

MÁSTER EN TRATAMIENTO ESTADÍSTICO COMPUTACIONAL
DE LA INFORMACIÓN

TRABAJO DE FIN DE MÁSTER

Desarrollo de algoritmos de procesamiento de imagen para
el análisis de estructuras macromoleculares por microscopía
electrónica

Adrián Sansiñena Rodríguez



UNIVERSIDAD COMPLUTENSE DE MADRID
FACULTAD DE CIENCIAS MATEMÁTICAS

UNIVERSIDAD POLITÉCNICA DE MADRID
ESCUELA TÉCNICA SUPERIOR DE INGENIEROS DE
TELECOMUNICACIÓN

Tutor: Carlos Óscar Sorzano Sánchez

Ponente: Ignacio Villanueva Díez

Madrid, 2023

Índice

1. Introducción	6
1.1. Motivación	9
2. Aprendizaje profundo	10
2.1. Redes Neuronales Convolucionales	10
2.2. Redes Residuales	12
3. Rotaciones	14
3.1. Ángulos de Euler	14
3.2. Cuaterniones	15
3.3. Continuidad	17
3.4. Simetrías	18
4. Datos y Software	24
4.1. Datos	24
4.2. Software Scipion	25
5. Centrado de partículas	27
5.1. Metodología	27
5.1.1. Datos	27
5.1.2. Arquitectura de la red	27
5.1.3. Entrenamiento	30
5.1.4. Transfer Learning	30
5.1.5. Consenso de modelos	31
5.2. Resultados	31
6. Estimación angular	34
6.1. Metodología	34
6.1.1. Datos	34
6.1.2. Arquitectura de la red	35
6.1.3. Distancia angular	35
6.1.4. Entrenamiento	36
6.1.5. Consenso de modelos	37
6.1.6. Moléculas con simetría	37
6.2. Resultados	38
6.2.1. Representación angular	38

6.2.2. Moléculas sin simetría	39
6.2.3. Moléculas con simetría	44
7. Conclusiones	49

Resumen

La criomicroscopía electrónica permite obtener imágenes de macromoléculas a partir de las cuales es posible reconstruir su estructura tridimensional. Para ello, es necesario llevar a cabo el alineamiento de las imágenes, lo cual consiste en reconocer la dirección de proyección de la macromolécula en cada imagen. Se han aplicado técnicas de aprendizaje profundo para llevar a cabo esta tarea, primero tratando de estimar la localización del centro de la molécula en la imagen y luego estimar su orientación. Se analizan las distintas complicaciones que han surgido durante el desarrollo de este trabajo, haciendo hincapié en la importancia de emplear una correcta representación angular y el tratamiento especial que requieren las macromoléculas con simetrías.

Abstract

Electron cryomicroscopy allows obtaining images of macromolecules from which it is possible to reconstruct their three-dimensional structure. To do this, it is necessary to perform image alignment, which involves recognizing the projection direction of the studied macromolecule in each image. Deep learning techniques have been applied to carry out this task, first by trying to estimate the location of the molecule's center in the image and then estimating its orientation. Various complications that have arisen during the development of this work are analyzed, emphasizing the importance of using a correct angular representation and the special treatment required for symmetrical macromolecules.

1. Introducción

La criomicroscopía electrónica (cryo-EM) es una técnica de microscopía electrónica utilizada en biología estructural para visualizar muestras biológicas tales como proteínas, complejos de proteínas o virus a una escala molecular [1]. La muestra a estudiar se congela rápidamente a temperaturas de hasta -190°C con el fin de preservar la estructura natural de la muestra. La muestra congelada es iluminada con un haz de electrones y la difracción resultante es capturada formando una imagen 2D de alta resolución, figura 1.

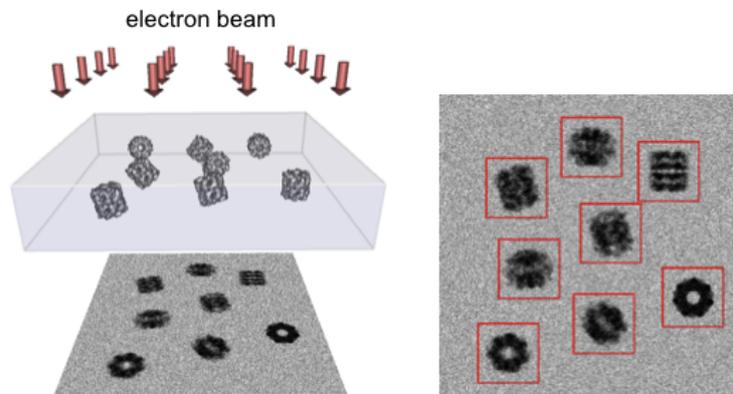


Figura 1: Esquema de la obtención de imágenes por un microscopio electrónico. Un haz de electrones incide sobre la muestra con las moléculas a estudiar. En la micrografía resultante aparecen distintas proyecciones de estas moléculas. Imagen extraída de <https://people.csail.mit.edu/gdp/cryoem.html>

Esta técnica se ha convertido en uno de los métodos principales para poder determinar la estructura 3D de macromoléculas [2], lo que es fundamental para comprender cómo funcionan en los procesos biológicos y cómo interactúan con otras moléculas. Por ejemplo, conocer la estructura tridimensional de una proteína puede proporcionar información importante sobre cómo se une a su sustrato, lo que puede ser utilizado para diseñar fármacos que se unan a la proteína de manera selectiva e inhiban su actividad.

Como resultado del proceso de preparación de las muestras, en una única imagen o micrografía del microscopio electrónico aparecen muchos especímenes o partículas de la macromolécula a estudiar. Se asume que cada partícula pertenece a la misma macromolécula y tiene la misma estructura. Estas partículas están orientadas aleatoriamente respecto de la dirección del haz de electrones por lo que en una micrografía se obtienen distintas proyecciones de la macromolécula. Con el fin de poder realizar una reconstrucción tridimensional de la estructura es necesario asignar el ángulo de proyección a cada partícula como se esquematiza en la figura 2.

Una vez estimada la orientación de un número suficiente de imágenes, la reconstrucción se realiza mediante el Teorema de la Sección Central de Fourier[3] el cual establece que la transformada de Fourier bidimensional de la sección central de un objeto tridimensional es igual a la proyección bidimensional de la transformada de

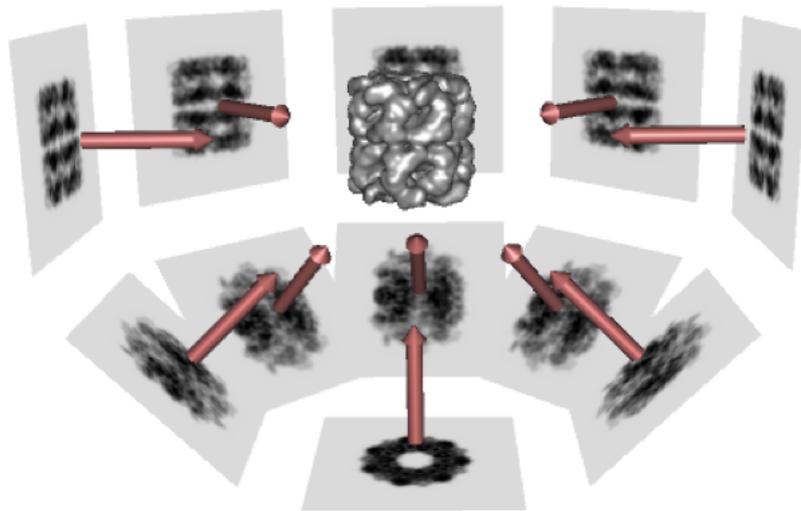


Figura 2: Representación de distintas proyecciones posibles que puede tener una molécula y orientación de las mismas. Imagen extraída de <https://people.csail.mit.edu/gdp/cryoem.html>

Fourier tridimensional del objeto. En la figura 3 se muestra un esquema de la reconstrucción tridimensional a partir de las transformadas de Fourier de las proyecciones de un objeto.

Cabe destacar que no es suficiente con determinar la orientación de las partículas, sino que es necesario conocer también la posición de los ejes de rotación en la imagen ya que puede que no estén centrados en la imagen individual extraída de la micrografía. De esta manera, son 5 los parámetros (3 ángulos de rotación y las 2 coordenadas de los ejes) que hace falta estimar antes de poder realizar la reconstrucción del mapa tridimensional de la molécula. El proceso de estimación de los parámetros es a menudo denominado *alineamiento*.

El esquema general de los métodos utilizados actualmente para realizar el alineamiento de las partículas sigue un proceso de optimización iterativo que parte del conocimiento a priori de la estructura de la molécula [5] [6] [7]. Dada una estructura inicial, se simulan sus proyecciones para muchas orientaciones y se comparan con cada una de las imágenes experimentales. Empleando una regla de ajuste que varía según el algoritmo concreto se asigna una orientación óptima a cada partícula. Todas las imágenes se combinan para obtener un nuevo volumen a partir del cual se vuelve a realizar otra asignación. Esto se repite hasta alcanzar la convergencia del algoritmo. Es un proceso exhaustivo que requiere de mucho tiempo y capacidad computacional. Una vez realizada la asignación global, se utilizan otras técnicas locales para refinar el mapa obtenido [8].

Además de comparar las distintas orientaciones posibles de la molécula, los algoritmos de alineamiento deben realizar una búsqueda espacial en la imagen ya que es común que la molécula se encuentre algo desplazada con respecto al centro de la imagen. En la práctica, las moléculas pueden estar desplazadas hasta 10 píxeles, lo que supone realizar 400 veces más comparaciones, una por cada píxel del cuadrado

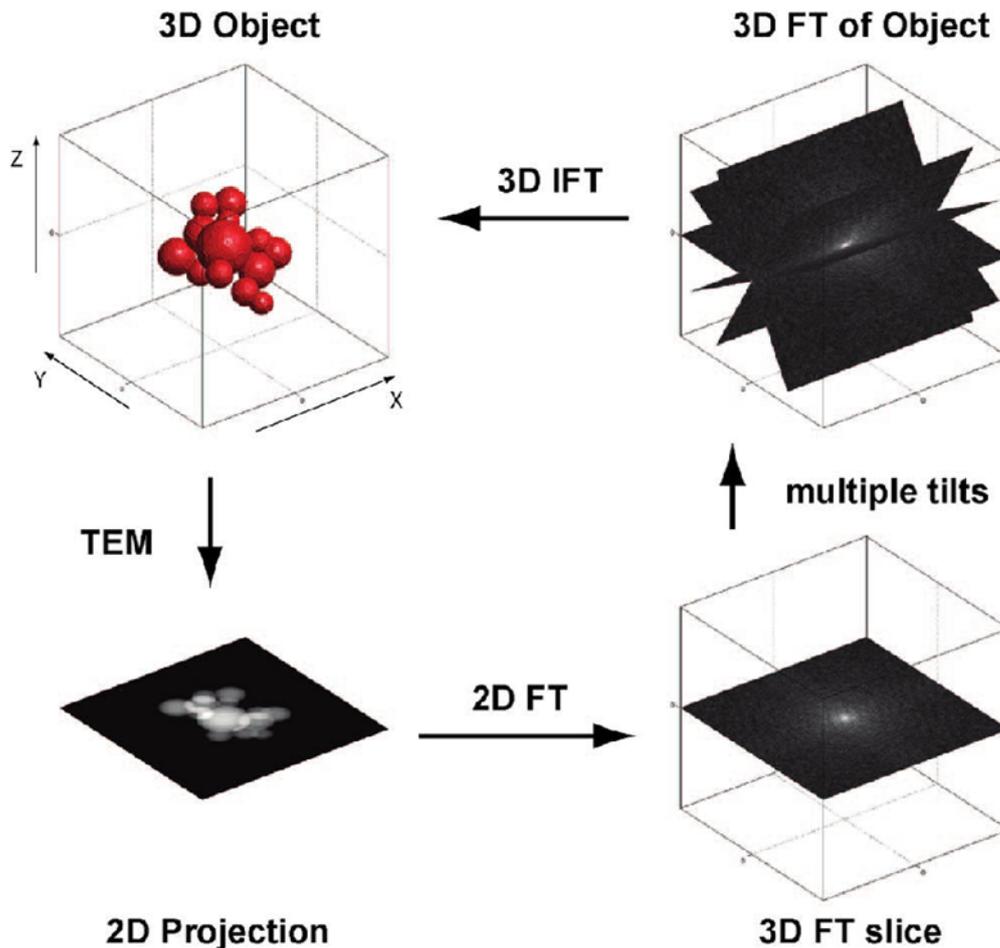


Figura 3: Representación gráfica del esquema de reconstrucción del objeto tridimensional a partir de sus proyecciones aplicando el Teorema de la Sección Central de Fourier [4].

20x20 central de la imagen. Ser capaces de ubicar rápidamente el centro de rotación de la molécula con un margen de error menor, por ejemplo 5 píxeles, permitiría reducir el número de comparaciones necesarias.

La principal dificultad a la que se deben enfrentar los métodos de alineamiento es el bajo ratio señal-ruido, típicamente de 1/100 o 1/1.000. El ruido proviene de distintas fuentes como la interacción de los electrones con el hielo en el que se encuentra la muestra, el ruido de conteo del detector o la influencia sobre los electrones de los distintos artefactos electrónicos cercanos y la radiación de fondo. Además, la Función de Transferencia de Contraste (CTF, por sus siglas en inglés) es otro factor en la criomicroscopía electrónica que afecta a la calidad de la imagen. La CTF describe cómo los electrones transmitidos a través de la muestra interactúan con el objetivo del microscopio y el detector [9]. La interacción con la muestra y los componentes del microscopio puede provocar una atenuación y un cambio de fase en la señal electrónica, lo que resulta en una modulación del contraste de la imagen. La CTF introduce un patrón de oscilación periódica en la imagen en forma de anillos de interferencia. En la figura 4 se muestra cómo la CTF y el ruido gaussiano afectan a

una imagen de microscopía electrónica.

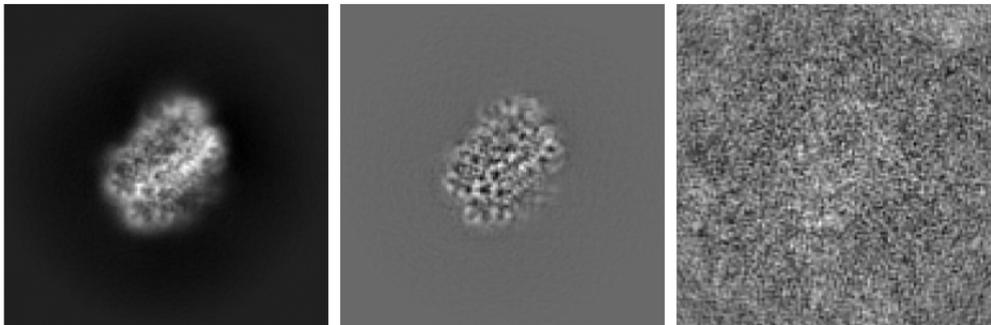


Figura 4: La imagen izquierda es una proyección simulada sin ruido de una molécula. En el centro se aplican los efectos de la CTF y en la imagen de la derecha se ha añadido ruido gaussiano como el que se encuentra en las imágenes experimentales.

1.1. Motivación

El objetivo de este trabajo es estudiar y analizar la posibilidad de utilizar técnicas de aprendizaje profundo para realizar el centrado de moléculas en imágenes experimentales y estimar su orientación. El interés práctico reside en que, según los resultados en cuanto a precisión y tiempo, se podría plantear introducir este tipo de técnicas en los algoritmos actuales de alineamiento con el fin de aligerar alguna de sus fases acelerando el proceso.

El presente texto trata de recoger y exponer las principales herramientas utilizadas y resultados obtenidos durante mi estancia como becario JAE Intro en el Centro Nacional de Biotecnología colaborando en la Unidad de Biocomputación. Se va a comenzar explicando los fundamentos teóricos necesarios para comprender los dos pilares sobre los que ha trabajado: el aprendizaje profundo, herramienta aplicada para tratar de resolver el problema planteado, y la representación matemática de las rotaciones de un objeto tridimensional, necesaria para modelar correctamente el problema. Después, se exponen los datos utilizados y el soporte informático empleado y se presentan y analizan los resultados obtenidos.

2. Aprendizaje profundo

Una red neuronal es un modelo computacional compuesto por nodos interconectados, llamados neuronas artificiales, que trabajan en conjunto para procesar y transmitir información.

Las neuronas de la red están generalmente organizadas en capas. A grandes rasgos se puede distinguir tres tipos de capas: la capa de entrada en la que cada neurona representa un parámetro de entrada, la capa de salida, en la que cada neurona representa un parámetro de salida, y las capas intermedias u ocultas. Las neuronas de cada capa están interconectadas con las neuronas de la capa anterior. De estas reciben entradas ponderadas, aplican una operación matemática y generan una salida. Los datos de entrada se introducen en la primera capa, se transmiten desde cada neurona a todas las neuronas de la siguiente capa y, finalmente, se obtiene un resultado en la capa de salida.

Las conexiones entre neuronas se representan por pesos. Los pesos son los parámetros que se deben ajustar para que la red realice la función deseada, lo cual se logra a partir del entrenamiento de la red. Para ello, se define una función de coste que se va a tratar de optimizar. La red procesa ejemplos, evalúa la función de coste comparando el resultado de la red con el esperado y ajusta los pesos mediante la propagación hacia atrás en cada paso del entrenamiento.

El aprendizaje profundo (deep learning en inglés) es una rama del aprendizaje automático (machine learning) que trata de modelar abstracciones de alto nivel a través de modelos o arquitecturas computacionales que aplican múltiples transformaciones sobre los datos de entrada. Las redes neuronales convolucionales (CNN) son una de las arquitecturas más destacadas del aprendizaje profundo debido a su extendido uso en tareas de procesamiento de imágenes y visión artificial [10].

2.1. Redes Neuronales Convolucionales

Una red convolucional recibe típicamente en su capa de entrada una imagen representada como un tensor de píxeles, una serie de capas convolucionales aplican filtros a los datos de entrada con el fin de extraer las características relevantes, una capa de aplanado convierte el tensor de características resultante en un tensor unidimensional que es procesado por una red densa y una última capa de salida realiza las predicciones finales [11].

