Universidad Nacional de
Educación a Distancia

Departamento de Inteligencia Artificial

Representación de imágenes geométricas con grafos y diagnóstico automático de la Figura Compleja de Rey

Oliver Hijano Cubelos

Tutor: Mariano Rincón Zamorano

Trabajo Fin de Máster

Máster Universitario en Investigación en Inteligencia Artificial

FECHA DE PRESENTACIÓN

Contenidos

1 Introducción

La Inteligencia Artificial ha revolucionado la gran mayoría de los sectores de desarrollo y producción en las últimas décadas. Sus aplicaciones son tan diversas y tan útiles que hoy en día es difícil encontrar un área de actividad que no esté impactada directa o indirectamente por la IA. En cuanto a las potenciales aplicaciones prácticas de métodos de Inteligencia Artificial en el ámbito de la salud, dos de las áreas más exploradas recientemente son el tratamiento del lenguage natural y la visión artificial. Por ejemplo, el tratamiento del lenguaje natural ha sido aplicado para predecir estados depresivos en ciertos sujetos en función de sus comentarios en Twitter o para analizar los informes médicos escritos por los doctores de forma automática [1]. La visión artificial también tiene infinidad de aplicaciones debido a que la mayoría de las decisiones médicas se basan en algún tipo de imagen como rayos X, escáneres de resonancia magnética o fotos de tejidos tomadas al microscopio. Algunos ejemplos prácticos pueden ser el análisis automático de tejidos tras una biopsia o la predicción de la evolución de cánceres de mama a partir de imágenes de resonancias magnéticas [2, 3].

En este trabajo voy a describir una potencial aplicación de visión artificial para la prevención y el diagnóstico de enfermedades neurodegenerativas. Existen numerosas pruebas y tests diseñados para detectar y evaluar este tipo de enfermedades. Un tipo de pruebas consiste en evaluar dibujos geométricos hechos a mano, generalmente copiando un dibujo original o siguiendo una serie de patrones, como por ejemplo el *Mini-Examen Cognoscitivo* [4], el *Test Barcelona* [5] o el *Test de Figura Compleja de Rey* [6–10], representado en la Fig. 1. Cada prueba tiene un sistema de evaluación en particular para que los expertos puedan analizar cuantitativamente y cualitativamente la calidad de la reproducción en función de las distorsiones del dibujo. Esta evaluación manual permite analizar, entre otros, la memoria y la capacidad visoconstructiva del sujeto. En general, la evaluación de este tipo de test es costosa. En concreto la Figura Compleja de Rey (FCR) se divide en 18 partes, tal y como se muestra en la Fig. 2. Cada una de las 18 partes se evalúa de forma independiente entre 0 y 2 puntos. 2 puntos si la ubicación del elemento y los trazos son correctos, 1 punto si al menos la ubicación o el trazo son correctos, y 0 puntos si el elemento no existe en el dibujo o su localización y trazos son incorrectos. En algunos casos se aplica tan sólo medio punto si la parte es reconocible pero está deformada e incorrectamente situada. De esta manera, la evaluación final es la suma de puntos de cada elemento y se mide en una escala entre 0 y 36 puntos [11]. El coste de estas evaluaciones podría verse reducido si la evaluación y el análisis de los dibujos pudiera hacerse parcial o totalmente de manera automática con ayuda de algoritmos de visión artificial. Además, puesto que las puntuaciones de cada parte son relativamente subjetivas, en muchos casos existen diferencias de hasta tres puntos entre dos

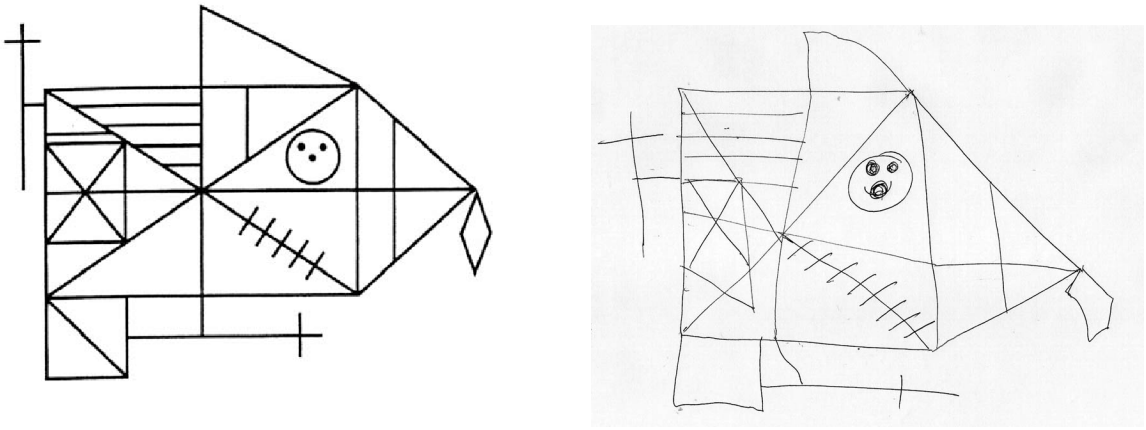


Fig. 1: Figura Compleja de Rey original (izquierda) y copia hecha a mano (derecha).

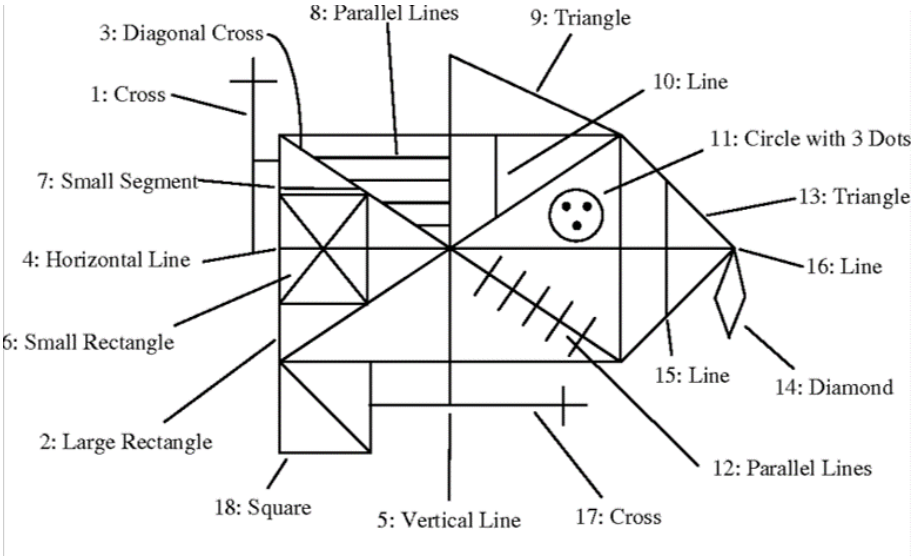


Fig. 2: La FCR se divide en 18 elementos. Cada elemento se evalúa de forma independiente entre 0 y 2 puntos. La puntuación total se obtiene como la suma de puntos de cada una de sus partes.

evaluadores humanos distintos [12].

La finalidad de este Trabajo de Fin de Máster es doble:

1. Por un lado quiero estudiar las estructuras de datos en forma de grafos para almacenar la información *relevante* de una imagen geométrica. Es decir, desarrollar un algoritmo capaz de representar cualquier dibujo de la FCR con un grafo. Las imágenes y la mayoría de las estructuras de datos más usadas (texto, video, audio, tensores, etc) se

presentan de forma natural con un orden de lectura predeterminado. Los grafos no presentan esta propiedad, ya que un grafo viene definido por su conjunto de nodos y arcos, pero a priori no existe un orden establecido en cuanto al orden de lectura de los nodos. La motivación de este trabajo es la de estudiar la evaluación automática de la FCR utilizando este tipo de estructuras de datos considerablemente diferente a las estructuras de imágenes generalmente utilizadas. Estos gráfos serán estudiados en función de su complejidad. Así mismo, cada nodo y cada arco serán estructuras de datos independientes que podrán almacenar información como la posición de las esquinas o la longitud, posición y ángulo de los trazos. Estas estructuras de datos tendrán que ser minuciosamente estudiadas para encontrar el equilibrio entre no perder demasiada información relevante del dibujo al mismo tiempo que se elimina la máxima cantidad de información redundante o ruido. Pese a que el algoritmo ha sido desarrollado para evaluar la FCR, todas sus partes son lo suficientemente robustas como para que pueda ser aplicado de forma genérica a otras imágenes geométricas, de manera que puede ser reciclado para ser utilizado en la evaluación de otros tests.

