, Geoffrey E Hinton y Ronald J Williams. “Learning representations by back-propagating errors”. En: *nature* 323.6088 (1986), págs. 533-536 (vid. pág. 20).
- [19] *MATLAB*. Accedido el 17 de mayo de 2023. URL: <https://es.mathworks.com/products/matlab.html> (vid. pág. 26).

- [20] *Deep Network Designer App*. Accedido el 17 de mayo de 2023. URL: <https://es.mathworks.com/help/deeplearning/ref/deepnetworkdesigner-app.html> (vid. pág. 26).
- [21] Kaiming He et al. “Deep Residual Learning for Image Recognition”. En: *arXiv preprint arXiv:1512.03385* (2015). DOI: [10.48550/arXiv.1512.03385](https://doi.org/10.48550/arXiv.1512.03385) (vid. pág. 37).