A continuación se detallan los tipos de capa más utilizados comúnmente en este tipo de redes neuronales:

- **Capa convolucional.** La capa convolucional es el elemento fundamental de las CNN. Esta capa realiza una convolución entre la matriz de entrada y una serie de matrices más pequeñas que funcionan como filtros. Cada filtro está formado por una matriz de pesos que se multiplican con una región de la imagen y se suman para producir un valor de activación. El filtro se desliza sobre la imagen extrayendo las características locales. Esto produce una nueva

matriz para cada uno de los filtros conocida como mapa de características o de activaciones.

- **Capa de activación.** En una red neuronal, la capa de activación, también conocida como función de activación, se aplica después de la suma ponderada en cada unidad de la red. Su función principal es introducir no linealidad en el modelo, permitiendo que la red aprenda y represente relaciones y patrones complejos en los datos.

Las capas de activación transforman la salida de una unidad o neurona en una forma no lineal. Sin una función de activación, la red neuronal se reduciría a una combinación lineal de las entradas y no sería capaz de aprender y modelar relaciones no lineales en los datos.

Existen varias funciones de activación utilizadas en las redes neuronales pero en el caso de las convolucionales, la función ReLU es la popular debido a su simplicidad y capacidad para manejar gradientes de manera eficiente durante el entrenamiento[12].

- **Capa de normalización por bloques.** La capa de normalización por bloques (Batch Normalization) es una técnica utilizada en las redes neuronales con el fin de normalizar las activaciones de cada capa. Para ello, calcula la media μ y la varianza σ^2 del lote actual y aplica la siguiente transformación [15]:

$$y_i = \gamma \frac{x_i - \mu}{\sqrt{\sigma^2 + \epsilon}} + \beta,$$

donde ϵ es una constante pequeña y γ y β son parámetros que se entrenan y sirven para re-escalar y descentrar los datos de nuevo.

Esta capa ayuda a solucionar el problema del cambio de distribución de los datos en diferentes lotes de muestras durante el entrenamiento lo que contribuye a una mayor estabilidad y velocidad de convergencia. Además introduce cierto ruido por lo que también puede tener un efecto regularizador, reduciendo el riesgo de sobreajuste en la red.

- **Reducción de dimensionalidad.** La capa de reducción de la dimensionalidad (Pooling Layer) reduce el tamaño de los mapas de características generados por las capas convolucionales. Para ello reemplaza el valor de salida de ciertas localizaciones del mapa por un resumen estadístico de las salidas cercanas.

El mapa de características es dividido en regiones, típicamente de tamaño 2x2, y en cada una de ellas se aplica una función. Existen muchas funciones que se pueden utilizar como la media o distintas normas pero lo más popular es tomar el valor máximo de cada región (max pooling).

Esta capa es útil para extraer características invariantes a pequeñas variaciones en la posición de los objetos en la imagen. Además, ayuda a disminuir el número de parámetros de la red, lo que puede ayudar a controlar el sobreajuste y acelerar el entrenamiento[13].

- **Capa de regularización.** La capa de regularización (Dropout) se aplica durante el entrenamiento y consiste en desactivar aleatoriamente un porcentaje de los nodos de la capa anterior en cada paso del entrenamiento.

Esto permite reducir el sobreajuste ya que elimina las adaptaciones complejas que puedan surgir entre neuronas obligando a las capas a adoptar un enfoque probabilístico[14].

- **Capa de aplanado.** La capa de aplanado (Flatten) simplemente toma la entrada multidimensional y la transforma en un vector unidimensional. Se utiliza comúnmente para conectar las capas convolucionales con las capas densas al final de la red.
- **Capa densa.** Las capas densas o completamente conectadas son las capas básicas de una red neuronal que conecta todas las neuronas de la capa anterior con cada neurona de la siguiente.

2.2. Redes Residuales

Los diferentes avances en el campo del aprendizaje profundo llevados a cabo hasta 2015 indicaban que cuanto mayor era la profundidad de una red, mayor la complejidad de las funciones que esta podía calcular. Gracias a los nuevos avances técnicos, el coste computacional que suponía añadir cada vez más capas dejaba de ser un inconveniente. Sin embargo, surgieron esencialmente dos problemas con este tipo de redes profundas. En primer lugar, el desvanecimiento del gradiente: los gradientes se vuelven muy pequeños a medida que se retropropagan hacia las capas iniciales, lo que dificulta el entrenamiento y afecta al rendimiento. Este fenómeno se puede aplacar con una buena inicialización de los pesos. El segundo problema es la degradación. Aunque teóricamente una red con más capas debería aproximar al menos igual de bien la función, pues basta con que las capas adicionales actúen como la función identidad, experimentalmente se observaba que en muchos casos, a partir de un cierto número de capas, una red más profunda no sólo no mejoraba el rendimiento si no que podía incluso empeorarlo. Este fenómeno se debe a que el sistema se vuelve más difícil de optimizar.

Con el fin de evitar estos problemas y permitir con ello continuar construyendo redes neuronales más profundas, surgen en 2015 las redes neuronales residuales[16]. Mientras que en una red neuronal convencional la información fluye a través de las capas de manera secuencial, de una capa a la siguiente, en una red residual se introducen conexiones atajo que permiten conectar neuronas de capas no consecutivas, saltando una o más capas intermedias.

Una red residual está construida a partir de bloques residuales. La información fluye a través de estos bloques por medio de una ruta residual y de una conexión atajo que realiza la operación identidad. Sea x la entrada en un bloque, entonces la salida del bloque se puede expresar matemáticamente como:

$$y = F(x) + x,$$

donde $F(x)$ es la transformación residual que realizan las capas dentro del bloque. En la figura 5 se muestra la arquitectura de un bloque residual básico en una red convolucional con dos capas convolucionales. Se suele denominar *bloque identidad* a este tipo de bloques ya que conservan la dimensión de entrada y la ruta atajo es precisamente la identidad. Si es necesario reducir la dimensión de entrada, el atajo realiza alguna operación de reducción de la dimensionalidad, generalmente mediante una capa convolucional con filtros de tamaño 1 y un espaciado mayor que la unidad. Este tipo de bloques son a veces denominados *bloques convolucionales* por la operación de convolución que se realiza en el atajo.

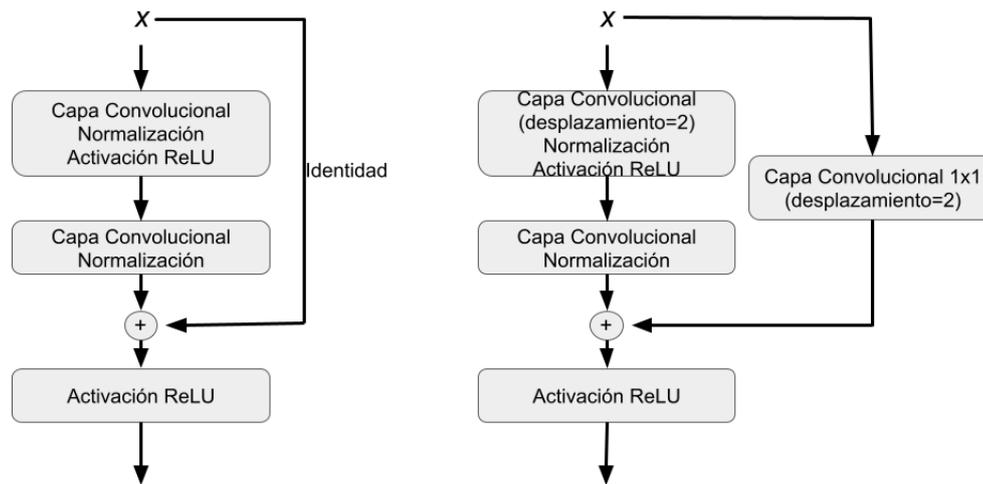


Figura 5: El esquema de la parte izquierda de la figura representa un bloque identidad en el que se existe un atajo pero sin reducción de la dimensionalidad. El esquema de la derecha representa un bloque convolucional en el que la dimensionalidad se reduce.

Un ejemplo clásico de red con este tipo de arquitectura es la ResNet34. Una variante de los bloques presentados con los conocidos como *Cuellos de botella* (*Bottleneck* en inglés) que reducen la dimensionalidad de la entrada antes de aplicar las capas adicionales para volver a expandir la dimensionalidad a la salida. Estos bloques son más eficientes en términos de cálculos y parámetros permitiendo construir redes residuales más profundas como las ResNet50, ResNet101 y ResNet152 [17]. Se ha trabajado mucho en este tipo de redes en los últimos años surgiendo interesantes variantes como la *HighwayNet* o la *FractalNet* que no se detallan ya que escapan de los objetivos de este trabajo al no haberse empleado estas arquitecturas.

3. Rotaciones

Aunque reconocer la orientación de un objeto en una imagen pueda parecer un problema sencillo ya que instintivamente lo realizamos en todo momento de nuestra vida cotidiana, a la hora de diseñar un algoritmo capaz de realizar esta tarea aparecen distintos problemas y consideraciones a tener en cuenta.

En primer lugar es necesario definir matemáticamente la orientación de un objeto. Fijando una orientación de referencia, cualquier orientación se puede describir como la rotación necesaria para alcanzar dicha orientación desde la de referencia.

Una rotación de un objeto tridimensional se representa como una aplicación lineal de \mathbb{R}^3 en si mismo. La matriz 3x3 asociada a la aplicación lineal se denomina matriz de rotación. Las rotaciones conservan ángulos y longitudes, de manera que, con respecto a la base canónica de \mathbb{R}^3 , las matrices de rotación son matrices ortogonales dado que sus columnas forman una base ortonormal. Además, las rotaciones preservan la orientación de manera que su determinante debe ser 1. El subgrupo de matrices ortogonales con determinante 1 se llama grupo ortogonal especial, denotado por $SO(3)$. La composición de dos rotaciones es una rotación cuya matriz asociada es el producto de las matrices de rotación. La operación composición se corresponde por lo tanto con el producto de matrices.

El Teorema de la rotación de Euler [18] dice que cualquier rotación en el espacio de un objeto tridimensional que mantenga un punto fijo también debe dejar fijo un eje completo. Conocido este eje, la rotación viene determinada por el ángulo de rotación θ . Tomando como eje Z el eje fijo, la matriz de rotación es:

$$\mathbf{R} = \begin{pmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

En la práctica, trabajar con matrices de rotación supone estimar y optimizar nueve parámetros con restricciones no lineales, lo que puede llegar a ser inviable. Es por ello que se buscan distintas maneras de representar las rotaciones tridimensionales con el objetivo de simplificar el problema y los cálculos. Dependiendo del problema concreto, una representación puede resultar mejor que otra. En el campo de la microscopía electrónica las dos representaciones más utilizadas son los ángulos de Euler y los cuaterniones.

3.1. Ángulos de Euler

Fijado un sistema de ejes X,Y y Z, la orientación de un objeto 3D puede ser descrita por tres ángulos denominados ángulos de Euler y denotados por ϕ , θ y ψ . ϕ representa la rotación alrededor del eje Z, θ es el segundo ángulo y representa la rotación entorno al nuevo eje Y. Estos dos ángulos describen la dirección de proyección respecto al objeto. Así, $\phi \in [0, 2\pi)$ y $\theta \in [0, \pi)$. ψ representa la rotación del objeto en el plano de proyección definido por ϕ y θ , alrededor del nuevo eje Z, de manera que $\psi \in [0, 2\pi)$. Si bien existen otras convenciones de ejes que resultan equivalentes en

la práctica, la presentada es la utilizada de manera general en microscopía.

Según el Teorema de la rotación de Euler, es posible expresar cualquier rotación en términos de estos tres ángulos[19]. Si \mathbf{R} es la matriz de dicha rotación, entonces:

$$\mathbf{R} = R_{z'}(\psi) \cdot R_{y'}(\theta) \cdot R_z(\phi),$$

donde:

$$R_z(\phi) = \begin{pmatrix} \cos(\phi) & -\sin(\phi) & 0 \\ \sin(\phi) & \cos(\phi) & 0 \\ 0 & 0 & 1 \end{pmatrix},$$

$$R_y(\theta) = \begin{pmatrix} \cos(\theta) & 0 & -\sin(\theta) \\ 0 & 1 & 0 \\ \sin(\theta) & 0 & \cos(\theta) \end{pmatrix},$$

$$R_z(\psi) = \begin{pmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

Esta representación resulta útil debido a que es intuitiva y sólo requiere 3 parámetros. Sin embargo, para los casos $\theta = 0$ y $\theta = \pi$, una rotación viene determinada por $\psi + \phi$ y $\psi - \phi$ respectivamente, no por los ángulos individuales, surgiendo ambigüedades. Esto es conocido como *bloqueo del cardán*[20].

3.2. Cuaterniones

Además de los ángulos de Euler, los cuaterniones son otra representación de las rotaciones muy utilizada por los algoritmos de reconstrucción en criomicroscopía electrónica. Su utilidad en el contexto del presente trabajo reside en que permiten calcular de forma sencilla y eficiente la rotación media entre un conjunto de rotaciones.

Los cuaterniones pueden ser interpretados como una extensión de los números complejos introduciendo dos nuevas unidades imaginarias, \mathbf{j} y \mathbf{k} [21]:

$$\mathbf{q} = q_0 + q_1\mathbf{i} + q_2\mathbf{j} + q_3\mathbf{k}.$$

De manera equivalente, un cuaternión puede ser expresado como un vector de cuatro componentes:

$$\mathbf{q} = \begin{pmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \\ q_4 \end{pmatrix} = \begin{pmatrix} q_0 \\ \vec{n}_q \end{pmatrix}.$$

El conjugado de un cuaternión se define de manera análoga al conjugado de un número complejo:

$$\mathbf{q}^* = q'_0 \mathbf{i} - q_2 \mathbf{j} - q_3 \mathbf{k} = \begin{pmatrix} q_0 \\ -\vec{n}_q \end{pmatrix}.$$

La operación producto \otimes entre dos cuaterniones se realiza componente a componente. La siguiente tabla resume las reglas de multiplicación entre las distintas componentes de un cuaternión:

\otimes	1	\mathbf{i}	\mathbf{j}	\mathbf{k}
1	1	\mathbf{i}	\mathbf{j}	\mathbf{k}
\mathbf{i}	\mathbf{i}	-1	\mathbf{k}	$-\mathbf{j}$
\mathbf{j}	\mathbf{j}	$-\mathbf{k}$	-1	\mathbf{i}
\mathbf{k}	\mathbf{k}	\mathbf{j}	$-\mathbf{i}$	-1

A partir de estas reglas se comprueba fácilmente que el producto en forma vectorial se puede escribir de la forma:

$$\mathbf{q} \otimes \mathbf{q}' = \begin{pmatrix} q_0 \\ -\vec{n}_q \end{pmatrix} \otimes \begin{pmatrix} q'_0 \\ -\vec{n}'_q \end{pmatrix} = \begin{pmatrix} q_0 q'_0 - \vec{n}_q \cdot \vec{n}'_q \\ q_0 \vec{n}'_q + q'_0 \vec{n}_q + \vec{n}_q \times \vec{n}'_q \end{pmatrix}.$$

La norma de un cuaternión viene dada por la raíz cuadrada del producto por su conjugado:

$$\|\mathbf{q}\| = \sqrt{\mathbf{q} \otimes \mathbf{q}^*} = \sqrt{q_0^2 + q_1^2 + q_2^2 + q_3^2}.$$

Una rotación de ángulo α entorno a un eje tridimensional \vec{u} se puede expresar como un cuaternión de norma 1 de la siguiente manera:

$$\mathbf{q}_{\vec{u}, \alpha} = \begin{pmatrix} \cos(\alpha/2) \\ \sin(\alpha/2) \vec{u} \end{pmatrix},$$

entonces, la matriz de rotación en términos de los parámetros q_0, q_1, q_2, q_3 del cuaternión es:

$$\mathbf{R} = \begin{pmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_0 q_1 - q_2 q_3) & 2(q_0 q_2 + q_1 q_3) \\ 2(q_0 q_1 + q_2 q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_1 q_2 - q_0 q_3) \\ 2(q_0 q_2 - q_1 q_3) & 2(q_1 q_2 + q_0 q_3) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{pmatrix}.$$

Para pasar de cuaterniones a matriz de rotación, se resuelve el sistema lineal resultante de tomar los elementos de la diagonal de la matriz R anterior y añadir la condición de que la norma del cuaternión sea 1:

$$\begin{pmatrix} 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{pmatrix} = \begin{pmatrix} r_{11} \\ r_{22} \\ r_{33} \\ 1 \end{pmatrix},$$

donde $\text{signo}(q_0) = 1$, $\text{signo}(q_1) = \text{signo}(r_{32} - r_{23})$, $\text{signo}(q_2) = \text{signo}(r_{13} - r_{31})$ y $\text{signo}(q_3) = \text{signo}(r_{21} - r_{12})$.

Con esta representación y la definición de producto entre cuaterniones, se demuestra de manera directa que, dadas dos rotaciones \mathbf{R} y \mathbf{R}' representadas por \mathbf{q} y \mathbf{q}' , entonces $\mathbf{q}' \otimes \mathbf{q}$ representa la rotación resultante de aplicar \mathbf{R} seguida de \mathbf{R}' .

Para calcular una rotación media entre un conjunto de rotaciones es necesario en primer lugar establecer una noción de distancia entre rotaciones en $SO(3)$. El ángulo entre dos vectores \vec{u} y \vec{u}' unitarios en \mathbb{R}^3 viene dado por

$$\alpha(\vec{u}, \vec{u}') = \arccos(\vec{u} \cdot \vec{u}').$$

Se define la distancia entre dos rotaciones \mathbf{R} y \mathbf{R}' como el mayor ángulo entre $\mathbf{R}(\vec{v})$ y $\mathbf{R}'(\vec{v})$ para un vector \vec{v} arbitrario, matemáticamente:

$$d_{SO(3)}(\mathbf{R}, \mathbf{R}') = \max_{\vec{v} \in S^2} \alpha(\mathbf{R}(\vec{v}), \mathbf{R}'(\vec{v})).$$

Dado un conjunto de rotaciones, $\{\mathbf{R}_i\}_{1 \leq i \leq N}$, la rotación promedio será aquella $\mathbf{R} \in SO(3)$ que minimice la expresión

$$\sum_{i=1}^N d_{SO(3)}^2(\mathbf{R}, \mathbf{R}_i).$$

Representando las rotaciones $\{\mathbf{R}_i\}_{1 \leq i \leq N}$ como cuaterniones, $\{\mathbf{q}_i\}_{1 \leq i \leq N}$, el autovector asociado al mayor autovalor de la matriz

$$\mathbf{T} = \sum_{i=1}^N \mathbf{q}_i \mathbf{q}_i^t,$$

es una aproximación de la rotación promedio[22].