2. El segundo objetivo es el de utilizar técnicas de Inteligencia Artificial para poder evaluar una copia de la FCR en forma de grafo y predecir el estado cognitivo del paciente o la puntuación del dibujo. Puesto que vamos a disponer de un conjunto de imágenes anotadas, se trata de un problema de aprendizaje supervisado. En definitiva, se trata de investigar la capacidad de los sistemas inteligentes para imitar la evaluación manual de los expertos.

El producto final se trata de una herramienta con todos los algoritmos integrados para poder evaluar automáticamente el test de la Figura Compleja de Rey. De entrada se necesita un dibujo escaneado, y como salida la herramienta estima una puntuación entre 0 y 36 al dibujo y clasifica al paciente como sano o paciente con deterioro congnitivo leve.

2 Trabajo relacionado

Convencionalmente las imágenes digitales en dos dimensiones se representan con un conjunto de pixeles. Hoy en día la gran mayoría de los tratamientos y análisis automáticos de este tipo de imágenes se hacen con Redes Neuronales Convolucionales [13] o estructuras de redes neuronales similares capaces de reconocer patrones *leyendo* el mapa de pixeles de una imagen. La primera parte de mi TFM consiste en investigar representaciones de imágenes alternativas, especialmente en forma de grafos [14]. Muchos de los test anteriormente mencionados se basan en imágenes compuestas por elementos geométricos simples. Por ejemplo, la FCR se compone

solamente de líneas rectas, tres puntos y un círculo. Esta estructura puede transformarse en nodos y arcos utilizando diversas heurísticas. Existen varios métodos para transformar imágenes en grafos, por ejemplo el conocido *Nearest Spanning Chain* [15] o el utilizado por Felzenszwalb *et al.* [16], que considera que cada pixel es un nodo en el grafo, y que hay un arco ponderado entre cada par de píxeles adyacentes si cumplen ciertas condiciones de similaritud. En el caso que nos ocupa, la investigación consistirá en representar elementos geométricamente simples con grafos. Hay muchas heurísticas posibles y una parte de mi trabajo será estudiar sus propiedades y su utilidad para el problema que nos ocupa.

Los grafos presentan varias ventajas frente a las imágenes *clásicas*. Una de ellas es la posibilidad de reducir las dimensiones del espacio multidimensional representado por las imágenes [17]. Esto se consigue si el número de nodos y arcos (es decir, el tamaño del grafo) es menor al tamaño de la imagen original. Por supuesto, se debe prestar atención a la pérdida de información relevante, por lo que la heurística para hacer la transformación deberá ser cuidadosamente estudiada. Otra ventaja es la posibilidad de estudiar sub-grafos dentro del grafo como si fueran grafos únicos [18]. Es decir, estudiar partes de la imagen de manera aislada. Algo que no se puede hacer (o resulta muy dificil) utilizando técnicas de *Deep Learning* con imágenes. Esta ventaja es particularmente interesante en el caso de la FCR, puesto que esta imagen ha sido diseñada como un conjunto de *piezas sueltas* que son evaluadas de manera independiente por los expertos. El trabajo de Roberto Cerrillo [19] recoge resultados preliminares utilizando técnicas de *graph matching* aplicadas a dibujos de test neuropsicológicos.

La segunda parte de mi TFM consiste en utilizar representaciones de las FCR junto a sus evaluaciones manuales para poder puntuar numéricamente los dibujos y para clasificar automáticamente a los pacientes basándonos en sus dibujos. Se trata de un problema de aprendizaje supervisado clásico [20]. Una etapa importante en el proceso de clasificación es partir de una base de datos *limpia* con las anotaciones pertinentes, por lo que será importante conseguir la robustez mencionada en la sección anterior cuando se construyan los grafos a partir de imágenes.

Eladio Estella [21] realizó un estudio sobre la clasificación automática de pacientes basándose en dibujos a mano de la FCR. Para ello utilizó redes convolucionales siamesas capaces de extraer características y patrones de las imágenes, obteniendo una precisión en torno al 80%. Estella utilizó un *dataset* con 887 imágenes de la FCR, de las cuales 477 estaban anotadas [22]. Los test de la FCR se llevaron a cabo en centros culturales con sujetos inicialmente sanos a los cuales se les hizo un seguimiento anual durante varios años para estudiar sus capacidades cognitivas. Cada sujeto se vió sometido al test entre 1 y 5 veces a lo largo del seguimiento.

Además de los tests de la FCR, a cada sujeto se le realizaron otros test neuropsicológicos para realizar una evaluación global del estado cognitivo. Cada test está orientado a la evaluación de distintas funciones cognitivas. Tras cada evaluación el paciente fue diagnosticado con un *Deterioro Cognitivo Leve* (DCL) si su puntuación en alguna de las pruebas realizadas estaba por debajo de 1.5 desviaciones típicas por debajo de la media de su grupo según edad y formación educativa, o se consideró sano en caso contrario.

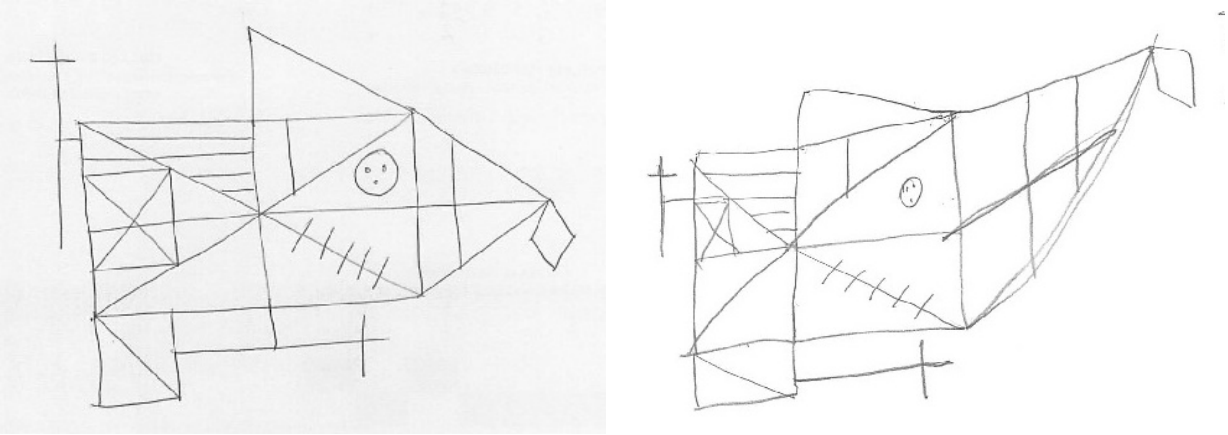
Cuando los datos disponibles son escasos o cuando hay alta variabilidad, se pueden utilizar técnicas de *Transfer Learning* (TL) para preentrenar un modelo. Por ejemplo, Estella utilizó imágenes del juego en línea *QuickDraw*. En general hay dos tipos de TL. El primero consiste en utilizar una base de datos con imágenes *similares* a las imágenes del *dataset* que se quiere utilizar para preentrenar un modelo. Después este modelo se adapta al problema en cuestión. El segundo tipo es simplemente partir de un modelo pre-entrenado para reentrenarlo con otros datos. Por ejemplo, el modelo CamemBERT [23] es un modelo utilizado para el tratamiento de texto que ha sido entrenado con un corpus de textos en francés y que puede ser reutilizado para otras aplicaciones. En nuestro caso podremos utilizar datos de otros test neuropsicológicos o incluso del mismo juego *QuickDraw* para crear *datasets* con los grafos pertinentes y poder preentrenar los modelos antes de comenzar el aprendizaje supervisado con los datos de la FCR.

Para construir las redes neuronales basadas en grafos me voy a apoyar en los estudios de Kipf *et al.* en 2017 [24] y de Zhang *et al.* en 2018 [25]. En ellos se describen nuevas estructuras de redes convolucionales que operan directamente en grafos y cuyas representaciones de las capas internas son capaces de codificar tanto propiedades locales de los grafos como propiedades de los nodos y arcos. Estas estructuras, además, aceptan grafos de estructura arbitraria, lo cual es realmente útil para grafos de la FCR que no tienen siempre el mismo número de nodos y arcos y cuyo ordenamiento tampoco está predefinido.

3 Materiales y métodos

3.1 Conjunto de datos utilizado

Durante el estudio presentado en [9] se realizaron diferentes sesiones en las que se recogieron, entro otros, cientos de dibujos de la FCR de varios sujetos. Para realizar este estudio se seleccionaron un grupo de pacientes sanos a los que se les hizo una serie de pruebas con un seguimiento anual. Estas pruebas tenían como objetivo hacer una evaluación de las capacidades cognitivas de los sujetos y de estudiar su evolución con el tiempo. Cada paciente



(a) Sujeto: 430 (evaluación: 3)
Edad: 71
Estudios: 22
Puntuación FCR: 36
Diagnóstico: DCLna

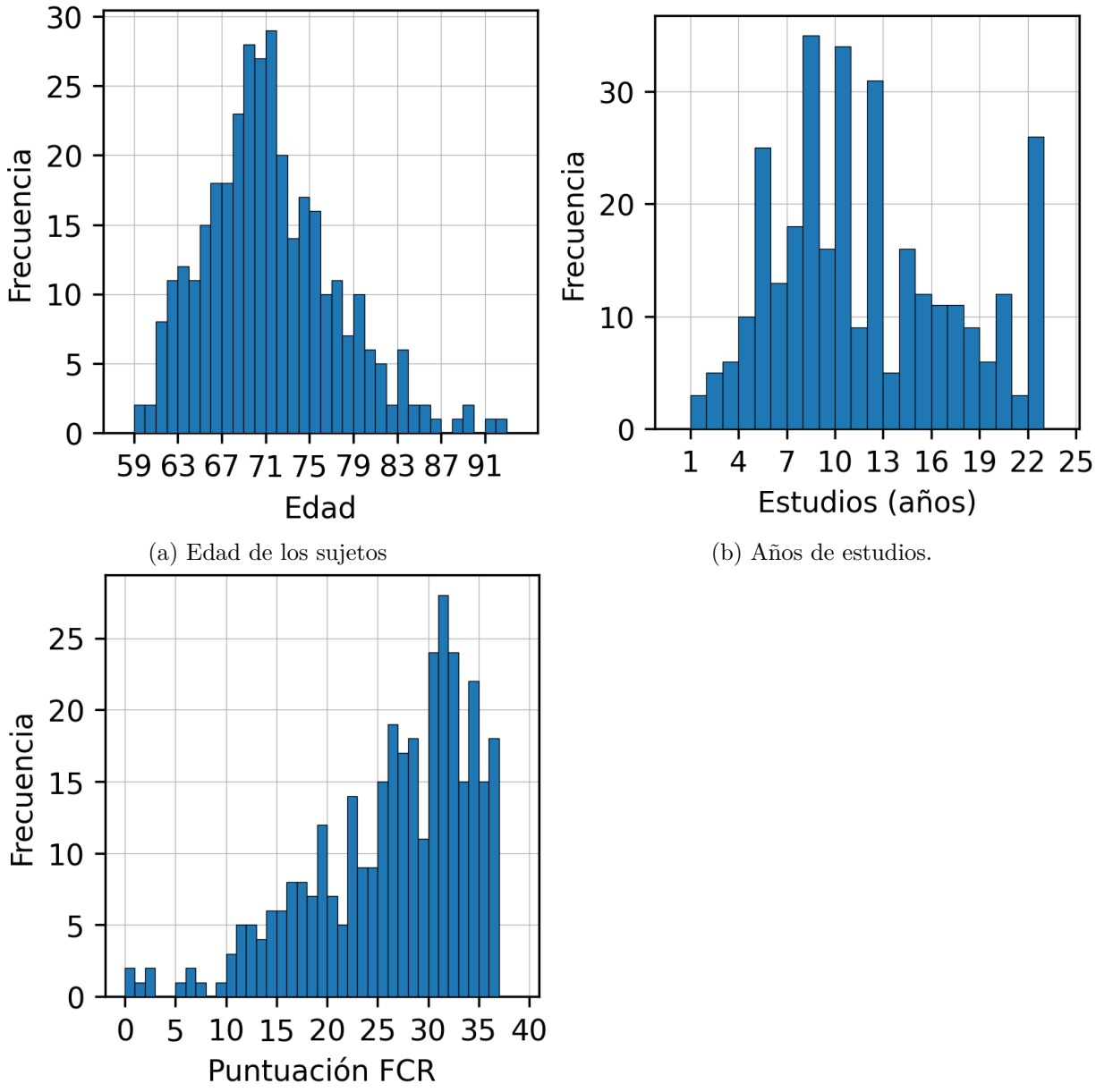
(b) Sujeto: 10 (evaluación: 3)
Edad: 71
Estudios: 4
Puntuación FCR: 25
Diagnóstico: DCLna

Fig. 3: Dos ejemplos de dibujos anotados.

realizó entre una y cinco sesiones de pruebas.

El conjunto de datos que he utilizado se compone de una fracción de los dibujos escaneados durante dicho estudio. Este conjunto de datos se compone de un total de 887 imágenes con dibujos de la FCR y un archivo de texto con 936 anotaciones de los sujetos que han realizado los dibujos. Estas anotaciones incluyen la edad, el nivel de estudios (medido en años de estudios superiores), la puntuación del dibujo de la FCR en cuestión y la clasificación del sujeto como sano o con deterioro cognitivo leve (DCL). Como ya mencionamos en la introducción, la puntuación del dibujo se mide en una escala numérica entre 0 y 36 puntos. Un sujeto es considerado sano por defecto y se considera que tiene DCL si en dos o más pruebas obtiene una puntuación por debajo de 1.5 desviaciones típicas debajo de la media con respecto a otros sujetos con edad y nivel de estudios similares. El deterioro cognitivo leve puede ser de tres tipos:

- DCLa (amnésico): Cuando el sujeto obtiene una puntuación por debajo de 1.5 desviaciones típicas debajo de la media en al menos dos pruebas del Test de Aprendizaje Verbal España-Complutense (TAVEC) [26].
- DCLna (no amnésico): Cuando las dos pruebas *falladas* no forman parte del TAVEC.
- DCLm (multidominio): Cuando una prueba *fallada* forma parte del TAVEC y la otra no.



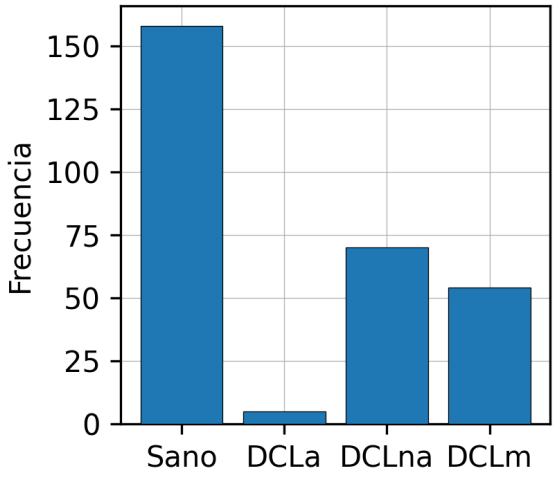
(a) Edad de los sujetos

(b) Años de estudios.