3.3. Continuidad

Las redes neuronales por lo general tienen dificultades a la hora de aproximar funciones con discontinuidades. Resulta entonces necesario estudiar la continuidad de las distintas representaciones angulares y definir, si es necesario, una nueva representación que evite los problemas de aprendizaje derivados de las discontinuidades presentes en otras representaciones.

Es sencillo encontrar por qué los ángulos de Euler no son una representación continua. Por ejemplo, fijando $\theta = \psi = 0$, el límite de ϕ en la matriz identidad no está definido ya que en una dirección es 0 y en la otra es 2π .

Se podría tantear la idea de tomar el seno y el coseno de cada ángulo en lugar de los propios ángulos, realizando la regresión de 6 parámetros (o 5 ya que para θ se puede prescindir del coseno). Esto solucionaría en primera instancia la discontinuidad entre 0 y 2π . Sin embargo, se mantiene el segundo de los inconvenientes de esta representación, y es que para $\theta = 0$ y $\theta = \pi$ las soluciones están degeneradas ya que

distintos valores de ψ y ϕ pueden llevar a una misma rotación. Esto provoca grandes errores para los valores límite de θ como se muestra experimentalmente en la parte de resultados de la sección Estimación Angular.

Se puede probar [23] que no existe ninguna representación continua del espacio de las rotaciones de un objeto tridimensional, $SO(3)$, en un subconjunto de \mathbb{R}^d con $d < 5$. Por lo tanto, para que una representación pueda ser continua es necesario que esté en un espacio de 5 o más dimensiones. Es decir, que ni los ángulos de Euler ni los cuaterniones pueden ser representaciones continuas ya que tienen 3 y 4 parámetros respectivamente. Teniendo esto en cuenta, se va a utilizar una representación en 6D que consiste en emplear las dos primeras columnas de la matriz de rotación.

Para poder pasar de esta nueva representación a ángulos de Euler o cuaterniones se obtiene como paso intermedio la matriz de rotación. El paso de $SO(3)$ a la representación 6D es trivial a partir de la definición:

$$g \left(\begin{bmatrix} | & | & | \\ a_1 & a_2 & a_3 \\ | & | & | \end{bmatrix} \right) = \begin{bmatrix} | & | \\ a_1 & a_2 \\ | & | \end{bmatrix}.$$

Para pasar de la representación 6D a $SO(3)$ es necesario asegurarse de que la matriz sea ortonormal:

$$f \left(\begin{bmatrix} | & | \\ a_1 & a_2 \\ | & | \end{bmatrix} \right) = \begin{bmatrix} | & | & | \\ b_1 & b_2 & b_3 \\ | & | & | \end{bmatrix},$$

siendo b_1 el vector a_1 normalizado, b_2 el vector $a_2 - (b_1 \cdot a_2)b_1$ normalizado y $a_3 = b_1 \times b_2$.

3.4. Simetrías

Existen muchos ejemplos de macromoléculas que presentan algún tipo de simetría rotacional. Existen proteínas como la hemoglobina de la figura 6 que pueden presentar simetría rotacional entorno a un único eje. En este caso, se dice que la macromolécula (o el objeto en general) tiene simetría C_n con n el orden de la simetría que presenta. Por ejemplo, dada una macromolécula con simetría C_4 , si es rotada un ángulo $2\pi/4$ entorno a su eje de simetría, observaremos la macromolécula exactamente en la misma posición que tenía inicialmente.

En el caso de muchos tipos de virus como el mostrado en la parte derecha de la figura 6 y otros tipos de proteína como la apoferritina, hablamos de simetría más alta porque tienen más de un único eje de simetría rotacional.

Vamos a formalizar matemáticamente el concepto de simetría. Se dice que un volumen $V(\mathbf{r})$ tiene simetría si $V(\mathbf{r}\mathbf{S}) = V(\mathbf{r})$ para alguna matriz \mathbf{S} diferente de la identidad cuyos autovalores tengan módulo la unidad. \mathbf{S} puede ser una matriz de rotación en $SO(3)$ pero también puede representar una inversión, una reflexión o una combinación de reflexión y rotación. El determinante de las matrices de rotación es

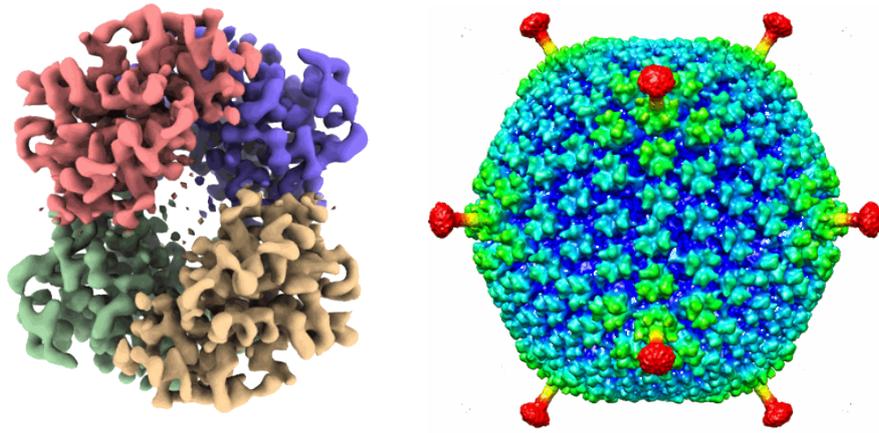


Figura 6: La hemoglobina (izquierda) y el adenovirus (derecha) son dos ejemplos de macromoléculas con simetría. Imágenes obtenidas de <https://www.ebi.ac.uk/emdb/>.

1 ya que preservan la orientación, y el del resto de operaciones de simetría es -1 . En el contexto de este trabajo estudiaremos únicamente las simetrías del primer tipo, es decir, rotaciones propias del grupo $SO(3)$.

El conjunto de matrices de simetría rotacional de un cuerpo tridimensional forma un subgrupo de $SO(3)$ [24]. El subgrupo asociado a las operaciones de simetría de un cilindro, por ejemplo, tiene infinitos elementos ya que podemos girar el objeto por su eje de simetría cualquier ángulo. Las moléculas están compuestas por átomos por lo que no pueden presentar una forma tan regular como la de un cilindro. El número de posibles operaciones de simetría deberá ser finito.

Podemos clasificar los subgrupos finitos de $SO(3)$ en cuatro clases o tipos de simetría:

- Simetría cíclica C_n : Subgrupo cíclico de orden n . Se corresponde con la rotación de ángulo $2\pi/n$ entorno a un eje.
- Simetría tetraédrica T : Subgrupo de $SO(3)$ de 12 elementos que se corresponden con las simetrías propias de un tetraedro regular. Es isomorfo al subgrupo alternante A_4 del grupo S_4 de las permutaciones de cuatro elementos. Una manera visual de entender el isomorfismo es identificar los vértices de un tetraedro con los cuatro elementos que permutan.
- Simetría octaédrica O : Es la propia de un octaedro regular o un cubo entre otros poliedros. El subgrupo cuenta con 24 simetrías rotacionales y es isomorfo al grupo simétrico S_4 . Tiene por lo tanto dos generadores, existe un eje de orden 4 y un eje de orden 3 a partir de los cuales se pueden describir todas las rotaciones.
- Simetría icosaédrica I : Cuenta con las 60 simetrías rotacionales propias del icosaedro regular. Este subgrupo es isomorfo al grupo alternante de 5 elementos A_5 .

Si una macromolécula presenta cierta simetría, dos imágenes pueden ser idénticas pero corresponderse con orientaciones distintas. En estas condiciones, no existe una función $\mathcal{F} : I \rightarrow R$ tal que a cada imagen haga corresponder una única orientación. Por lo tanto, un modelo entrenado con las imágenes de una molécula con simetría puede tener problemas de aprendizaje al no existir realmente la función que se quiere aproximar.

Con el fin de resolver estas ambigüedades se ha probado a implementar el método que se describe en el artículo [25]. La idea que se propone para resolver este problema es identificar entre sí todas las orientaciones equivalentes conocida la simetría de la molécula, haciendo que todas las imágenes equivalentes etiquetadas en un primer lugar con distintos ángulos pasen a estar etiquetadas con los de una única orientación. Para ello, dado un subgrupo de matrices de simetría M , se define la siguiente función dada cualquier matriz de rotación $\mathbf{R} \in SO(3)$:

$$\text{Map}(\mathbf{R}) = \mathbf{R}\hat{\mathbf{S}}^{-1},$$

donde

$$\hat{\mathbf{S}} = \arg \min_{\mathbf{S}} \|\mathbf{R}\mathbf{S}^{-1} - I\|_F,$$

para $\mathbf{S} \in M$. $\|\cdot\|_F$ representa la norma de Froebenius. Esta función lleva cualquier matriz de rotación a la matriz de rotación equivalente según el subgrupo M más cercana en el sentido de la norma de Froebenius a la matriz identidad. De esta forma, dadas dos matrices de rotación equivalentes $\mathbf{R}_1 \sim_M \mathbf{R}_2$, $\text{Map}(\mathbf{R}_1) = \text{Map}(\mathbf{R}_2)$ resolviendo las ambigüedades.

Mediante la función Map conseguimos que la función $\mathcal{F} : I \rightarrow R$ que queremos aproximar exista. No obstante, si M no es trivial, \mathcal{F} es discontinua. En la figura 7 se representa cómo actúa la función Map sobre las rotaciones de un objeto con simetría C_2 tomando una proyección paralela a su eje de simetría. Cada rotación tiene una equivalente situada en el extremo opuesto de la esfera. La imagen de ambas por la aplicación Map es la misma: la rotación más cercana a la identidad de entre las dos. De esta forma, las rotaciones de ángulos entre $3\pi/2$ y $\pi/2$ no cambian al aplicar Map mientras que las rotaciones entre $\pi/2$ y $3\pi/2$ tienen como imagen su equivalente. La discontinuidad se produce entonces en los puntos $\pi/2$ y $3\pi/2$ ya que una rotación de ángulo $\pi/2 - \epsilon$ tiene como imagen ella misma mientras que la imagen de la rotación $\pi/2 + \epsilon$ es la rotación de ángulo $3\pi/2 + \epsilon$.

La manera que el artículo propone para trabajar con estas discontinuidades es dividir el espacio de rotaciones en dominios de manera que aquellos dominios en los que la función \mathcal{F} sea continua se mantengan igual porque en ellos no habrá problemas de entrenamiento para la red mientras que aquellos dominios con puntos en los que existan discontinuidades se aplique otra aplicación similar a Map capaz de resolver dichas discontinuidades.

Sea \mathbf{R} una matriz de rotación que no sea la identidad, entonces existen dos matrices de rotación \mathbf{A} y \mathbf{B} distintas tales que $\mathbf{A}^2 = \mathbf{B}^2 = \mathbf{R}$. Para ver que existen, sea \vec{v} el autovector asociado al autovalor 1 de \mathbf{R} y α el ángulo de rotación entorno a este eje, entonces tomamos como \mathbf{A} la matriz que describe una rotación entorno a \vec{v} de ángulo $\alpha/2$ y como \mathbf{B} , la matriz que describe una rotación entorno a \vec{v} de ángulo

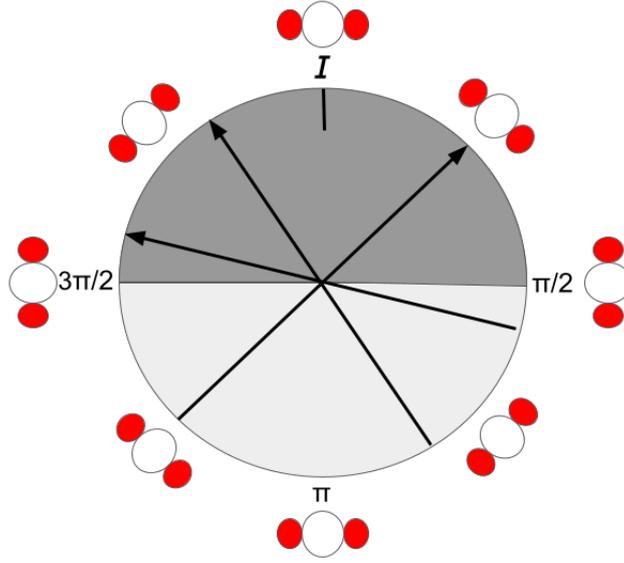


Figura 7: Esquema del funcionamiento de la función Map para simetría $C2$.

$\pi + \alpha/2$. Por construcción, se tiene que $\mathbf{A}^2 = \mathbf{B}^2 = \mathbf{R}$. Para ver que son las únicas matrices de rotación, supongamos que existe otra matriz de rotación \mathbf{C} que también cumple $\mathbf{C}^2 = \mathbf{R}$. Sea \vec{w} el autovector asociado al autovalor 1 de \mathbf{C} , entonces $\mathbf{C}\vec{w} = \vec{w}$ y, por lo tanto, $\mathbf{R}\vec{w} = \mathbf{C}\mathbf{C}\vec{w} = \mathbf{C}\vec{w} = \vec{w}$. Esto implica que \vec{w} también es autovector asociado al valor 1 de \mathbf{R} y como \mathbf{R} no es trivial, $\vec{w} = \vec{v}$. Por lo tanto, \mathbf{C} es una rotación entorno al mismo eje que \mathbf{R} . Sea β su ángulo de rotación, entonces se tiene que dar $\beta + \beta = \alpha$ o $\beta + \beta = \alpha + 2\pi$, por lo que $\beta = \alpha/2$ o $\beta = \pi + \alpha/2$ y $\mathbf{C} = \mathbf{A}$ o $\mathbf{C} = \mathbf{B}$.

Sea M el subgrupo finito de las matrices de simetría. Definimos el conjunto \sqrt{M} como:

$$\sqrt{M} = \{\mathbf{R} \in SO(3) : \mathbf{R}^2 \in M\}.$$

Por ejemplo, si la simetría es $C2$ y $M = \{I, \mathbf{R}_\pi\}$, este conjunto es:

$$\sqrt{M} = \{I, \mathbf{R}_{\pi/2}, \mathbf{R}_\pi, \mathbf{R}_{3\pi/2}\}.$$

Los dominios que buscamos se generan a partir de $\sqrt{M} = \mathbf{R}_k$ y Map:

$$\Omega^{(k)} = \{\mathbf{R} : \hat{\mathbf{S}}(\mathbf{R}) = \mathbf{R}_k\},$$

donde $\hat{\mathbf{S}}(\mathbf{R})$ es el de la función Map tomando \sqrt{M} en lugar de M .

En el caso de existir un único eje de simetría de orden M , sea $\mathbf{R}_{\pi/M}$ la matriz generadora del grupo de menor rotación, los dominios $\Omega^{(k)}$ con k par son equivalentes entre sí y los dominios $\Omega^{(k)}$ con k impar son equivalentes entre sí de manera que podemos trabajar con únicamente dos dominios, $\Omega^{(0)}$ y $\Omega^{(1)}$. Entonces se define la función Map':

$$\text{Map}'(\mathbf{R}) = \mathbf{R}\hat{\mathbf{S}}^{-1},$$

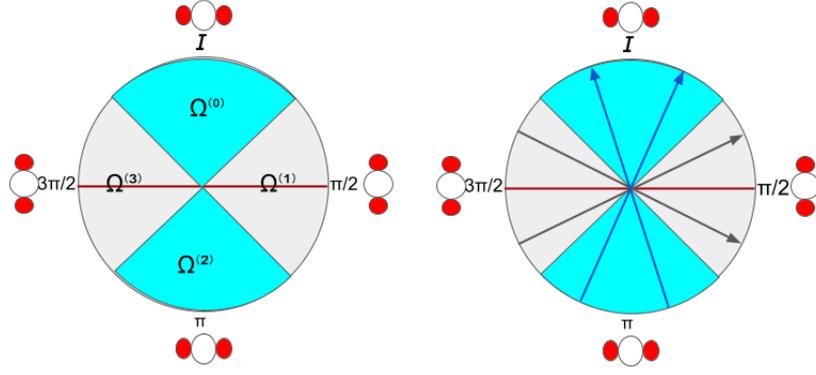


Figura 8: Esquema del funcionamiento de la función Map' para simetría $C2$

donde

$$\hat{\mathbf{S}} = \begin{cases} \arg \underset{\mathbf{S}}{\text{mín}} \|\mathbf{R}\mathbf{S}^{-1} - I\|_F & \text{si } \text{Map}(\mathbf{R}) \in \Omega^{(0)}, \\ \arg \underset{\mathbf{S}}{\text{mín}} \|\mathbf{R}\mathbf{S}^{-1} - \mathbf{R}_{\pi/M}\|_F & \text{en otro caso.} \end{cases}$$

Volvemos al caso de simetría $C2$ para poder visualizar la nueva aplicación Map'. En la imagen izquierda de la figura 8 se representan los dominios definidos. Dominios equivalentes bajo la simetría tienen el mismo color. En la parte derecha de la figura se esquematiza cuál sería la imagen de cada rotación por la aplicación Map' teniendo en cuenta la división por dominios. Como se puede comprobar, ya no aparecen discontinuidades dentro de los dominios.