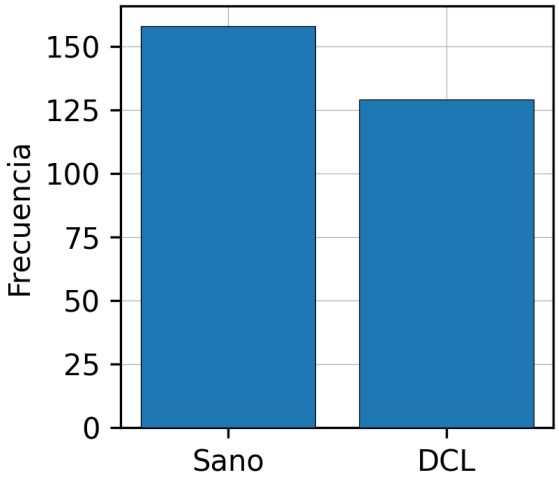
(c) Puntuación 0-36 de los dibujos FCR.

Fig. 4: Distribución de los datos anotados disponibles.

Puesto que se trata de un estudio longitudinal, en muchos casos existen múltiples datos para un mismo paciente. La disparidad entre el número de imágenes y el número de anotaciones se debe a que, por un lado, existen varias imágenes para una única evaluación de un sujeto y, por otro lado, a que existen anotaciones de imágenes que no están disponibles en el conjunto de datos. Existen múltiples razones por las que puedan existir varios dibujos de un mismo

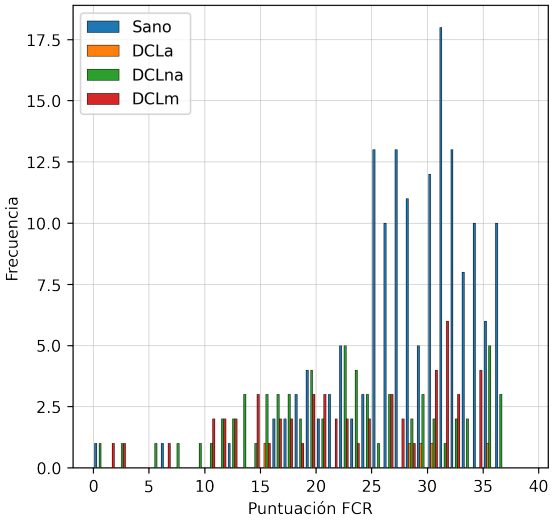


(a) Frecuencia de cada posible diagnóstico.

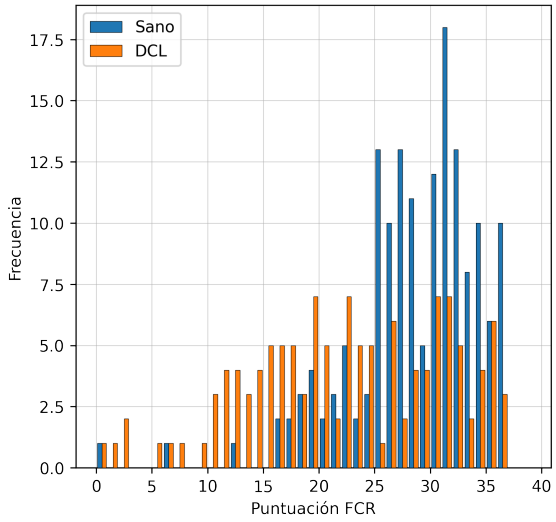


(b) Agrupación de los tres tipos de DCL.

Fig. 5: Distribución de los diagnósticos de los sujetos.



(a) Cuatro diagnósticos posibles.



(b) Agrupación de los tres tipos de DCL.

Fig. 6: Distribución de la puntuación de los dibujos en función del diagnóstico.

sujeto en una instancia dada, como puede ser que el sujeto haya decidido repetir la prueba o que haya dibujos incompletos.

En total hay 347 imágenes anotadas, 540 imágenes sin anotaciones y 589 anotaciones incompletas o sin imágenes disponibles. La Fig. 3 muestra dos de los dibujos disponibles con sus anotaciones correspondientes. Para la realización de este trabajo hemos utilizado únicamente el conjunto de 347 imágenes con anotaciones disponibles. Cada imagen disponible

muestra la FCR debidamente recortada y en posición horizontal.

La Fig. 4 muestra que la gran mayoría de los sujetos tienen más de 60 años, puesto que el deterioro cognitivo leve afecta principalmente a las personas más ancianas y por lo tanto son el sector de la población más relevante para realizar el estudio. Un elemento muy importante a destacar es la distribución de la puntuación de los dibujos; la mayoría de los dibujos tienen buena o muy buena puntuación. Muy pocos dibujos tienen menos de 20 puntos. Esto puede resultar un problema a la hora de entrenar nuestros modelos de inteligencia artificial, puesto que hay un sesgo claro hacia dibujos de mayor puntuación. La Fig. 5 muestra el reparto en cuanto al diagnóstico de los sujetos. La mayoría de ellos están sanos y hay sólo 5 casos de sujetos diagnosticados con DCLa (amnésico). Cuando se reagrupan los tres tipos de DCL el reparto es más equitativo. Por esta razón no voy a entrenar un modelo de IA capaz de distinguir entre los tipos de DCL; no existe una cantidad de datos suficientemente grande como para entrenar y evaluar un modelo con precisión. Sin embargo sí vamos a poder hacer una clasificación más genérica entre sanos y no sanos. La Fig. 6 muestra la distribución de las puntuaciones de los sujetos en función de su diagnóstico. Como se puede apreciar la mayoría de los sujetos sanos tienen puntuaciones muy elevadas con sólo tres casos en los que el dibujo tiene menos de 15 puntos y sólo 12 casos con menos de 20 puntos. El reparto de la puntuación de pacientes con DCL está más repartido, con puntuaciones que varían desde los 10 puntos hasta los 36 puntos. Es por ello que hay un alto solapamiento en los diagnósticos de pacientes con dibujos con alta puntuación.

3.2 Modelo propuesto

Para realizar este trabajo he utilizado una red neuronal de grafos (GNN en adelante, de *Graph Neural Network*) como las descritas en el último párrafo de la sección 2[1]. La arquitectura de esta GNN acepta como entrada grafos de estructura arbitraria y permite, entre otros, realizar predicciones sobre el grafo en su totalidad y no solo de forma individual en sus nodos. Estas propiedades se adaptan perfectamente a nuestro objetivo ya que cada dibujo de la FCR es transformado en un grafo con un número de nodos y arcos variable y estamos interesados en puntuar o clasificar el grafo en su conjunto, no evaluar cada nodo por separado. La arquitectura de la GNN tiene las siguientes características principales:

- La arquitectura se puede dividir en dos partes principales. La primera es una arquitectura con capas convolucionales (*Graph Convolutional*) aplicadas al grafo capaces de extraer propiedades de los nodos. Estas capas convolucionales son muy similares a las clásicas

[1] Para ello he utilizado el paquete de Python stellargraph.