La definición de Map' se puede generalizar para simetrías con más de un eje de rotación. Vamos a explicar cómo se generaliza para el caso de 2 ejes. El caso de simetrías con tres ejes de rotación es análogo. Sean u y v los ejes de rotación de orden M y N respectivamente. Siguiendo la definición de los dominios $\Omega^{(k)}$ dada anteriormente, denotamos $\Omega^{(0)}$ al dominio asociado a la I , $\Omega^{(1)}$ al asociado a $\mathbf{R}_{\pi/M}^u$, $\Omega^{(-1)}$ al asociado a $\mathbf{R}_{-\pi/M}^u$, $\Omega^{(2)}$ al asociado a $\mathbf{R}_{\pi/N}^v$ y $\Omega^{(-2)}$ al asociado a $\mathbf{R}_{-\pi/N}^v$. Entonces la aplicación Map' es:

$$\text{Map}'(\mathbf{R}) = \mathbf{R}\hat{\mathbf{S}}^{-1},$$

donde

$$\hat{\mathbf{S}} = \begin{cases} \arg \underset{\mathbf{S}}{\text{mín}} \|\mathbf{R}\mathbf{S}^{-1} - I\|_F & \text{si } \text{Map}(\mathbf{R}) \in \Omega^{(0)}, \\ \arg \underset{\mathbf{S}}{\text{mín}} \|\mathbf{R}\mathbf{S}^{-1} - \mathbf{R}_{\pi/M}^u\|_F & \text{si } \text{Map}(\mathbf{R}) \in \Omega^{(1)} \vee \text{Map}(\mathbf{R}) \in \Omega^{(-1)}, \\ \arg \underset{\mathbf{S}}{\text{mín}} \|\mathbf{R}\mathbf{S}^{-1} - \mathbf{R}_{\pi/N}^v\|_F & \text{si } \text{Map}(\mathbf{R}) \in \Omega^{(2)} \vee \text{Map}(\mathbf{R}) \in \Omega^{(-2)}, \\ \arg \underset{\mathbf{S}}{\text{mín}} \|\mathbf{R}\mathbf{S}^{-1} - \mathbf{R}_{\pi/M}^u \mathbf{R}_{\pi/N}^v\|_F & \text{en otro caso.} \end{cases}$$

Implementar este método supone tener que entrenar un clasificador y tantos regresores como dominios. Esto puede conllevar dos problemas: el aumento del tiempo de entrenamiento, al tener que trabajar con varias redes distintas; y además, al tener

que hacer dos predicciones, primero clasificar la imagen y después realizar la estimación angular, la probabilidad de error aumenta. Por esta razón, se propone un método alternativo para trabajar con partículas con simetría C_n .

Durante el alineamiento y la reconstrucción de moléculas con este tipo de simetría se establecen los ejes de manera que el eje Z coincida con el eje de simetría. Teniendo esto en cuenta, supongamos que n es el orden del grupo de simetría propio de la molécula. Entonces, todas las orientaciones se pueden describir a partir de la terna de ángulos de Euler (ϕ, θ, ψ) con ϕ en el intervalo $[0, 2\pi/n)$. El problema es que existe una discontinuidad en $\phi = 0$ y $\phi = 2\pi/n$ ya que $(0, \theta, \psi) \sim (2\pi/n, \theta, \psi)$. Pero si se multiplica por n el primer ángulo, entonces se tiene $(0, \theta, \psi) \sim (n2\pi/n, \theta, \psi) = (2\pi, \theta, \psi) = (0, \theta, \psi)$, solucionando la discontinuidad. El método propuesto consiste en identificar la orientación de cada imagen con su equivalente de manera que $\phi \in [0, 2\pi/n)$, lo cuál es trivial sin más que realizar el "módulo" $2\pi/n$ de su ángulo de Euler ϕ , y, después, multiplicar el resultado por n . Lo que se está consiguiendo con esto es identificar las fronteras equivalentes del dominio fundamental de la simetría para formar la esfera a partir de la misma lo que permite trabajar como si la molécula no tuviese simetría. Esto no es generalizable a moléculas con simetría más alta.

4. Datos y Software

4.1. Datos

Los conjuntos de datos empleados han sido extraídos de la base de datos *Empiar*, <https://www.ebi.ac.uk/empiar/>. En esta página web se pueden encontrar diversos conjuntos de datos con imágenes de macromoléculas obtenidas por microscopía electrónica. Dependiendo del conjunto de datos, puede contener desde las micrografías que se obtienen directamente del microscopio electrónico hasta imágenes de moléculas individuales ya procesadas. Nos interesan aquellas imágenes individuales que ya han sido alineadas, es decir, a las que ya se ha asignado la posición del centro y la orientación y por lo tanto están etiquetadas para el entrenamiento de los modelos.

Aunque existen otros formatos, la mayoría de imágenes se encuentran en archivos con extensión *.mrc* o *.mrcs*[26]. En los conjuntos de datos utilizados, la información referente a cada imagen como la orientación y posición de la partícula, información de la CTF o las coordenadas de la imagen en la micrografía original se encuentran en archivos de texto con extensión *.star*[27].

Las imágenes muestran la densidad electrónica en una escala monocromática que suele representarse en escala de grises donde píxeles más claros equivalen a densidades más altas.

Las imágenes son cuadradas, con un tamaño en píxeles que varía de un conjunto a otro. Con el fin de que todas tengan el mismo número de píxeles se ha reducido el tamaño de las imágenes a 128x128 por ser una cota inferior del tamaño de estas. Una vez reducido el tamaño, se aplica un filtro de Wiener[28] para corregir los efectos de la CTF. En la figura 9 se muestra una imagen experimental de una molécula como las que se pueden encontrar en la base de datos de Empiar y la corrección de la misma imagen con un filtro de Wiener. Es difícil apreciar la molécula en la parte central de la imagen debido al bajo ratio señal-ruido sumados a los efectos de la CTF. El filtro aplicado permite distinguir su forma algo mejor.

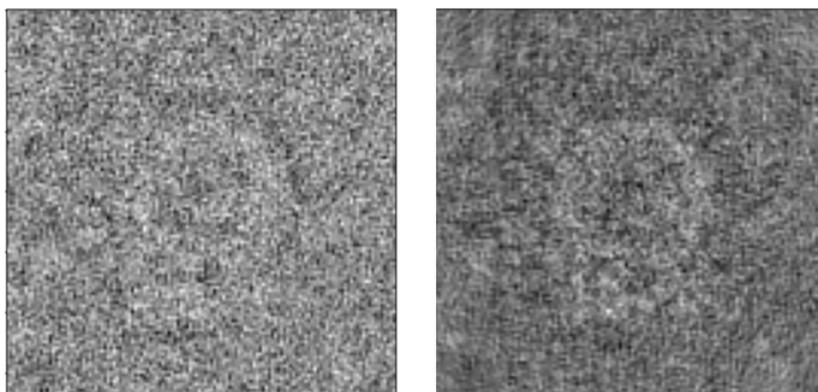


Figura 9: Imagen experimental de una molécula sin corregir la CTF (izquierda) y corrigiendo la CTF (derecha).

El último paso del preprocesamiento de los datos consiste en normalizar las imágenes. Para ello se resta a cada entrada de la matriz la media de todos los valores de la matriz y se divide por su desviación estándar.

4.2. Software Scipion

Scipion [29] es una plataforma de software que integra diversos algoritmos para el procesamiento de imágenes de microscopio electrónico. Cuenta con una interfaz sencilla que facilita el uso de estos algoritmos y la reproducibilidad, trazabilidad y el análisis del flujo de trabajo llevado a cabo.

En la figura 10 se muestra el flujo de trabajo de un proyecto de Scipion. Cada caja representa un programa o protocolo que recibe unos datos de entrada, los procesa y genera una salida. Cuando dos protocolos están unidos por una línea significa que el que está más abajo de los dos está utilizando como entrada la salida (o parte de ella) del protocolo al que está unido. Si no reciben entrada de otro protocolo porque, por ejemplo, carga datos externos, se une al cuadrado *Project*.

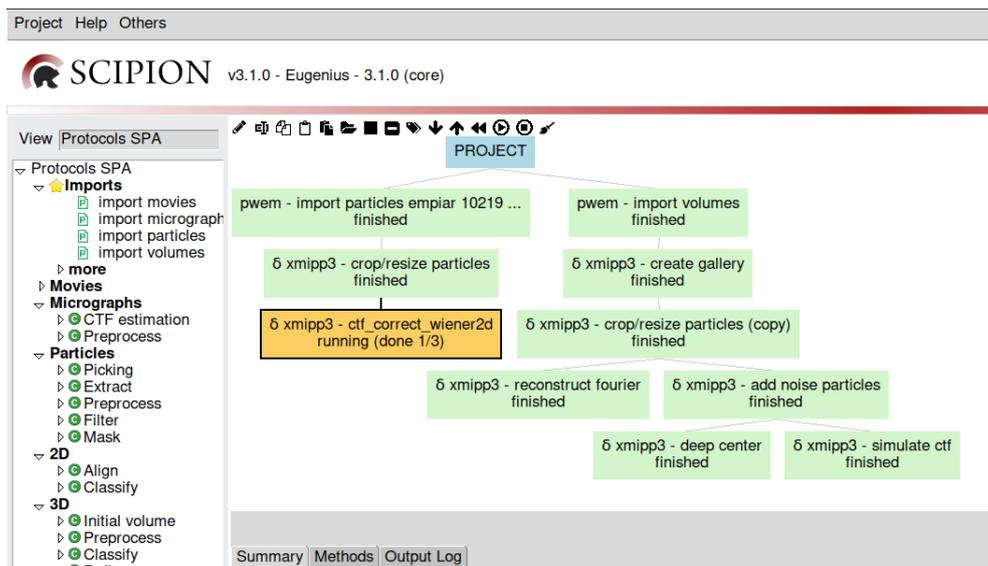


Figura 10: Ventana del programa Scipion con un ejemplo de un proyecto de procesamiento de imagen de microscopía electrónica.

Se ha llevado a cabo buena parte del preprocesamiento de los datos utilizando protocolos ya existentes en este software. El protocolo o programa *pwem - import particles* permite cargar las imágenes experimentales de los archivos *.mrc* entre otros y asignar sus valores a partir del archivo *.star*. Con el protocolo *xmipp3 - crop/resize particles* se ha reducido el tamaño en píxeles de las imágenes y con el protocolo *xmipp3 - ctf_correct_wiener2d* se ha aplicado el filtro de Wiener. Además, el protocolo *xmipp3 - create gallery* genera imágenes simuladas sin ruido de las proyecciones de una molécula distribuidas de la forma más homogénea posible a partir del mapa tridimensional de la misma.

Se ha trabajado en todo momento dentro de este software. Para ello se han creado

cuatro nuevos protocolos que se integran en el mismo: *xmipp3 - deep center*, *xmipp3 - deep center predict*, *xmipp3 - deep global assignment* y *xmipp3 - deep global assignment predict*. El primero de estos programas, *xmipp3 - deep center*, entrena una red neuronal para el centrado de moléculas a partir de un conjunto de imágenes previamente etiquetadas. *xmipp3 - deep center predict* predice a partir de la red entrenada el centro de las moléculas de un conjunto de imágenes. El funcionamiento y propósito de *xmipp3 - deep global assignment* y *xmipp3 - deep global assignment predict* son similares, pero en lugar de centrar las partículas, se estiman sus orientaciones. Los protocolos permiten al usuario ajustar algunos parámetros del entrenamiento como, por ejemplo, el número de épocas, la paciencia o la tasa de aprendizaje. La arquitectura de las redes neuronales y el proceso de aprendizaje se desarrollan en las secciones 5 y 6.

El lenguaje de programación empleado ha sido Python. Se ha utilizado la biblioteca NumPy para manejar las matrices de datos y realizar el preprocesamiento y TensorFlow para implementar y trabajar con las redes neuronales.

5. Centrado de partículas

Recordemos que el alineamiento de partículas consiste en asignar a cada imagen de una partícula o molécula individual los ángulos de proyección. Durante este proceso se establecen unos ejes de coordenadas sobre un mapa tridimensional a priori de la molécula entorno a los cuales se define la rotación necesaria para pasar de una posición inicial de referencia a la posición que aparece proyectada en la imagen experimental. Es común establecer el origen de coordenadas en el interior de la molécula, lo más centrado posible, por simplicidad. En adelante, llamaremos centro de la molécula a dicho origen de coordenadas.

Las proyecciones simuladas durante el proceso de alineamiento sitúan en el centro de la imagen la proyección del centro de la molécula. En las imágenes experimentales es frecuente encontrar la molécula desplazada cierta cantidad de píxeles respecto al centro. Los algoritmos de alineamiento deben tener esto en cuenta lo que supone un mayor coste computacional.

El objetivo del centrado de partículas es entrenar un modelo capaz de estimar la posición en píxeles del centro de la molécula que aparece en una imagen de microscopía electrónica. De esta forma, se podría restringir el espacio de búsqueda de los algoritmos acelerando con ello el alineamiento de partículas.

Se han utilizado imágenes de distintos tipos de molécula para entrenar un modelo capaz de centrar cualquier molécula sin necesidad de volver a ser entrenado ni tener imágenes previamente etiquetadas. Además, se ha llevado a cabo un re-entrenamiento del modelo con el fin de adaptarlo a una macromolécula en particular para observar los cambios en la precisión del mismo.

5.1. Metodología

5.1.1. Datos

Los conjuntos de datos escogidos de la base de datos de Empiar para el entrenamiento son los conjuntos EMPIAR-10391[30], EMPIAR-10482[31], EMPIAR-11233[32] y EMPIAR-10421[33]. En las figuras 11, 12, 13 y 14 se muestran los modelos 3D de las moléculas correspondientes a cada conjunto de datos empleado. Se ha procurado escoger moléculas de distinto tipo y forma con el objetivo de obtener una mayor generalización.

Una vez entrenada la red, se ha utilizado el conjunto de datos EMPIAR-10330[38] con imágenes experimentales pertenecientes a la molécula de la figura 15 para probar su funcionamiento y capacidad de generalización.

5.1.2. Arquitectura de la red

La red utilizada es una red convolucional clásica sencilla. En la figura 16 se muestra un esquema de la arquitectura de esta red. Recibe como entrada tensores de 128x128 y los hace pasar por filtros convolucionales 3x3. El número de filtros aumenta según se reduce la dimensión de los tensores con el objetivo de conservar la información

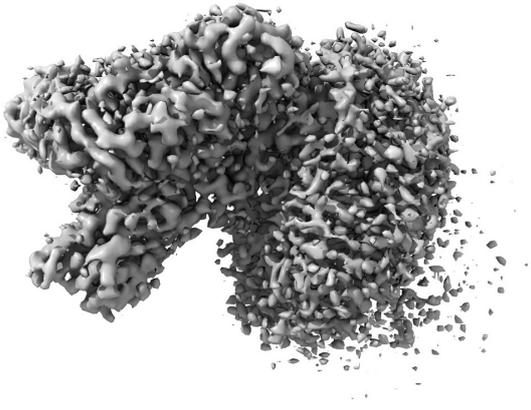


Figura 11: Imagen extraída de <https://www.ebi.ac.uk/emdb/EMD-21600>. La molécula se corresponde con la Arabinoxiltransferasa AftD de una micobacteria, una enzima presente en la pared celular de algunas micobacterias como la *Mycobacterium tuberculosis* [34].

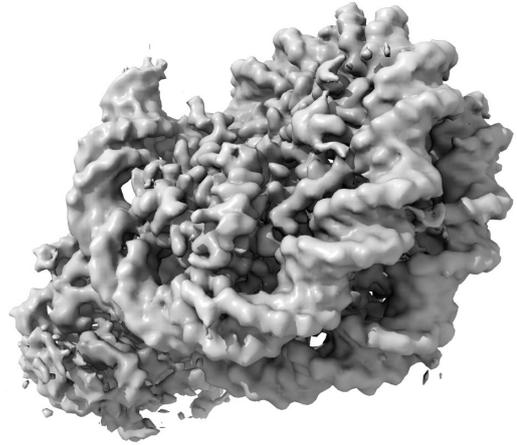


Figura 12: Imagen extraída de <https://www.ebi.ac.uk/emdb/EMD-9718>. Se trata de un complejo celular en el que interaccionan una enzima ISWI y un nucleosoma [35].

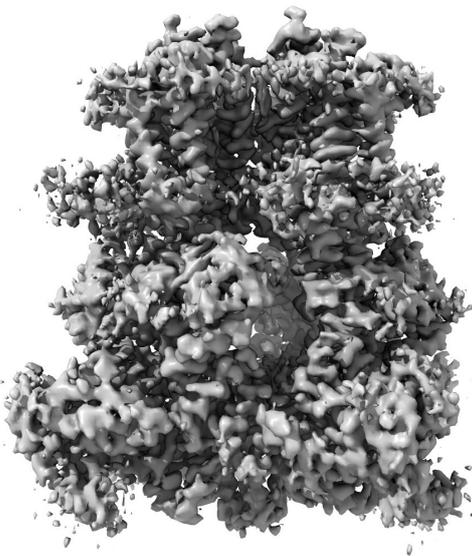


Figura 13: Imagen extraída de <https://www.ebi.ac.uk/emdb/EMD-9718>. La macromolécula que se muestra es el TRPM8, una proteína que permite la entrada de iones de sodio y calcio a las células [36].

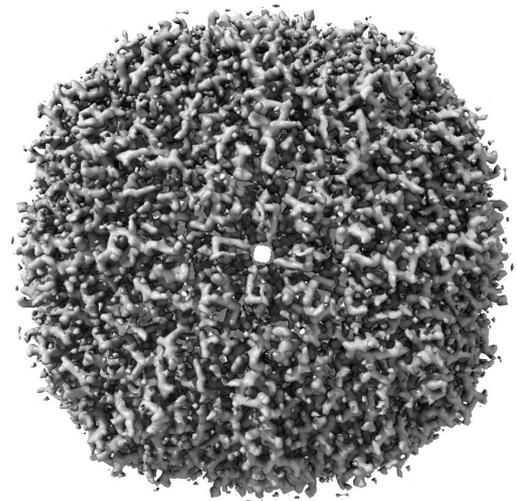


Figura 14: Imagen extraída de <https://www.ebi.ac.uk/emdb/EMD-21951>. Modelo tridimensional de la apoferritina humana, es la capa exterior de la ferritina, proteína de las células encargada de almacenar y liberar el hierro [37].



Figura 15: Imagen extraída de <https://www.ebi.ac.uk/emdb/EMD-20806>. La Chloroquine Resistance Transporter (PfCRT) del Plasmodium es una proteína presente en el parásito causante de la malaria que le confiere resistencia a ciertos tratamientos [39].

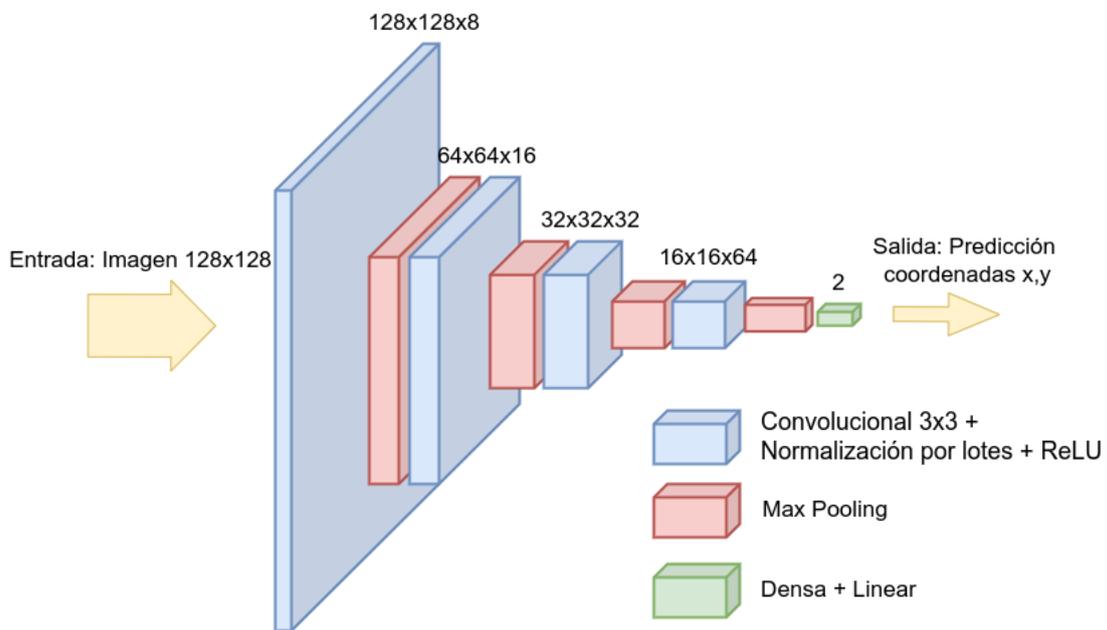


Figura 16: Arquitectura de la CNN propuesta para el centrado de partículas.

esencial. La última capa de MaxPooling se conecta con una capa densa de salida de dos neuronas mediante una capa de aplanado. El valor de la activación de estas dos últimas neuronas representa las coordenadas x e y predichas para el centro de la molécula en la imagen de entrada.

Esta red tiene un total de 33.058 parámetros entrenables. Es por tanto una red relativamente pequeña pero experimentalmente ha resultado suficiente. Pruebas con redes con más parámetros y arquitecturas más complejas han derivado en resultados similares a los que hemos obtenido con esta red y que se presentan más adelante.

5.1.3. Entrenamiento

Los datos de entrada se dividen de manera aleatoria entre conjunto de entrenamiento y de validación en una proporción 80-20. El entrenamiento de la red se realiza a partir de lotes. En cada época, en lugar de recibir el conjunto de entrenamiento completo, la red recibe los datos iterativamente en lotes más pequeños, los procesa y ajusta sus pesos. Esta técnica reduce el uso de memoria ya que permite procesar solo un lote a la vez, lo que reduce los requisitos de memoria y permite entrenar modelos más grandes. También mejora la capacidad de generalización del modelo por el carácter aleatorio de los lotes.

En cada época, se ha llevado a cabo una aumentación de datos consistente en desplazar cada imagen un número aleatorio de píxeles en horizontal y vertical. Las imágenes se reetiquetaban sumando el desplazamiento introducido.

Los principales objetivos de la aumentación de datos son mejorar la generalización del modelo y reducir el sobreajuste. Al aumentar el conjunto de entrenamiento de esta manera, se reduce también el número de partículas que es necesario alinear en la práctica, ganando en tiempo de cómputo. Además, se expone al modelo a nuevos ejemplos que puede no estén en el conjunto de entrenamiento como desplazamientos más altos o nuevas orientaciones.

Típicamente se pueden encontrar desplazamientos de unos 10 píxeles respecto al centro en las imágenes experimentales. Teniendo esto en cuenta se ha decidido aplicar un desplazamiento aleatorio con distribución uniforme entre -15 y 15 píxeles en ambos ejes a las imágenes.

El número de épocas del entrenamiento se ha establecido en 100. No obstante, se ha introducido un parámetro de paciencia de manera que si el error sobre el conjunto de validación no mejora después del número de épocas establecido, el entrenamiento finaliza. Se ha escogido una paciencia de 5 épocas.

Una vez finalizado el entrenamiento, se guarda el estado del modelo con el que se haya obtenido un menor error sobre el conjunto de validación durante el entrenamiento.

El ajuste de los pesos de la red se ha realizado mediante un optimizador tipo Adam[40] con una tasa de aprendizaje de 0,001. Se ha tomado como función de coste la norma L1. Se compara, por lo tanto, la posición del centro real (x, y) con la predicha (x', y') y la pérdida es $d = |x' - x| + |y' - y|$.

5.1.4. Transfer Learning

La transferencia de aprendizaje (o transfer learning en inglés) consiste en aprovechar un modelo ya entrenado para entrenarlo en una nueva tarea diferente en lugar de entrenar un modelo desde cero. Esto permite transferir las características aprendidas

que pueden ser útiles para el nuevo problema.

La idea es tomar el modelo entrenado para centrar cualquier macromolécula y ajustar sus pesos para especializarlo en una macromolécula concreta. Esto, en principio, debería suponer una mejora en la precisión del modelo al trabajar con las imágenes de esta macromolécula.

Sin embargo, para poder realizar transferencia de aprendizaje, es necesario etiquetar previamente un subconjunto de las imágenes experimentales con otro método para poder ser utilizadas durante el entrenamiento.

5.1.5. Consenso de modelos

Confiar en un único modelo para realizar predicciones puede tener varios inconvenientes. Por un lado, sólo es posible tener una idea global de la bondad de las predicciones a partir de los resultados de validación y test, pero no permite realizar una valoración individual de cada predicción en concreto. Por ejemplo, a partir de los resultados sobre el conjunto de test, si se ha realizado correctamente, en la práctica cabría esperar que la distribución del error en las predicciones fuese similar a la obtenida en el test, pero se conoce muy poco sobre el error de una predicción en concreto. Además, se tiene riesgo de sobreajuste o de falta de robustez y es sensible a la inicialización y al conjunto de entrenamiento.

Para tratar de aplacar estos inconvenientes que pueden surgir, se propone entrenar múltiples modelos independientes y luego combinar sus predicciones de alguna manera para obtener una predicción final más precisa y robusta. En concreto, la idea es dividir el conjunto de entrenamiento en subconjuntos que pueden ser no disjuntos pero lo suficientemente diferenciados y entrenar con cada uno un modelo.

Una vez se tiene un cierto número de modelos entrenados, se realizan las predicciones con cada uno de ellos. Para cada imagen se calcula la predicción promedio como el promedio en x y en y de todas las predicciones. Se compara este promedio con cada predicción individual. Con el fin de detectar posibles predicciones anómalas que puedan afectar al promedio final se establece una tolerancia. Si la distancia al promedio de la predicción más alejada es mayor que el valor de la tolerancia, se descarta dicha predicción y se recalcula la media de las predicciones restantes. Este proceso se repite para encontrar más datos atípicos hasta que todas las predicciones estén dentro de la tolerancia o el número de predicciones eliminadas sea más alto que un cierto umbral. En el primer caso la predicción final es el último promedio calculado. En el segundo caso, se considera que no existe un consenso entre las predicciones y la imagen se descarta porque el riesgo de error en la predicción es alto.

5.2. Resultados

Se ha tomado un subconjunto aleatorio compuesto por 40.000 imágenes, 10.000 imágenes de cada conjunto de datos de las moléculas para el entrenamiento.

Con estas 40.000 imágenes se han entrenado 5 redes neuronales con la arquitectura

indicada, de manera que cada una de ellas ha sido entrenada con 8.000 imágenes, utilizando 2.000 del resto de imágenes para validación.

En la figura 17 se muestra la monitorización del entrenamiento de uno de los modelos. Se observa una fuerte caída del valor de la función de pérdida durante las primeras etapas del entrenamiento que se va relajando. No se llega a completar las 100 etapas debido a que el entrenamiento concluye en la época 33 al no haberse obtenido una mejora respecto a la pérdida obtenida en el conjunto de validación en la época 28. Aunque el error de validación es algo superior al de entrenamiento como suele ser habitual no se observa un claro sobreajuste. Esto es gracias al parámetro de pérdida, con el que se detiene el entrenamiento antes de que se produzca, y a las distintas técnicas aplicadas para evitarlo, como la normalización por bloques o la aumentación de datos.

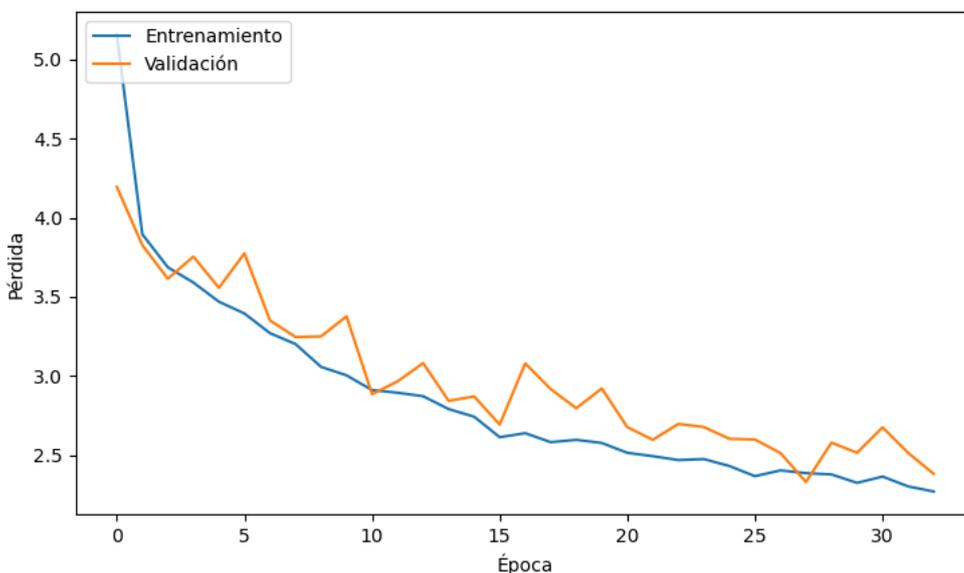


Figura 17: Entrenamiento de una de las redes para el centrado de moléculas.

Una vez entrenados, los modelos se han probado sobre los conjuntos de imágenes completos de los que se extrajo el conjunto de entrenamiento. Cabe esperar que los resultados sean similares a los obtenidos sobre el conjunto de validación. La idea es observar si existe alguna diferencia apreciable entre la precisión de las predicciones llevadas a cabo sobre las imágenes correspondientes a distintas moléculas que puedan tener que ver con su forma o con el ruido de las imágenes. También se comparan las predicciones llevadas a cabo por modelos individuales con la predicción conjunta de los cinco modelos. En este último caso se ha establecido una tolerancia de 3 píxeles y un mínimo de 3 (de un máximo de 5) predicciones dentro del radio marcado por dicha tolerancia. En la tabla 1 se recogen los resultados obtenidos.

Por último, a modo de test se ha utilizado el modelo entrenado para realizar predicciones sobre las imágenes del conjunto de datos EMPIAR-10330. A continuación se resumen los resultados obtenidos:

- Modelo entrenado: error medio 1.84 píxeles y 16/16905 imágenes descartadas

Molécula	Un modelo	Cinco modelos		Cinco modelos + Transferencia	
	Error Medio	Error Medio	Descartadas	Error Medio	Descartadas
10391	2,89	2,33	0/37.797	2,20	27/37.797
10482	2,49	1,89	947/166.165	1,67	357/166.165
11233	3.31	2,49	81/41.669	2,6	371/41.669
10421	1.78	1,92	416/100.000	1,89	443/100.000

Cuadro 1: Error medio y número de partículas descartadas sobre el total en las predicciones del centrado de partículas para cada molécula.

- Modelo entrenado + Transfer learning: error medio 1,95 píxeles y 14/16.905 imágenes descartadas
- Nuevo modelo entrenado desde 0 con 10.000 imágenes de esta molécula: error medio 2,38 píxeles y 261/16.905 imágenes descartadas.

Los resultados sobre la nueva molécula resultan similares a los obtenidos sobre las moléculas utilizadas en el entrenamiento por lo que el modelo ha demostrado capacidad de generalización en este caso. Además, se ha obtenido un error medio con el modelo entrenado inferior al obtenido entrenando un modelo desde cero específico para la molécula particular. Esto es positivo ya que el objetivo de haber entrenado un modelo universal era precisamente no necesitar entrenar nuevos modelos desde cero, evitando tener que etiquetar los datos y el proceso de entrenamiento.

6. Estimación angular

Denominamos estimación angular a la parte del alineamiento de partículas que consiste en estimar la orientación de la molécula en cada imagen de microscopio electrónico. Esto es equivalente a calcular los tres ángulos de proyección que definen la rotación que debe realizar la molécula desde la proyección de referencia hasta la proyección que aparece en la imagen.

Trabajar con partículas centradas en la imagen facilita el entrenamiento y predicción de los modelos. Por eso, en este trabajo planteamos la estimación angular como un paso posterior al centrado de partículas, es decir, en primer lugar se estima dónde está el centro de la partícula en la foto para, después, estimar los ángulos de proyección.

Como las proyecciones de una molécula dependen completamente de cómo sea esta, en lugar de tratar de entrenar un modelo universal a partir de imágenes de moléculas distintas como hemos hecho en el caso del centrado de partículas, se ha decidido entrenar modelos individuales. Dado un conjunto de imágenes experimentales, el procedimiento a seguir consiste en etiquetar primero un subconjunto de estas imágenes mediante algún algoritmo tradicional de alineamiento con el fin de entrenar con ellas un modelo que sirva finalmente para realizar predicciones sobre el resto de imágenes experimentales.

6.1. Metodología

Con el fin de resolver el problema de la estimación angular aplicando redes neuronales se ha llevado a cabo una serie de aproximaciones al problema a partir de los cuales se ha evidenciado la necesidad de emplear una correcta representación angular así como de escoger la arquitectura y aplicar un tratamiento especial en el caso de moléculas con simetría. Se pretende mostrar el proceso de razonamiento llevado a cabo y, para ello, se presentan las distintas opciones planteadas. En el apartado Resultados de esta sección se comparan dichos planteamientos aplicados al problema con el fin de justificar el finalmente escogido.

6.1.1. Datos

Se han utilizado los mismos conjuntos de datos que se usaron para el centrado de partículas. Es importante destacar la simetría que tienen estas moléculas ya que en primer lugar se han tratado moléculas sin simetría (simetría $C1$). De las 5 moléculas presentadas, aquellas que no cuentan con simetría rotacional son las correspondientes a los conjuntos de imágenes EMPIAR-10391, EMPIAR-10482 y EMPIAR-10330. La molécula correspondiente a EMPIAR-11233 tiene simetría cíclica de orden 4 entorno a un eje, simetría $C4$, mientras que la de EMPIAR-10421, la apoferritina, tiene simetría octaédrica, O .

Además de estos conjuntos de imágenes experimentales, se ha recurrido a la simulación de imágenes a partir del mapa tridimensional de la molécula. El mapa utilizado con este propósito se han obtenido de la base de datos EMBD (<https://www.ebi.ac.uk/emdb/>):

(//www.ebi.ac.uk/emdb/), en concreto es el mapa EMD-9718 que se corresponde con la misma molécula que el conjunto de imágenes EMPIAR-10482.

6.1.2. Arquitectura de la red

Se comenzó trabajando con una red convolucional clásica como la que se ha utilizado para el centrado de partículas. Se probó con distintas arquitecturas, variando el número de capas convolucionales y filtros e introduciendo o quitando capas de regularización y normalización. A partir de un cierto número de parámetros, los resultados no variaban mucho de unas redes a otras. La mejora sustancial en cuanto a la precisión se produjo al introducir las conexiones residuales.

La precisión de las distintas redes residuales probadas, todas basadas en la arquitectura de la ResNet-34 con capas de normalización, ha resultado ser muy similar en todos los casos. Por lo tanto, se ha optado por simplificar todo lo posible la red en cuanto a profundidad. Se ha utilizado el número de filtros en cada capa con los que mejores resultados se han obtenido. La arquitectura de la red con la que se ha trabajado finalmente se muestra en la figura 18

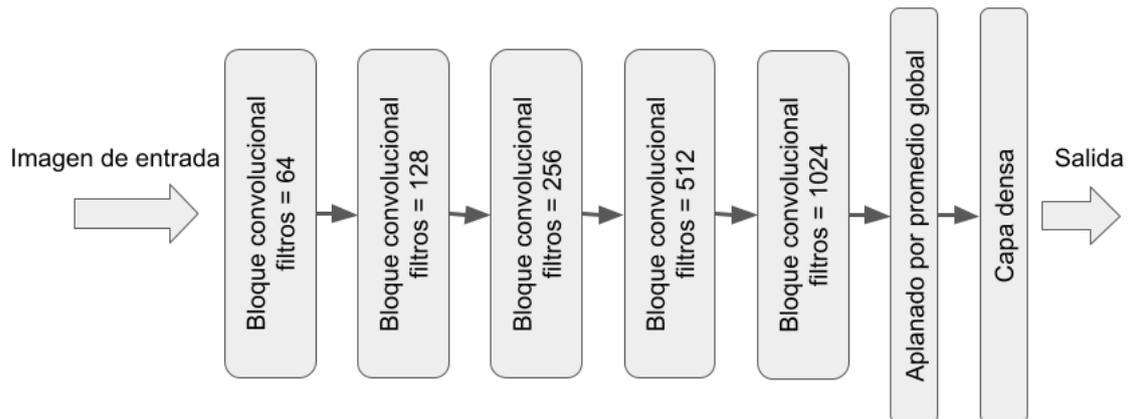


Figura 18: Esquema de la arquitectura de la red residual utilizada para la estimación angular.