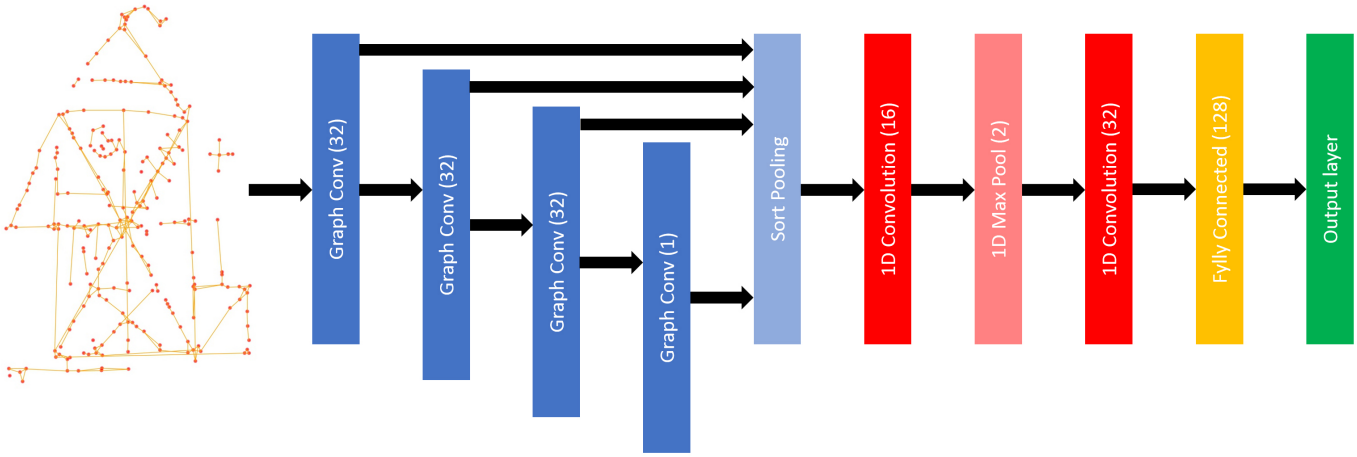


Fig. 7: Estructura de la GNN utilizada.

capas convolucionales aplicadas en el tratamiento de imágenes. La segunda parte es una estructura de red neuronal clásica con varias capas convolucionales y/o densas cuya entrada es un tensor numérico tradicional. Estas dos partes se unen gracias a la capa que se describe a continuación.

- Tal y como se muestra en la figura Fig. 7, se introduce una capa llamada *Sort Pooling* cuya función es la de ordenar los nodos del grafo bajo una heurística de forma que el grafo pueda ser leído como una secuencia de nodos. Se trata de un puente entre las capas convolucionales del grafo *Graph Convolutional* y las capas tradicionales de aprendizaje profundo. La ordenación de los nodos se realiza atendiendo a las funciones estructurales que los nodos tienen dentro del grafo. Esta ordenación es por lo tanto robusta ya que los nodos de dos grafos distintos tendrán la misma posición relativa en la ordenación si tienen funciones estructurales similares en sus respectivos grafos.

- La primera parte de la arquitectura está compuesta por una o varias capas convolucionales de grafos. Se asume que las propiedades de cada nodo vienen representadas por un tensor numérico. Estas capas convolucionales agregan la información de los nodos en vecindarios locales para extraer información local en la estructura del grafo. Es decir, tienen una función similar a las capas convolucionales aplicadas a imágenes.

La Fig. 7 muestra la estructura del modelo concreto utilizado. La entrada a la red neuronal es un grafo representando la FCR (las propiedades de estos grafos será detallada en la sección 4.1.3). Este grafo se puede describir con dos matrices; una para representar las conexiones entre los nodos y otra para describir las propiedades (numéricas) de cada nodo. Las primeras cuatro capas son *Graph Convolutional*; las tres primeras con 32 unidades cada una

y la última con 1 unidad. Todas ellas con la función de activación tangente hiperbólica. La salida de estas cuatro capas se agrupa en una sola capa, que alimenta una capa convolucional de una dimensión con 16 filtros y función de activación ReLu. A esta capa le sigue una segunda capa convolucional ReLu con 32 filtros, y finalmente una capa completamente conectada con 128 unidades y función de activación sigmoide.

La última capa dependerá del entrenamiento que queramos realizar. En la sección 4.2 explicaremos cómo podemos entrenar la red para predecir la puntuación numérica del grafo, en cuyo caso la última capa será una capa conectada con una sola unidad de activación lineal, ya que se trata de un problema de regresión. Pero también podemos entrenar la GNN para predecir la clase del sujeto, en cuyo caso la última capa será una capa conectada con función de activación *softmax* y con un número de unidades igual al número de clases totales.

3.3 Objetivos y métricas

3.3.1 Objetivos de evaluación de la FCR

Tal y como habíamos descrito en la sección 3.1, para cada imagen de la FCR disponemos tanto de la puntuación numérica (entre 0 y 36) del dibujo como del diagnóstico cognitivo del sujeto que lo ha hecho. Con estos datos podemos definir dos objetivos principales.

Objetivo 1: problema de regresión
Este primer objetivo consiste en utilizar únicamente la puntuación numérica real de cada grafo para resolver un problema de regresión. Se trata por lo tanto de entrenar una red neuronal capaz de precedir la puntuación de un grafo de la FCR dado.

Objetivo 2: problema de clasificación
Para este segundo objetivo vamos a utilizar únicamente el diagnóstico del paciente de cada grafo para resolver un problema de clasificación. La red neuronal deberá diagnosticar al sujeto como *sano* o *DLC* (agrupando los tres tipos de DLC) en función del grafo resultante de su dibujo.

Para cada uno de los dos objetivos descritos voy a dividir el conjunto de datos de forma que el 80% de los datos disponibles sean usados para el entrenamiento y el 20% restante para la evaluación de los modelos. Como veremos más adelante, la distribución de puntuaciones en el conjunto de entrenamiento es similar a la distribución de puntuaciones en el conjunto de test (ver Fig. 13). Así mismo la proporción de clases sano/DCL es de 126/149 (84.6%) en el conjunto de entrenamiento y de 32/37 (86.5%) en el conjunto de test. Si el número de datos

disponibles hubiese sido menor, otra alternativa habría sido utilizar la técnica de validación cruzada para maximizar el número de dibujos disponibles para el entrenamiento.

3.3.2 Métricas utilizadas

Problema de regresión

Para resolver el problema de regresión he entrenado una red neuronal cuya función de pérdida es el error cuadrático medio (*mse* de *Mean Squared Error*). Esta función de pérdida viene dada por

$$mse = \frac{1}{N}\sum_{i=1}^{N}(P_i^{true} - P_i^{neural})^2$$

Donde N es el número de dibujos disponibles, P_i^{true} es la puntuación real del i-ésimo dibujo, P_i^{neural} la puntuación predicha por la red neuronal.

Una vez la red neuronal ha sido entrenada no sólo tenemos acceso al mse total sino que también podemos calcular la puntuación de cada dibujo. Este problema de regresión lo podemos transformar en un problema de clasificación si introducimos el parámetro τ para determinar la tolerancia del error máximo que se puede producir a la hora de puntuar un dibujo. Si la diferencia absoluta entre la puntuación real y la puntuación otorgada por la red neuronal es igual o inferior a dicho τ podemos decir que el dibujo está bien puntuado. Si la diferencia es superior al límite de tolerancia τ, entonces el dibujo estará mal puntuado. Cuanto mayor sea τ, mayor margen de error podemos tolerar en la predicción de la puntuación y mayor será la precisión de la red neuronal. Conociendo el número de dibujos bien puntuados y el número de dibujos mal puntuados podemos calcular la precisión de puntuación de la red en función del parámetro τ. Idealmente la red neuronal debería tener buena precisión de puntuación para valores de τ relativamente bajos.