6.1.3. Distancia angular

Dadas dos rotaciones del grupo $SO(3)$, \mathbf{P} y \mathbf{Q} , la rotación necesaria para pasar de \mathbf{Q} a \mathbf{P} es equivalente a regresar al origen desde \mathbf{Q} , con la rotación \mathbf{Q}^t , y después realizar la rotación \mathbf{P} . La matriz de rotación que representa la diferencia entre ambas viene dada entonces por $\mathbf{R} = \mathbf{P}\mathbf{Q}^t$. La distancia $d_{SO(3)}$ entre las dos rotaciones es el ángulo de la rotación respecto al eje fijado por \mathbf{R} . Dado que \mathbf{R} es una matriz ortogonal, su traza no depende de la base escogida. Por lo tanto, el ángulo de rotación se puede calcular a partir de la traza de \mathbf{R} :

$$\text{tr}\mathbf{R} = 1 + 2 \cos \Theta \implies \Theta = \arccos \frac{\text{tr}\mathbf{R} - 1}{2}.$$

Esta distancia, a la que nos referiremos como distancia angular es la que se ha utilizado para calcular la diferencia entre las rotaciones predichas y las reales.

Del mismo modo se podría plantear emplear la distancia angular como función de coste durante el entrenamiento de la red. No obstante, el cálculo de Θ puede resultar numéricamente impreciso debido a que la derivada de la función \arccos tiende a infinito cuando Θ tiende a 0. La serie de Taylor de \arccos es

$$\arccos(x) = \frac{\pi}{2} - x - \frac{x^3}{6} - \frac{3x^5}{40} + O(x^7) \dots$$

Se ha truncado la serie en el término de grado 1 y se ha utilizado $-\text{tr}\mathbf{R} = -\text{tr}\mathbf{P}\mathbf{Q}^t$ como función de coste. La ventaja teórica de esta función de coste sobre otras comunes como las normas l1 y l2 es que se adapta mejor que estas a la topología del espacio sobre el que se trabaja. Sin embargo, en la práctica se han obtenido mejores resultados aplicando la norma l2 y por ello se ha decidido utilizar esta.

En el caso de moléculas con simetría, al calcular la distancia angular hay que tener en cuenta la existencia de distintos puntos equivalentes. De esta forma, la distancia se deberá calcular como la distancia angular al punto equivalente más cercano. Para ello, sean $S = \{\mathbf{S}_i\}$ las matrices de rotación del grupo de simetría asociado a la molécula, la distancia angular es:

$$\Theta(\mathbf{P}, \mathbf{Q}, S) = \min \left\{ \arccos \frac{\text{tr}\mathbf{P}\mathbf{S}_i\mathbf{Q}^t - 1}{2} : \mathbf{S}_i \in S \right\}.$$

6.1.4. Entrenamiento

La estrategia de entrenamiento es similar a la empleada para el centrado de partículas. Los datos de entrenamiento y validación se dividen en una proporción 80-20. De nuevo, se ha utilizado el entrenamiento por lotes. Durante el entrenamiento, se va guardando el estado de la red que mejores resultados arroja sobre el conjunto de validación y se establece un parámetro de paciencia.

La aumentación de datos ha consistido, en primer lugar, en desplazar las imágenes una cantidad aleatoria de píxeles en horizontal y vertical. Con el fin de obtener mejores resultados, se supone que las imágenes ya están previamente centradas de manera que no es necesario aplicar desplazamientos grandes. Se ha aplicado un desplazamiento máximo de 2 píxeles en cada dirección. Además de desplazar las imágenes se aplica una rotación aleatoria entre 0 y 360°. Se actualiza la etiqueta de la imagen añadiendo el ángulo rotado al ángulo de Euler ψ . La rotación se realiza mediante la función *rotate* de la biblioteca *scipy* utilizando el modo *'reflect'* que rellena los píxeles que se quedan vacíos al rotar la imagen cuadrada reflejando los bordes del último píxel [41]. En la figura 19 se muestra cómo esta función rota una imagen. Se utiliza el modo *'reflect'* porque en la práctica ha resultado ser el que menos artificios introduce en la nueva imagen. La existencia de artificios como

distintos tipos de formas en lugar del ruido gaussiano de la imagen original en los nuevos bordes afectan negativamente al entrenamiento.

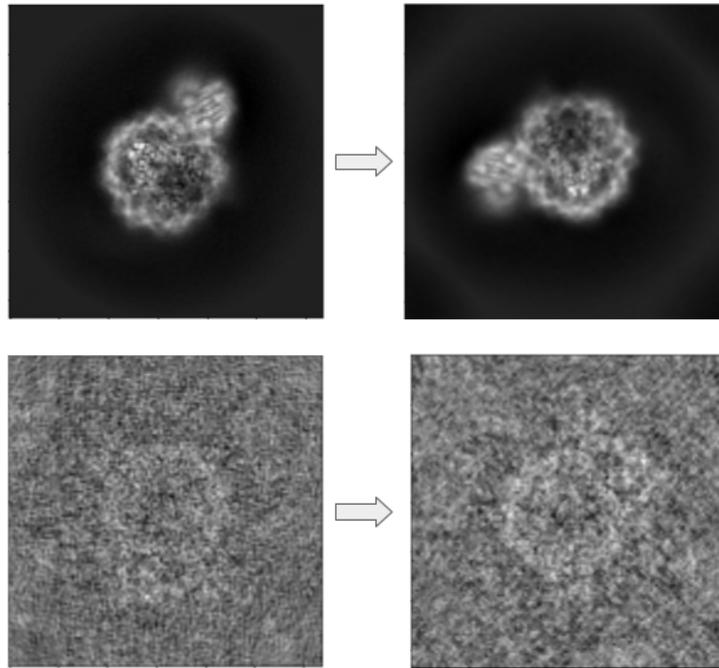


Figura 19: Efecto de la rotación sobre dos imágenes, la primera simulada y la segunda experimental, que se aplica durante la aumentación de datos. En ambos casos la rotación es de 135° .

6.1.5. Consenso de modelos

Al igual que se hizo al trabajar con el problema del centrado de partículas, se propone realizar un ensamblaje de modelos consistente en entrenar 5 redes con distintas muestras del conjunto de datos con el fin de mejorar la precisión de los resultados y, sobretodo, poder descartar el máximo número posible de predicciones erróneas.

Las rotaciones predichas para una imagen son convertidas a cuaterniones con el fin de calcular fácilmente una aproximación de la rotación media de las cinco predicciones. Se calcula la distancia angular de cada predicción a esta media y se descarta la más alejada si su distancia es mayor a cierta tolerancia. Se repite el proceso hasta que, o bien todas las predicciones están dentro de la tolerancia, en cuyo caso se utiliza la media como predicción del modelo final, o bien se han descartado demasiadas predicciones, en cuyo caso se descartaría también la imagen por no existir un consenso.

6.1.6. Moléculas con simetría

En el caso de moléculas con simetría rotacional se ha implementado el método descrito en el artículo [25] que se ha descrito más arriba. Como la función Map'

separa las rotaciones por dominios continuos, se debe entrenar una red para cada dominio. Con el fin de conocer a qué dominio pertenece cierta imagen y utilizar el regresor entrenado adecuado, se entrena un clasificador. De esta forma, la predicción sobre una imagen se hace en dos pasos: el clasificador predice en primer lugar cuál es el dominio correspondiente, y en función del mismo, se escoge una de las redes entrenadas que realiza la predicción final.

Tanto el clasificador como los predictores son redes residuales con la misma arquitectura presentada en la figura 18. En el clasificador cambia la capa densa de salida, que en lugar de tener 6 neuronas y una función de activación lineal, tiene el mismo número de neuronas que clases (o dominios en este caso) posibles y la función de activación *SoftMax* [42]. Cada neurona representa una clase y el valor de su salida es una medida de la probabilidad de que la imagen pertenezca a dicha clase. La función de pérdida utilizada para el clasificador es la entropía cruzada categórica [43].

Con el fin de evitar sesgos, se entrena el clasificador realizando un muestreo balanceado con el mismo número de imágenes pertenecientes a cada clase. Para ello se realiza un sobremuestreo de las clases con menos imágenes en el conjunto de entrenamiento. Cada regresor es entrenado con las imágenes disponibles de la clase correspondiente a dicho regresor.

Si se entrenan varios modelos, en lugar de, como en el caso de moléculas sin simetría, realizar las predicciones individuales de cada modelo para después buscar un valor medio a través del consenso entre modelos, se ha decidido dividir este consenso en dos pasos. En primer lugar, los clasificadores de cada modelo realizan sus predicciones. Se toma como clase predicha final por votación, es decir, se suman el valor que da cada clasificador a cada clase y aquella con una suma más alta es la clase predicha. Conocida la clase, se realizan las predicciones utilizando la red de cada modelo correspondiente a la clase predicha y se realiza el consenso entre las predicciones como se ha hecho en el caso de moléculas sin simetría.

En caso de que moléculas con simetría C_n , se ha probado también el método propuesto para este tipo de moléculas. Para ello se han reetiquetado las imágenes identificando orientaciones equivalentes y cambiando los valores del ángulo ϕ para trabajar en toda la circunferencia $[0, 2\pi)$. El resto del proceso de entrenamiento es idéntico al seguido para moléculas sin simetría.

6.2. Resultados

6.2.1. Representación angular

Se han entrenado dos redes neuronales con la misma arquitectura a partir de 2000 imágenes simuladas de la molécula EMPIAR-10482. Una de las redes se ha entrenado con las imágenes etiquetadas con los senos y cosenos de los ángulos de Euler ($\sin \phi, \cos \phi, \sin \theta, \cos \theta, \sin \psi, \cos \psi$) y, la otra red, con la representación 6D, con el fin de exponer las diferencias entre ambas representaciones. Se han empleado imágenes simuladas en lugar de las experimentales porque permiten analizar mejor en qué puntos falla la representación con ángulos de Euler: al ser imágenes ideales sin ruido, la precisión del modelo es muy alta y se observa mejor los errores fruto de una mala

elección de la representación.

Se ha entrenado durante 100 épocas con lotes de 32 imágenes y una paciencia de 5 épocas. El entrenamiento de la red entrenada con imágenes etiquetadas a partir de los ángulos de Euler finalizó en la época 32 y la entrenada empleando representación 6D convergió en 65 épocas.

Una vez entrenadas, las redes se han probado mediante 10.000 imágenes simuladas. En la figura 20 se representa el histograma del error angular en grados, es decir, la distancia angular entre la rotación predicha y la real. La mayoría de los errores son inferiores a 20° pero existen imágenes para las cuales la predicción ha resultado mucho peor.

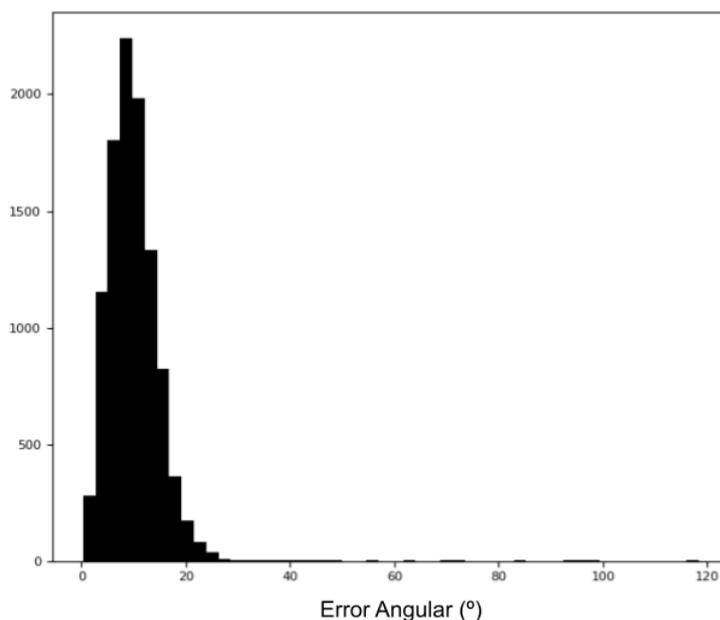


Figura 20: Histograma del error angular obtenido al predecir sobre el conjunto de imágenes simuladas utilizando una representación angular basada en los ángulos de Euler.

La figura 21 representa el error en la predicción frente al ángulo θ real. Este gráfico permite observar que los errores superiores a 20° se corresponden a proyecciones con ángulo de rotación θ cercano a 0 y a 180° . Esto se debe a las ambigüedad que surge del bloqueo del cardán en la representación con ángulos de Euler.

En la figura 22 se comprueba que empleando la representación 6D no surgen estos errores altos. Además, el error angular medio es mucho menor, $3,01^\circ$ frente a los $9,95^\circ$ obtenidos utilizando la otra representación. La representación 6D es la más indicada en este caso y es la que se ha utilizado en adelante.

6.2.2. Moléculas sin simetría

Se presentan los resultados obtenidos sobre los conjuntos de datos correspondientes a moléculas sin simetría.

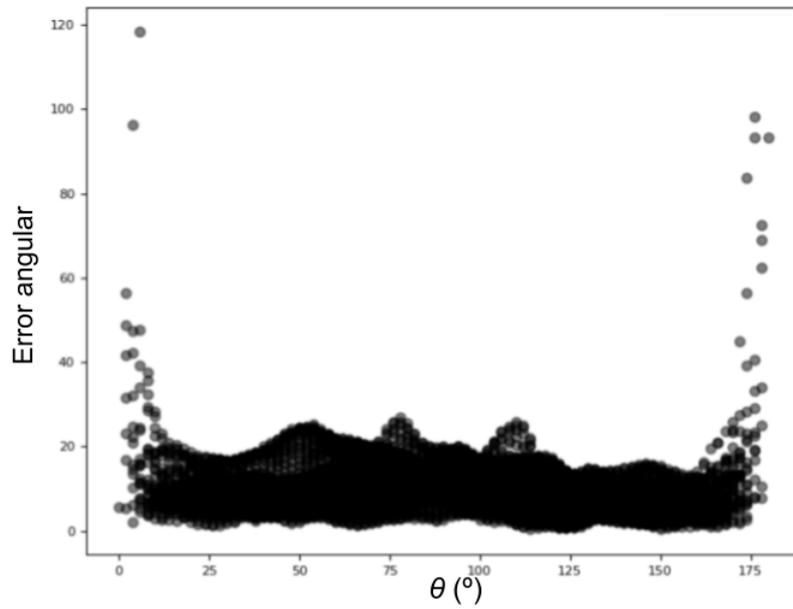


Figura 21: Error angular en la predicción de cada imagen respecto al ángulo θ correspondiente.

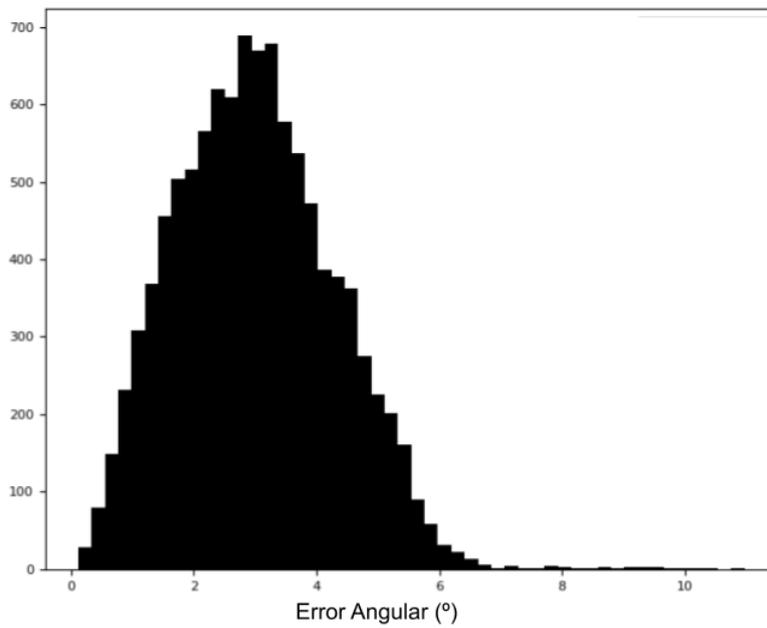


Figura 22: Histograma del error angular obtenido al predecir sobre el conjunto de imágenes simuladas utilizando la representación 6D.

EMPIAR-10391

Se han escogido de manera aleatoria 30.000 imágenes para el conjunto de entrenamiento y el de validación (24.000 y 6.000 respectivamente) y 10.000 para el conjunto de test de las 166.165 imágenes que tiene el conjunto de datos en total. Se ha toma-

do estos subconjuntos relativamente pequeños respecto del total porque uno de los objetivos es poder entrenar los modelos sin tener que etiquetar un gran número de imágenes.

Se han entrenado 5 redes neuronales utilizando para cada una de ellas 4.800 imágenes de entrenamiento y 1.200 de validación. El número máximo de épocas ha sido 100, la paciencia, 5, y la tasa de aprendizaje se ha establecido en 0,001. En la figura 23 se muestra la variación del error sobre el conjunto de entrenamiento y el de validación durante el entrenamiento de una de las redes.

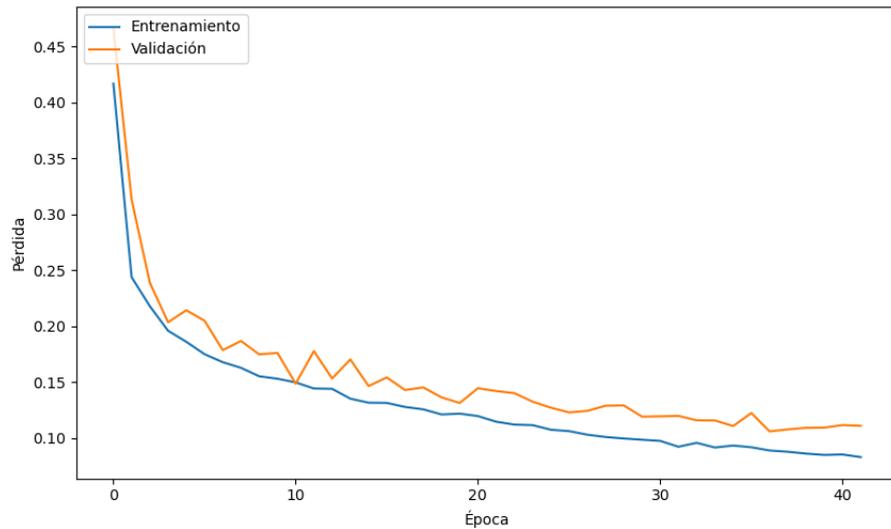


Figura 23: Entrenamiento de una de las redes neuronales.

Una vez entrenadas las redes, se ha realizado la predicción conjunta sobre las imágenes de test. La tolerancia angular se ha establecido en 15° y el número máximo de predicciones descartadas por imagen, en 2. El error o distancia angular media obtenida ha sido $13,30^\circ$ y se han descartado 2.836 imágenes de las 10.000 de test.

Siempre es interesante comparar la precisión de las predicciones utilizando las 5 redes en conjunto con la precisión que se obtiene utilizando una única red como modelo. En este último caso, el error angular medio obtenido resulta de $24,10^\circ$, por lo que el empleo de 5 modelos supone una importante mejoría. En la figura 24 se muestran los histogramas de los errores obtenidos en cada caso. Se observa que al utilizar varias redes ha sido posible descartar o mejorar la predicción de muchas de las imágenes sobre las que el modelo individual ha arrojado predicciones con error superior a 15° .

Con las predicciones angulares obtenidas, es posible realizar una reconstrucción aplicando el método de Fourier. En Scipion, se ha utilizado para ello el protocolo *xmipp3 - reconstruct Fourier* con una resolución de 6Å. También se ha reconstruido con las imágenes con la etiqueta original con la finalidad de comparar ambas. En la figura 25 se muestra en la parte superior varios cortes de los mapas tridimensionales obtenidos y en la parte inferior, una imagen de cada mapa. Como cabía esperar, el mapa tridimensional obtenido a partir de las predicciones del modelo entrenado está peor definido aunque mantiene la misma estructura. Sería necesario aplicar algún otro

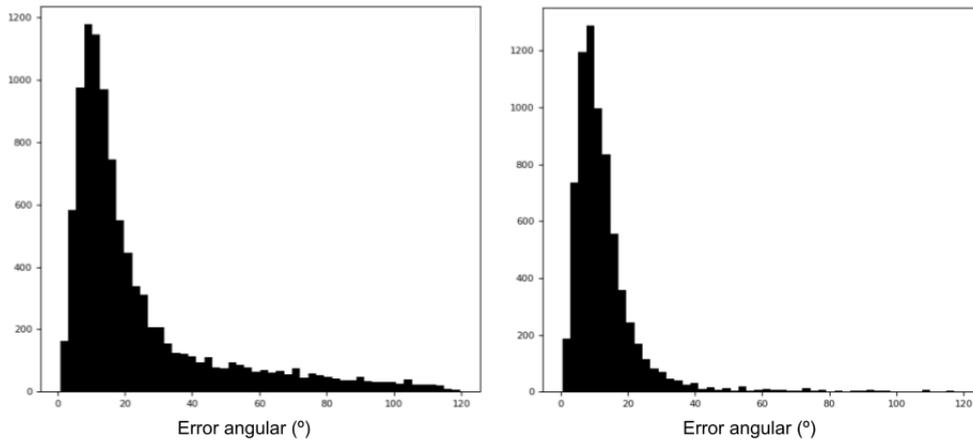


Figura 24: Histograma de los errores sobre el conjunto de test de las predicciones del modelo con una única red (izquierda) y del modelo que implementa 5 redes (derecha).

método como el expuesto en [8] para refinar esta reconstrucción.

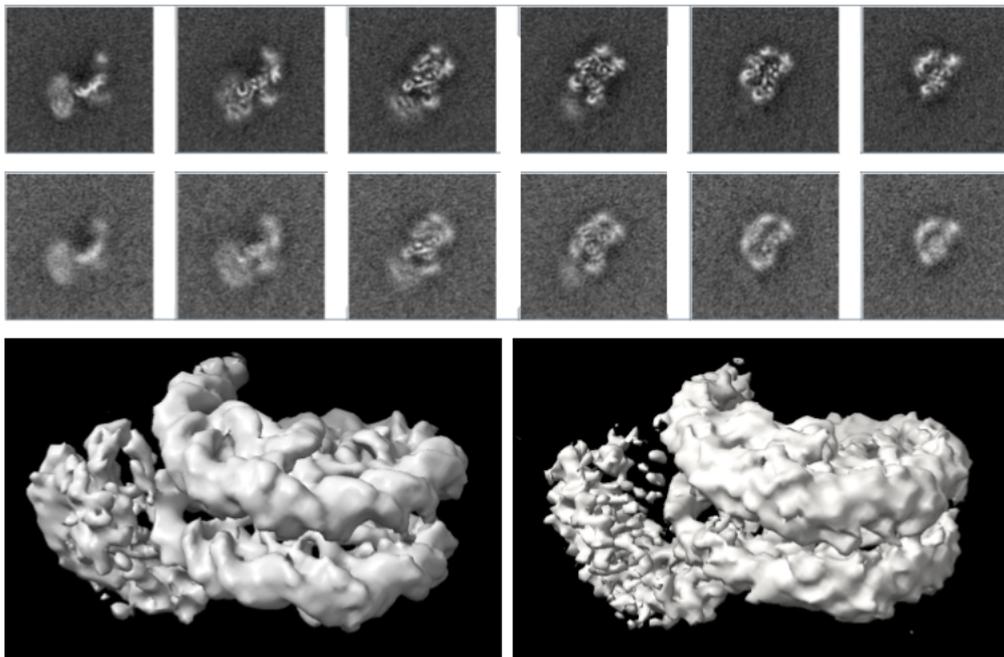


Figura 25: En la parte superior: cortes de la reconstrucción de la molécula con varios planos, la primera fila se corresponde con la reconstrucción a partir de las etiquetas originales y las imágenes de la segunda fila, con la reconstrucción a partir de las predicciones. Parte inferior: modelos 3D reconstruidos a partir de las orientaciones originales (izquierda) y de las predichas (derecha).

Siempre es interesante observar los mapas de activación de las capas de la red ya que muchas veces permite entender cómo trabaja. En la figura 26 se muestra los mapas de activación del primer bloque convolucional de una de las redes entrenadas cuando

es alimentada con la imagen experimental que aparece en la figura. En varios mapas se observa cómo la red ha sido capaz de evitar buena parte del ruido y recuperar al menos el contorno de la molécula en la imagen.

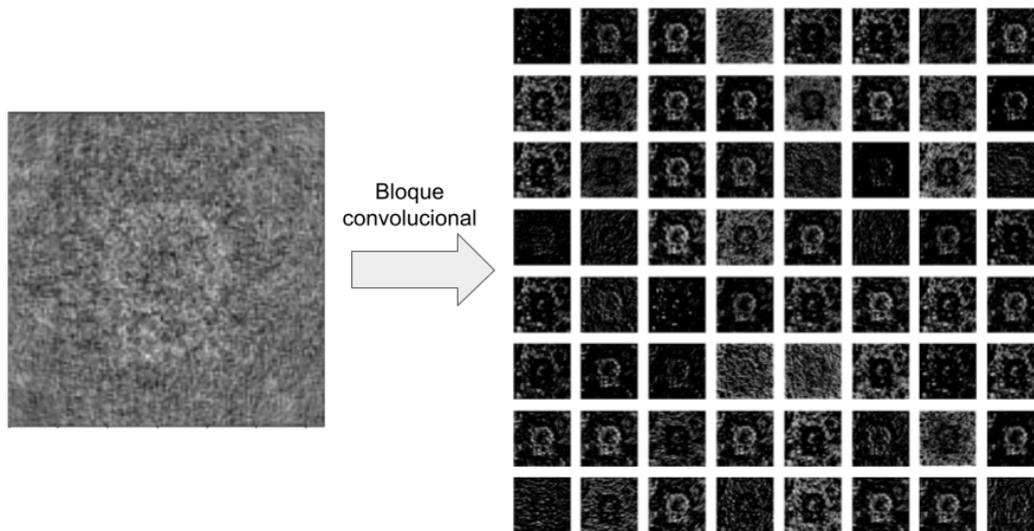


Figura 26: Mapas de activación obtenidos al hacer pasar la imagen de la izquierda por el primer bloque convolucional de una de las redes entrenadas.

EMPIAR-10482

Como el conjunto de imágenes total cuenta con solo 37.814 imágenes se ha decidido utilizar dos tercios de las imágenes para entrenamiento y validación y el tercio restante para test. Es interesante comprobar si de esta manera, con 4.033 imágenes para entrenar cada uno de los 5 modelos y 1.008 de validación es suficiente para obtener resultados comparables con los obtenidos con el conjunto EMPIAR-10391.

De nuevo, se ha entrenado durante un máximo de 100 épocas con una paciencia de 5 y una tasa de aprendizaje de 0,001. Se ha obtenido un error angular medio sobre el conjunto de test de $13,99^\circ$, similar al obtenido con el anterior conjunto de datos por lo que aparentemente el menor tamaño en el conjunto de entrenamiento no ha afectado negativamente a la precisión, aunque también habría que tener otros factores en cuenta como el ruido propio de las imágenes de uno u otro conjunto que podría hacer que el aprendizaje sobre uno de los dos resulte más sencillo. Se han descartado 3410 imágenes de las 12604 utilizadas para el test.

EMPIAR-10330

En este conjunto de datos, el número de imágenes disponibles es drásticamente inferior al de los anteriores: 16.905 imágenes. Se ha dividido el conjunto total tomando $2/3$ para entrenamiento y validación y $1/3$ para test. La idea de esto no es entrenar el mejor modelo posible para este conjunto de datos, para lo que se podría tomar un

80 % de las imágenes para entrenamiento, un 10 % para validación y el 10 % restante para test, sino comprobar la capacidad del modelo trabajando con un conjunto pequeño de imágenes.

En este caso, teniendo en cuenta el limitado conjunto de entrenamiento, se ha establecido una paciencia de 10 y se ha reducido la tasa de aprendizaje a 0,0001. Pese a esto, los resultados son mucho peores que en los casos anteriores: error medio de 40,4° y 4.404 de 5.635 imágenes descartadas.

Se ha entrenado una nueva red neuronal utilizando para ello un conjunto mayor con 7.000 imágenes, tamaño similar a los conjuntos utilizados para entrenar los modelos anteriores. Sin embargo, el error sobre el conjunto de test de las predicciones de la nueva red es 40,98°, mayor que los 24,10° obtenidos con la red individual del conjunto de datos EMPIAR-10391.

De estos resultados se puede deducir que la baja precisión del modelo entrenado no se debe únicamente al reducido número de imágenes disponibles, sino a otros factores que pueden afectar el entrenamiento y la predicción como la calidad de las imágenes en cuanto al ratio señal-ruido o la precisión con la que fueron etiquetadas.

6.2.3. Moléculas con simetría

EMPIAR-11233

La macromolécula correspondiente al conjunto de datos EMPIAR-11233 tiene simetría C_4 y, por lo tanto, tiene simetría rotacional entorno a un eje de orden 4. Se recuerda que en moléculas con este tipo de simetría es común establecer los ejes de referencia de manera que el eje Z coincida con el eje de simetría. Con esta referencia, las orientaciones descritas por los ángulos (ϕ, θ, ψ) y $(n\pi/2 + \phi, \theta, \psi)$ con $n = 1, 2, 3$ son equivalentes y las imágenes de sus proyecciones, iguales salvo ruido y desplazamientos.

Con el fin de ilustrar la necesidad de aplicar un tratamiento especial a las moléculas con simetrías se va a entrenar en primer lugar dos modelos con este conjunto de datos: el primero siguiendo la misma estrategia que se ha seguido con moléculas sin simetría y el segundo, cambiando la función de coste para calcular la distancia angular teniendo en cuenta las simetrías.

Se ha entrenado un modelo con cinco redes neuronales utilizando las dos estrategias tomando para ello 30.000 imágenes, 4.800 como conjunto de entrenamiento y las 1.200 imágenes restantes como conjunto de validación para cada red neuronal. El conjunto de test consta de 10.510 imágenes. Se ha establecido un máximo de 100 épocas, con una paciencia de 5 épocas y una tasa de aprendizaje igual a 0,001.

El error angular medio de las predicciones sobre el conjunto de test del modelo entrenado sin tener en cuenta la simetría es 60,04° y 9.896 descartadas del total de 10.510 imágenes con una tolerancia de 15°. El resultado es muy malo como se podría esperar.

En el caso del modelo entrenado adaptando la función de coste a la simetría, el error angular medio con una tolerancia de 15° es 17,85° y 1.607 imágenes descartadas. El valor del error y el número de imágenes descartadas son similares a los que se obtuvo

al trabajar con partículas sin simetría por lo que podría parecer que son resultados aceptables. Sin embargo, como se puede observar en la figura 27, el error en las predicciones para ciertos valores de ϕ aumenta. Las regiones donde el error es alto están separadas 90° y se deben a las ambigüedades provocada por la simetría. Esto hace que, aunque el error medio no sea excesivamente alto, las predicciones obtenidas no sirvan para reconstruir la molécula porque se pierde la información de parte de su estructura.

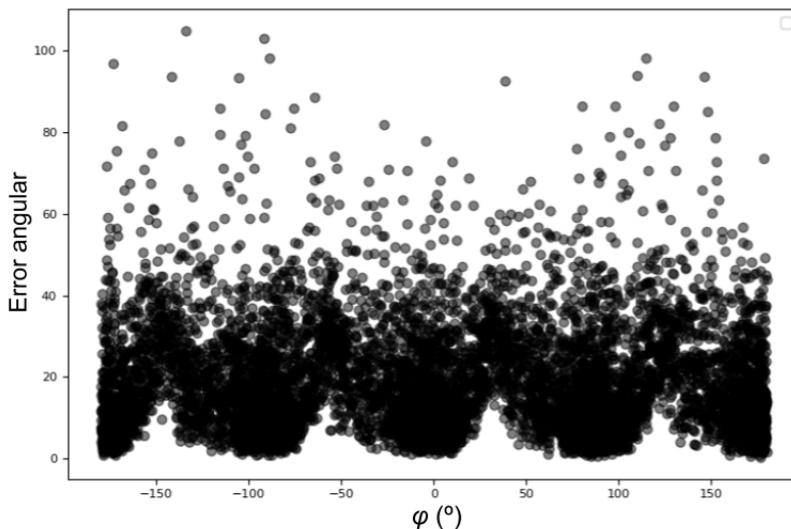


Figura 27: Error angular frente al ángulo de rotación ϕ obtenido en las predicciones sobre el conjunto de test.

Se ha probado a implementar el método del artículo [25] descrito más arriba que trata de resolver las ambigüedades producidas por la simetría. Como la molécula tiene un único eje, se clasifican las imágenes en 2 regiones y se entrena un predictor para cada una de ellas. Se necesita entrenar entonces 3 redes neuronales (un clasificador y los dos predictores) por modelo, y, de nuevo, se han entrenado 5 modelos. Los parámetros de entrenamiento son los mismos para clasificadores y predictores: máximo de 100 épocas, paciencia igual a 5 y tasa de entrenamiento de 0,001.

Con una tolerancia de 15° se ha obtenido un error medio de $14,32^\circ$ descartando 1.654 imágenes. En la figura 28 se comprueba que ahora el error se distribuye de manera más o menos homogénea respecto a ϕ .

Teniendo en cuenta estos resultados, parece que el método aplicado ha sido eficaz resolviendo las ambigüedades producidas por la simetría.

Con el fin de comparar ambos métodos para resolver las ambigüedades, se ha aplicado el último de los métodos descritos, que se propone utilizar para partículas C_n . Como la simetría es C_4 , este ha consistido en multiplicar por 4 el valor del ángulo ϕ de cada imagen y después aplicar la operación "módulo" para trabajar en el rango $[0, 2\pi)$ habiendo identificado entre sí las orientaciones equivalentes. Se han entrenado 5 modelos con los mismos parámetros que en los casos anteriores. El error medio obtenido es $10,73^\circ$, superando en precisión al modelo entrenado a partir del método del artículo.

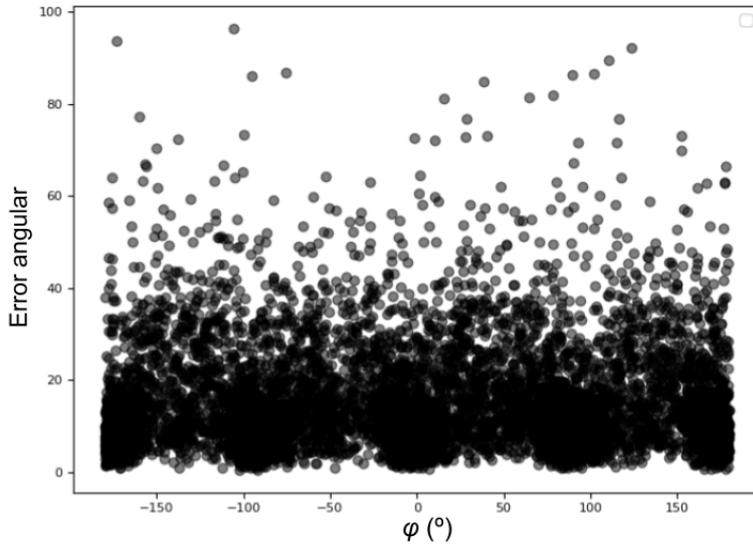


Figura 28: Error angular frente al ángulo de rotación ϕ obtenido en las predicciones sobre el conjunto de test aplicando el método para resolver las ambigüedades.

Se ha realizado una reconstrucción a partir de las imágenes utilizando las orientaciones predichas. En la figura 29 se comparan varias secciones transversales de las reconstrucciones llevadas a cabo con las de la reconstrucción obtenida con las orientaciones originales. Aunque pierde resolución respecto de esta última como es lógico, la reconstrucción realizada a partir de las predicciones del método propuesto es superior al resto.