Problema de clasificación

Tal y como definimos en la sección 3.3.1, la clasificación de los dibujos es binaria; pertenecen a sujetos sanos o sujetos diagnosticados con DCL. Introduciendo un sistema de codificación *one-hot*, los dibujos de sujetos sanos los vamos a representar con el vector $(1, 0)$ y los dibujos de los sujetos con DCL los vamos a representar con el vector $(0, 1)$. Además, vamos a interpretar la salida de las dos[2] neuronas softmax de la última capa de la red neuronal (cuya suma es igual

[2]Tal y como describimos anteriormente en la sección 3.2, el número de neuronas en la última capa debería ser el número de clases existentes. No obstante, para el caso en particular de la clasificación binaria sano/DCL, también podríamos utilizar una única neurona de salida con función de activación sigmoide y clasificar la salida imponiendo un límite a la función de activación. En este trabajo continuamos con la estructura genérica de dos neuronas softmax para generalizar el problema y poder reciclar el trabajo si se realiza el estudio con más de dos clases.

a 1) como un vector de probabilidades sano-DCL. Si el primer elemento del vector es superior al segundo, la red neuronal clasificará el dibujo como dibujo de sujeto sano, y viceversa en caso contrario. La red neuronal la he entrenado utilizando la función de pérdida entropía cruzada categórica (*cce* de *Categorical Cross Entropy*). Esta función viene dada por

$$cce = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{2}(y_{i,j}^{true} log(y_{i,j}^{neural}))$$

Donde $(y_{i,1}^{true}, y_{i,2}^{true})$ es el vector que representa el diagnóstico del i-ésimo dibujo y $(y_{i,1}^{neural}, y_{i,2}^{neural})$ el vector que representa la salida de las dos neuronas softmax de la red neuronal.

Puesto que conocemos la clasificación real de los dibujos y que la red neuronal puede predecir la clase de cada dibujo gracias a la salida de las dos neuronas softmax, podemos construir la matriz de confusión y evaluar la precisión de la red neuronal en la tarea de clasificación.

4 Resultados

4.1 Transformación de imágenes en grafos

Para poder transformar cualquier dibujo de la FCR en un grafo he desarrollado una serie de tres algoritmos independientes. El primer algoritmo es capaz de transformar cualquier dibujo de la FCR en un dibujo en blanco y negro, con un tamaño predefinido y en el que se pueden distinguir claramente los trazos realizados por el sujeto. El segundo algoritmo coge como entrada una imagen producida por el primer algoritmo y es capaz de detectar los puntos y las líneas de interés; esquinas, puntos de corte, trazos del sujeto, etc. El tercer y último algoritmo es utilizado para construir un grafo a partir de los puntos y de las líneas de interés detectados anteriormente. Este grafo no se trata de una representación visual sino de una construcción *matemática* con algunas de las propiedades del dibujo realizado por el sujeto.

4.1.1 *Limpieza* de dibujos

En nuestro conjunto de datos todos los dibujos de la FCR están debidamente orientados y *recortados*, sin embargo el tamaño de los dibujos no es constante y existen notables diferencias. También existen diferencias en cuanto al contraste de los dibujos (debido, por ejemplo, a la presión que hace el sujeto sobre el papel a la hora de dibujar las líneas) y en cuanto a la calidad del escaneado. Para poder solucionar este problema he creado un algoritmo para

normalizar las imágenes. Tras aplicar el algoritmo a nuestro conjunto de datos, obtenemos un nuevo conjunto de datos en el que todas las imágenes tienen el mismo tamaño y los dibujos aparecen en blanco y negro *puro* (fondo completamente negro y trazos del dibujo completamente blancos). Las etapas del algoritmo son las siguientes:

1. En primer lugar abrimos cada imagen transformando los colores a una escala de grises. Esto evita, por ejemplo, diferencias entre dibujos realizados con bolígrafos de distintos colores.

2. A continuación escalamos las imágenes a un tamaño predefinido de 1200x800 píxeles utilizando una interpolación de vecinos más próximos. Este tamaño es suficientemente grande para poder capturar los elementos a escalas más pequeñas del dibujo, como los cortes de líneas o los trazos de menor tamaño.

3. La tercera fase de este algoritmo consiste en transformar la imagen en blanco o negro (en vez de escala de grises)[3]. Para ello, primero invierto la escala de grises para que el fondo de la imagen sea oscuro y los trazos claros. Puesto que hay muchos más píxeles oscuros que claros, considero que todos los píxeles que estén por debajo de la mediana deben ser considerados fondo de la imagen. Para el resto de píxeles (combinación de fondo y trazos reales del dibujo) calculo su media y desviación estándar y cambio del valor de los píxeles por encima y por debajo de una desviación estándar para evitar valores atípicos. A continuación hago una renormalización lineal de manera que el valor más oscuro de la imagen tenga un valor de 0 en la escala de grises y el valor más claro de 255. Finalmente, considero que todos los píxeles que tengan un valor igual o superior a 40 son blancos y los que tengan un valor inferior negros. El límite de 40 lo he tomado de resultados experimentales; he generado las imágenes en blanco y negro para varios valores del límite y he seleccionado aquel que producía las imágenes más nítidas.

4. A continuación elimino parte del ruido de la imagen. Debido a la calidad del escaneado o a la granularidad del papel, en muchas ocasiones se generan muchos puntos blancos esparcidos por toda la imagen. Para eliminarlos voy a considerar las coordenadas (x,y) de todos los píxeles blancos (candidatos a ser píxeles de trazo y no píxeles de fondo) y voy a utilizar un algoritmo de clustering para distinguir grupos de puntos suficientemente *grandes* como para ser considerados trazo y grupos de puntos aislados que pueden ser considerados ruido. Concretamente hago uso del algoritmo de clustering DBSCAN (*Density-Based Spatial Clustering of Applications with Noise*)[27]. Este algoritmo permite

[3]La idea de realizar la distinción binaria entre píxeles de trazo y píxeles de fondo ha sido aportada por mi tutor Mariano Rincón. Los detalles de algoritmo así como la elección de los mejores parámetros han sido diseñados por el autor.

encontrar clusters de geometría arbitraria que cumplan con la propiedad de que todos sus puntos tengan un número mínimo de vecinos predefinido. Esta propiedad es particularmente interesante por dos razones. La primera es que los trazos dilatados de los dibujos de la FCR son *objetos* con geometría arbitraria y por lo tanto fácilmente detectables como grupos de puntos que pueden agruparse en un único clúster. La segunda es que el ruido puede definirse como aquellos puntos que no tienen *demasiados* vecinos alrededor. Concretamente he limitado la distancia máxima de 1 píxel para buscar vecinos y un número mínimo de 25 vecinos para poder definir un clúster. Todos los puntos que no formen parte de un clúster son coloreados en negro y considerados como fondo de la imagen. De nuevo, el valor de 25 vecinos y los otros parámetros del algoritmo los he determinado tras realizar numerosas pruebas sobre el conjunto de dibujos disponibles. Estos parámetros dependen del tamaño de la imagen. Puesto que no dispongo de datos anotados o de imágenes con y sin ruido para poder hacer comparaciones cuantitativas, he tenido que probar diferentes configuraciones de los parámetros del algoritmo DBSCAN sobre todo el conjunto de datos disponible y seleccionar los mejores parámetros en función de una evaluación empírica de los resultados. La Fig. 8 muestra una imagen en la que se ha detectado una buena parte del ruido gracias a esta fase del algoritmo.

5. Por último utilizo el paquete cv2 para dilatar los trazos de la imagen para que sean más gruesos y facilitar las posteriores fases de la transformación de imágenes en grafos. Así mismo genero una erosión en el interior de los trazos para facilitar la detección de esquinas y puntos de interés.

La Fig. 9 muestra los procesos de transformación de una de las imágenes del conjunto de datos. Como se observa, el producto final de esta etapa de limpieza son imágenes con fondo negro y trazos blancos y ahuecados, razonablemente nítidas y con poco ruido, en las cuales se puede diferenciar relativamente bien los trazos de la FCR realizados por el sujeto. Además, el tamaño de estas imágenes no depende del tamaño o la calidad del dibujo original, y por lo tanto facilita su posterior evaluación.

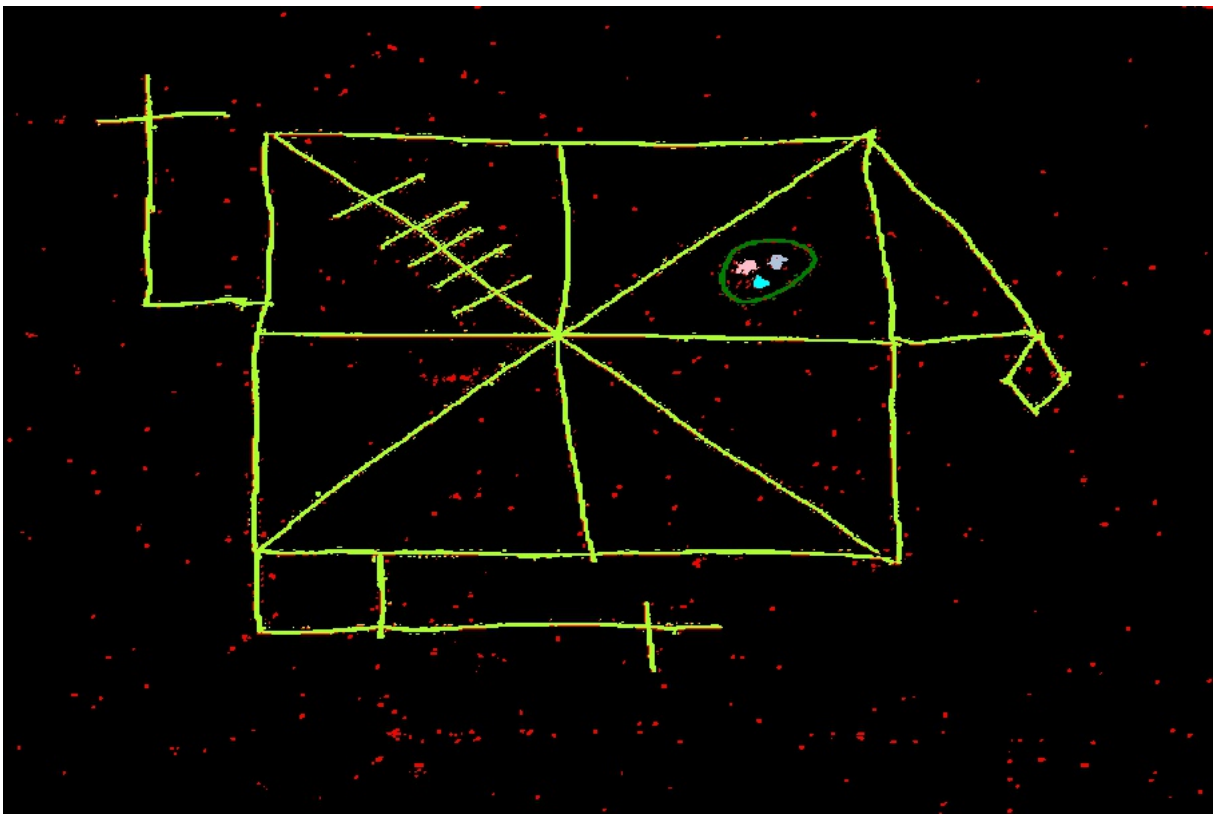
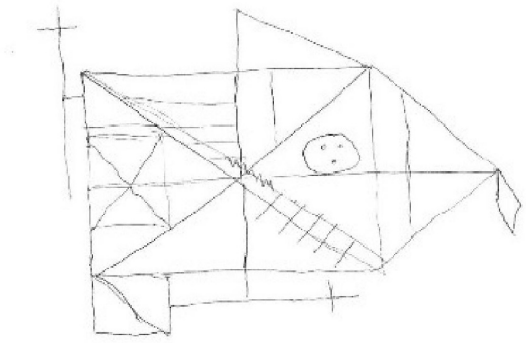
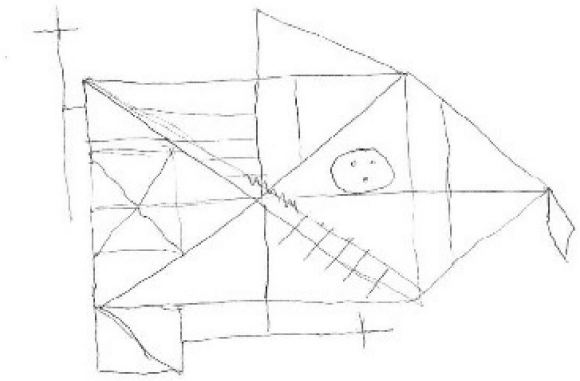


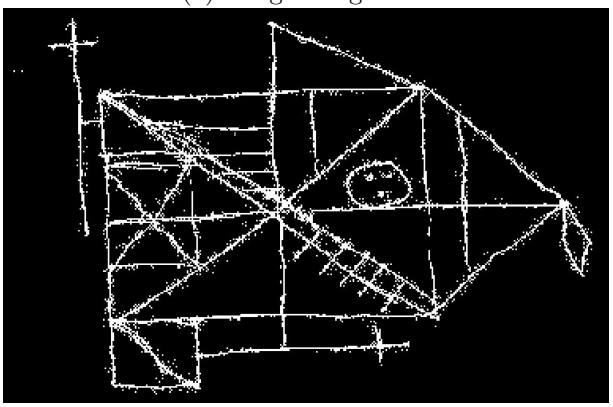
Fig. 8: Ejemplo de detección de ruido utilizando DBSCAN. Cada punto blanco de la imagen (puntos que a priori no son de fondo) ha sido coloreado en función del clúster al que pertenece. Los puntos rojos son puntos que no pertenecen a ningún clúster. En la imagen se observan un total de cinco clústers (estructura general de la FCR en verde lima, circulo en verde oscuro y tres puntos dentro del círculo en rosa, cyan y gris) que deben ser considerados como parte del dibujo del sujeto y un conjunto de puntos rojos que deben ser considerados como ruido.



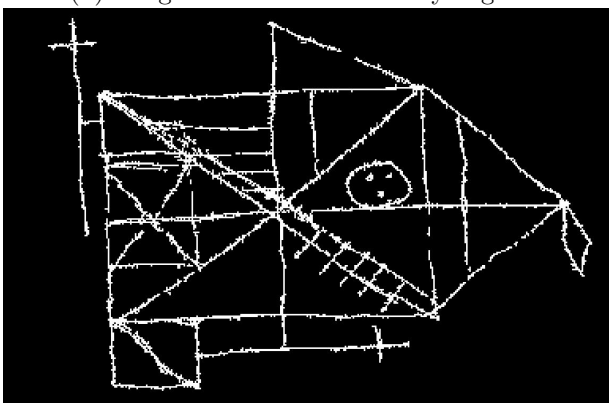
(a) Imagen original.



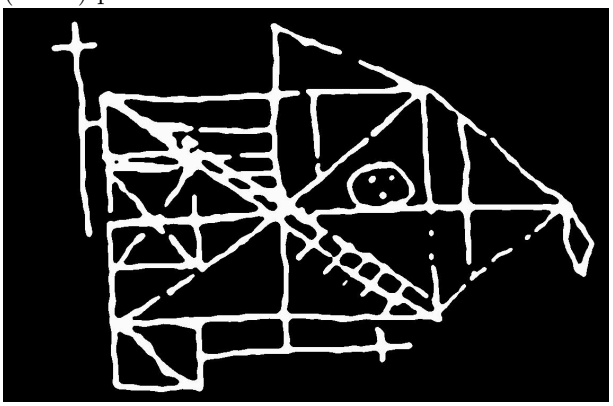
(b) Imagen escalada en blanco y negro.