EMPIAR-10421

La macromolécula correspondiente al conjunto EMPIAR-10421 es la apoferritina, con simetría octaédrica. Es posible generar las 24 rotaciones del grupo de simetría octaédrica a partir de un eje de rotación de orden 4 y otro eje de rotación de orden 3. Se han empleado las siguientes dos matrices generadoras:

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \quad \text{y} \quad \mathbf{B} = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

La matriz \mathbf{A} representa una rotación de ángulo $2\pi/3$ entorno al eje de dirección $(1, 1, 1)$ y la matriz \mathbf{B} representa una rotación entorno al eje de dirección $(0, 0, 1)$ de ángulo $\pi/2$.

Se ha aplicado el método del artículo. Este el caso de una simetría con dos ejes de rotación, de manera que las orientaciones de las imágenes se clasifican en cuatro dominios. Se han entrenado 5 modelos con un clasificador y cuatro regresores cada uno. Los parámetros de entrenamiento de las cinco redes neuronales son los mismos: máximo de 100 épocas, paciencia igual 5 y una tasa de aprendizaje de 0,001.

El conjunto de entrenamiento se compone de 32.000 imágenes y el de validación consta de 8.000 imágenes. Se ha calculado el dominio correspondiente a cada imagen y

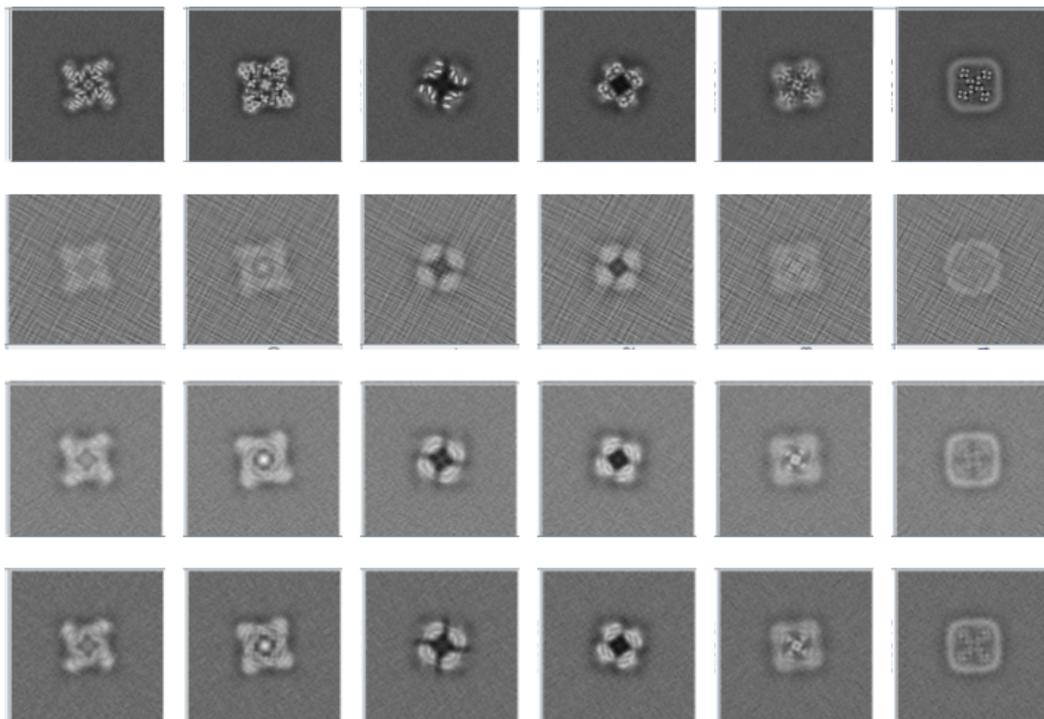


Figura 29: Cortes del mapa tridimensional reconstruido a partir de: orientaciones originales (primera fila); orientaciones predichas simplemente adaptando la función de coste a la simetría (segunda fila); orientaciones predichas aplicando el método del artículo (tercera fila); y orientaciones predichas aplicando el método propuesto para moléculas C_n (última fila).

se ha encontrado que el número de imágenes por dominio está muy descompensado: 13.662 imágenes pertenecen al primer dominio; 1.192, al segundo; 13.505, al tercero; y 8.310 corresponden al cuarto dominio. Esto puede afectar negativamente al entrenamiento de las redes neuronales correspondientes al cuarto y, sobretodo, al segundo dominio. Con el fin de no reducir todavía más el número de imágenes con las que se entrena estas redes se ha tomado como conjunto de entrenamiento de cada modelo un tercio aleatorio del conjunto de entrenamiento.

Una vez entrenados los modelos, se han realizado las predicciones sobre un conjunto de test de 10.000 imágenes. Como debido a las simetrías y las numerosas rotaciones equivalentes el rango de posibles orientaciones se reduce, se ha establecido una tolerancia de 5° en este caso. El error angular medio es $7,80^\circ$ y se descartan 2.598 imágenes. Este error es bajo, pero si se distingue por dominios, el error medio para los dominios 1 y 3, a los que pertenecían la mayoría de imágenes del conjunto de entrenamiento, es $6,27^\circ$ y $5,67^\circ$ respectivamente, mientras que el error al predecir sobre las imágenes pertenecientes a los dominios 2 y 4, es $24,56^\circ$ y $25,86^\circ$ respectivamente. Se obtiene además que el clasificador clasifica mal el 44,5% de las imágenes de test pertenecientes al dominio 4 y el 40,3% de las pertenecientes al dominio 2. En el caso de los dominios 1 y 3, el clasificador falla en el 14,8 y el 12,3% de las imágenes respectivamente.

Los resultados en este caso no son satisfactorios. Se requiere un análisis más en profundidad para poder determinar la razón de la diferencia de resultados por dominios. Una de las posibles explicaciones es que el número reducido de imágenes impide a las redes neuronales generalizar correctamente en esos dominios. También podría ser que el ruido haga más difícil distinguir entre las imágenes de un dominio que entre las de otro. Una última opción es que algún dominio no estuviese correctamente definido existiendo discontinuidades dentro del mismo aunque unos primeros resultados sobre imágenes simuladas limpias indican que estas discontinuidades o bien no existen o bien son pocas ya que se obtiene un error medio de $3,5^\circ$. Este análisis más exhaustivo no se ha llegado a hacer por falta de tiempo.

En todo caso, el método aplicado supone entrenar muchas redes para poder realizar predicciones provocando que sea muy lento. El tiempo de entrenamiento de los modelos con una única red se ha situado entorno a los 20 minutos, el tiempo de entrenamiento del clasificador y las 4 redes regresoras ha sido de unos 45 minutos. En total, si se entrenan 5 modelos para poder realizar el consenso entre predicciones, el tiempo de entrenamiento pasa de 1 hora y 20 minutos a 3 horas y media.

7. Conclusiones

Ha sido posible diseñar modelos de aprendizaje profundo capaces de llevar a cabo tareas de centrado de partículas y estimación de orientaciones sobre imágenes experimentales de moléculas de microscopía electrónica. Se ha demostrado con ello la capacidad de este tipo de arquitecturas para tratar con el ruido presente en las imágenes y realizar tareas de regresión.

Se han expuesto también los distintos problemas que surgieron durante la realización del trabajo y las soluciones propuestas. La representación 6D resuelve las discontinuidades que aparecen al utilizar otras representaciones más habituales como los cuaterniones o los ángulos de Euler y que dificultan el entrenamiento de las redes. La simetría presente en algunas moléculas producen ambigüedades que han podido ser tratadas satisfactoriamente para el caso de simetrías con un único eje pero no para simetrías más altas. Queda pendiente estudiar cómo trabajar con moléculas con más de un eje de simetría.

Los resultados obtenidos para el centrado de partículas pueden ser utilizados para acelerar algoritmos de microscopía electrónica ya que permiten reducir la búsqueda espacial que se debe realizar sobre las imágenes. Un análisis más profundo de los algoritmos actuales y de la precisión y tiempo de los modelos presentados sería necesaria para poder valorar si pueden resultar útiles en la práctica. En cualquier caso, se abre la puerta a seguir trabajando en este campo para tratar de mejorar los resultados obtenidos. Se podría probar otros tipos de arquitectura de redes neuronales para tratar de mejorar la precisión de las predicciones. También se podría plantear otro tipo de enfoques para resolver el problema, como modelos generativos más complejos capaces de reconstruir el mapa tridimensional a partir de las imágenes o un modelo universal, como se ha hecho para el caso del centrado de moléculas, que, a partir de un mapa a priori de la molécula, estime orientaciones sin necesidad de un entrenamiento previo que requiera etiquetar un subconjunto de las imágenes experimentales con otros métodos.

Referencias

- [1] Milne, J.L.S., Borgnia, M.J., Bartesaghi, A., Tran, E.E.H., Earl, L.A., Schauder, D.M., Lengyel, J., Pierson, J., Patwardhan, A. and Subramaniam, S. (2013), *Cryo-electron microscopy – a primer for the non-microscopist*. FEBS J, 280: 28-45. <https://doi.org/10.1111/febs.12078>
- [2] F. DiMaio, W. Chiu, (2016) *Chapter Ten - Tools for Model Building and Optimization into Near-Atomic Resolution Electron Cryo-Microscopy Density Maps*, Editor(s): R.A. Crowther, Methods in Enzymology, Academic Press, Volume 579, Pages 255-276, ISSN 0076-6879, ISBN 9780128053829, <https://doi.org/10.1016/bs.mie.2016.06.003>.
- [3] G. T. Herman, *Fundamentals of computerized tomography: image reconstruction from projections*, 2nd ed. (Springer, 2009)
- [4] Vilas Prieto, J. L. (2019). *Local quality assessment of cryoem reconstructions and its applications, From local resolution to local sharpening*. Facultad de Ciencias. Universidad Autónoma de Madrid.
- [5] Sjors H.W. Scheres, (2012) *A Bayesian View on Cryo-EM Structure Determination*, Journal of Molecular Biology, Volume 415, Issue 2, Pages 406-418, ISSN 0022-2836, <https://doi.org/10.1016/j.jmb.2011.11.010>.
- [6] Leandro Farias Estrozi, Jorge Navaza, (2008) *Fast projection matching for cryo-electron microscopy image reconstruction*, Journal of Structural Biology, Volume 162, Issue 2, Pages 324-334, ISSN 1047-8477, <https://doi.org/10.1016/j.jsb.2008.01.014>.
- [7] Punjani, A., Rubinstein, J., Fleet, D. et al. (2017) *cryoSPARC: algorithms for rapid unsupervised cryo-EM structure determination*. Nat Methods 14, 290–296. <https://doi.org/10.1038/nmeth.4169>
- [8] C.O.S. Sorzano, J. Vargas, J.M. de la Rosa-Trevín, A. Jiménez, D. Maluenda, R. Melero, M. Martínez, E. Ramírez-Aportela, P. Conesa, J.L. Vilas, R. Marabini, J.M. Carazo, (2018) *A new algorithm for high-resolution reconstruction of single particles by electron microscopy*, Journal of Structural Biology, Volume 204, Issue 2, Pages 329-337, ISSN 1047-8477, <https://doi.org/10.1016/j.jsb.2018.08.002>.
- [9] R.H. Wade. (1992) *A brief look at imaging and contrast transfer*, Ultramicroscopy, Volume 46, Issues 1–4, Pages 145-156, ISSN 0304-3991, [https://doi.org/10.1016/0304-3991\(92\)90011-8](https://doi.org/10.1016/0304-3991(92)90011-8).
- [10] Jürgen Schmidhuber. (2015) *Deep learning in neural networks: An overview*. Neural networks 61, Pages 85-117.
- [11] O’Shea, K., & Nash, R. (2015). *An introduction to convolutional neural networks*. arXiv preprint arXiv:1511.08458.
- [12] Banerjee, C., Mukherjee, T., & Pasiliao Jr, E. (2019, April). *An empirical study on generalizations of the ReLU activation function*. In *Proceedings of the 2019 ACM Southeast Conference* (pp. 164-167).
- [13] Scherer, Dominik; Müller, Andreas C.; Behnke, Sven (2010). *Evaluation of Pooling Operations in Convolutional Architectures for Object Recognition*. 20th International Conference Artificial Neural Networks (ICANN): 92-101.
- [14] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). *Dropout: a simple way to prevent neural networks from overfitting*. The journal of machine learning research, 15(1), 1929-1958.

- [15] Bjorck, N., Gomes, C. P., Selman, B., & Weinberger, K. Q. (2018). *Understanding batch normalization*. Advances in neural information processing systems, 31.
- [16] H. Kaiming, Z. Xiangyu, R. Shaoqing, S. Jian (2015). *Deep Residual Learning for Image Recognition*. arXiv:1512.03385.
- [17] Wu, Z., Shen, C., & Van Den Hengel, A. (2019). *Wider or deeper: Revisiting the resnet model for visual recognition*. Pattern Recognition, 90, 119-133.
- [18] Weisstein, E. W. (2009). *Euler angles*. <https://mathworld.wolfram.com/>.
- [19] Gothen, P., & Guedes de Oliveira, A. (2023). *On Euler's Rotation Theorem*. The College Mathematics Journal, 54(3), 171-175.
- [20] Wikipedia contributors. (2023, February 8). *Gimbal lock*. In Wikipedia, The Free Encyclopedia. Retrieved 21:00, September 5, 2023, from https://en.wikipedia.org/w/index.php?title=Gimbal_lock&oldid=1138224221
- [21] Vicci, L. (2001). *Quaternions and rotations in 3-space: The algebra and its geometric interpretation*. Chapel Hill, NC, USA, Tech. Rep.
- [22] Markley, Landis & Cheng, Yang & Crassidis, John & Oshman, Yaakov. (2007). *Averaging Quaternions*. Journal of Guidance, Control, and Dynamics. 30. 1193-1196. 10.2514/1.28949.
- [23] Y. Zhou, C. Barnes, J. Lu, J. Yang and H. Li, textitOn the Continuity of Rotation Representations in Neural Networks, 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 2019, pp. 5738-5746, doi: 10.1109/CVPR.2019.00589.
- [24] Butler, P. H. (2012). *Point group symmetry applications: methods and tables*. Springer Science & Business Media.
- [25] Giorgia Pitteri, Michaël Ramamonjisoa, Slobodan Ilic, Vincent Lepetit. (2019) *On Object Symmetries and 6D Pose Estimation from Images*. 3DV, Québec, Canada. fhal-02509435f
- [26] Crowther, R.A.; Henderson, R.; Smith, J.M. (January 1996). "MRC Image Processing Programs". Journal of Structural Biology. 116 (1): 9–16. doi:10.1006/jsbi.1996.0003. PMID 8742717
- [27] Hall, Sydney R.; Spadaccini, Nick (1994). "The STAR File: detailed specifications". J. Chem. Inf. Comput. Sci. 34 (3): 505–508. doi:10.1021/ci00019a005
- [28] Fernandez, J. J., Li, S., & Crowther, R. A. (2006). *CTF determination and correction in electron cryotomography*. Ultramicroscopy, 106(7), 587-596.
- [29] J.M. de la Rosa-Trevín, A. Quintana, L. del Cano, A. Zaldívar, I. Foche, J. Gutiérrez, J. Gómez-Blanco, J. Burguet-Castell, J. Cuenca-Alba, V. Abrishami, J. Vargas, J. Otón, G. Sharov, J.L. Vilas, J. Navas, P. Conesa, M. Kazemi, R. Marabini, C.O.S. Sorzano, J.M. Carazo, (2016) *Scipion: A software framework toward integration, reproducibility and validation in 3D electron microscopy*, Journal of Structural Biology, Volume 195, Issue 1, Pages 93-99, ISSN 1047-8477, <https://doi.org/10.1016/j.jsb.2016.04.010>.
- [30] <https://www.ebi.ac.uk/empiar/EMPIAR-10391/>
- [31] <https://www.ebi.ac.uk/empiar/EMPIAR-10482/>
- [32] <https://www.ebi.ac.uk/empiar/EMPIAR-11233/>
- [33] <https://www.ebi.ac.uk/empiar/EMPIAR-10421/>

- [34] Wikipedia contributors. (2023, January 23). Arabinosyltransferase. In Wikipedia, The Free Encyclopedia. Retrieved 17:27, August 23, 2023, from <https://en.wikipedia.org/w/index.php?title=Arabinosyltransferase&oldid=1135223302>
- [35] Wikipedia contributors. (2023, August 23). Chromatin remodeling. In Wikipedia, The Free Encyclopedia. Retrieved 17:40, August 23, 2023, from https://en.wikipedia.org/w/index.php?title=Chromatin_remodeling&oldid=1171861903.
- [36] Wikipedia contributors. (2023, February 28). TRPM8. In Wikipedia, The Free Encyclopedia. Retrieved 07:03, August 24, 2023, from <https://en.wikipedia.org/w/index.php?title=TRPM8&oldid=1142025267>
- [37] Wikipedia contributors. (2023, July 31). Ferritin. In Wikipedia, The Free Encyclopedia. Retrieved 08:28, August 24, 2023, from <https://en.wikipedia.org/w/index.php?title=Ferritin&oldid=1168055718>
- [38] <https://www.ebi.ac.uk/empiar/EMPIAR-10330/>
- [39] Wikipedia contributors. (2023, August 13). Plasmodium. In Wikipedia, The Free Encyclopedia. Retrieved 09:57, August 26, 2023, from <https://en.wikipedia.org/w/index.php?title=Plasmodium&oldid=1170069959>
- [40] Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.
- [41] <https://docs.scipy.org/doc/scipy/reference/generated/scipy.ndimage.rotate.html>
- [42] Wikipedia contributors. (2023, August 17). *Softmax function*. In Wikipedia, The Free Encyclopedia. Retrieved 21:17, September 5, 2023, from https://en.wikipedia.org/w/index.php?title=Softmax_function&oldid=1170880967
- [43] Koidl, K. (2013). *Loss functions in classification tasks*. School of Computer Science and Statistic Trinity College, Dublin, 1-5.