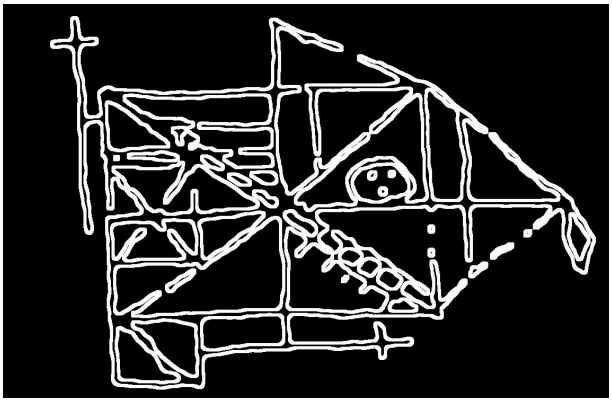
(c) Transformación a blanco (trazos) y negro (fondo) puro.



(d) Eliminación parcial del ruido.



(e) Dilatación de trazos.



(f) *Ahuecamiento* de trazos.

Fig. 9: *[Parte 1/2, ver Fig. 10]* Ejemplo de transformación de uno de los dibujos de la FCR y creación del grafo correspondiente.

4.1.2 Detección de puntos y líneas de interés

A continuación he desarrollado un algoritmo capaz de detectar puntos y líneas de interés a partir de las imágenes *limpias* producidas en la etapa anterior. Este algoritmo sigue las siguiente etapas:

1. En primer lugar el algoritmo detecta una exhaustiva colección de puntos candidatos a ser *puntos de interés*, es decir esquinas, cortes y finales de líneas. Para ello, he utilizado el algoritmo *Harris Corner Detector*[28] sobre el dibujo con los trazos gruesos ahuecados (como el de la Fig. 9f). El interés de utilizar la imagen con los trazos ahuecados es poder detectar puntos y esquinas que se encuentran dentro de los trazos blancos sin ahuecar. Cuando se realiza el mismo ejercicio sin ahuecar los trazos, la gran mayoría de puntos detectados se encuentran por fuera de los trazos dilatados.

2. El detector de Harris detecta muchos puntos de interés redundantes. Para simplificar el número de puntos he generado la siguiente rutina de dos etapas.

   (a) Dada una distancia d, utilizo de nuevo el algoritmo de clustering DBSCAN para agrupar esquinas que se encuentren en el mismo cluster y sustituyo todas las esquinas por el punto central del cluster. De esta manera todas las esquinas que se encuentran alrededor de la misma esquina real del dibujo de la FCR se agrupan en un sólo punto.

   (b) A continuación reposiciono cada punto para maximizar la cantidad de trazo blanco a su alrededor. Para cada punto de interés evalúo el *nivel de blanco* en un entorno de 6 píxeles de distancia utilizando la imagen con los trazos dilatados sin ahuecar, como la de la Fig. 9e. El nivel de blanco es calculado como la contribución de los píxeles blancos en el entorno considerado atenuados de forma gaussiana con la distancia[4]. Evalúo además el nivel de blanco de los ocho[5] píxeles vecinos. El punto original es reemplazado por el punto que mayor nivel de blanco tenga en su vecindad. Este proceso se repite hasta que ningún punto pueda ser mejorado (es decir, hasta que ninguno de los ocho vecinos tenga un nivel de blanco superior). Esto permite que los puntos queden bien centrados con respecto a las esquinas *reales* dibujadas por el sujeto.

[4] Por eso mismo no evalúa la contribución de blanco que aportan los píxeles a más de 6 unidades de distancia; la atenuación gaussiana hace que la contribución de píxeles más alejados sea prácticamente nula, y se acelera considerablemente el rendimiento del algoritmo.

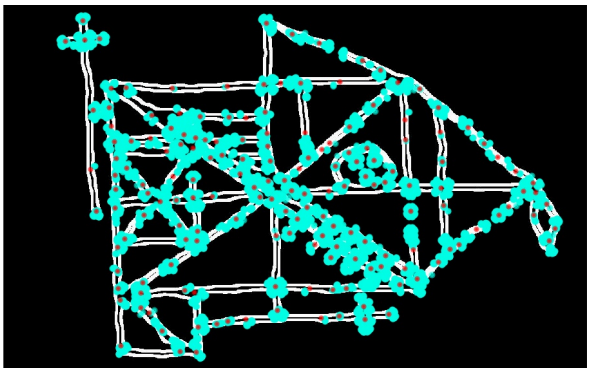
[5] Los cuatro píxeles arriba, abajo izquierda y derecha y los cuatro píxeles en las diagonales del punto considerado.

Esta rutina de dos etapas se repite hasta 20 veces, incrementando el valor de d (para realizar el clustering de DBSCAN) desde $d = 1$ hasta $d = 20$. Este incremento se hace para evitar agrupar esquinas que están lo suficientemente cerca pero que representan vértices del dibujo relativamente lejos. Este problema se evita si en primer lugar estas esquinas se agrupan y desplazan a escalas inferiores. El resultado final es un reducido número de puntos de interés relativamente bien posicionados dentro de los trazos del dibujo.

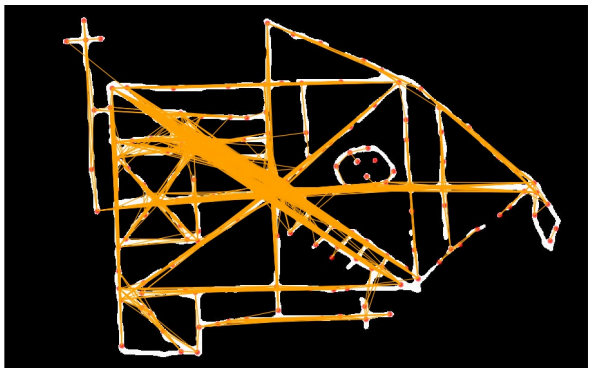
3. A continuación el algoritmo genera de forma exhaustiva todas las líneas candidatas a representar un trazo en el dibujo original. Para ello se evalúan todas las posibles parejas de puntos de interés generados en la etapa anterior. Dada una pareja de puntos, se traza una línea recta entre ellos y se evalúan todos los píxeles que *tocan* a dicha línea. De entre todos los píxeles que *pertenecen* a la línea, se calcula el número de píxeles blancos (parte del dibujo) y el número de píxeles negros (parte del fondo) utilizando la imagen con los bordes dilatados (como en Fig. 9e). Si el número de píxeles blancos es igual o superior al 70% del total, se considera que sí existe una línea entre dicha pareja de puntos. En caso contrario se considera que esos dos puntos no están conectados por un trazo del sujeto. El límite de 70% lo he impuesto tras evaluar empíricamente el rendimiento del algoritmo para varios valores de este límite. Cuando el límite es muy alto, entonces muchas líneas *reales* no son detectadas porque los trazos del sujeto a veces son un poco curvos o existe ruido de fondo y se pierde información sobre el trabajo realizado por el sujeto. Al mismo tiempo, cuando el límite es bajo en muchos casos se generan líneas *fantasma* que unen puntos que no están realmente unidos en el dibujo original.

4. En la mayoría de los dibujos una buena fracción de las líneas generadas son redundantes; están muy cerca entre sí y son prácticamente paralelas. Para eliminar las líneas redundantes he añadido una etapa al algoritmo para simplificar las líneas generadas. Consideremos una línea con los extremos A y B. Si existe un punto de interés C (de los generados en las etapas anteriores) que esté a menos de 20 píxeles de distancia de la línea y aque además cumpla que los ángulos $\widehat{BAC}$ y $\widehat{ABC}$ sean iguales o inferiores a 30^{o}, entonces esa línea se sustituye por las dos líneas (más cortas) AC y CB. Este proceso se realiza en bucle hasta que ninguna línea pueda ser simplificada, evitando siempre tener líneas repetidas. De nuevo, los valores de 20 píxeles y 30^{o} han sido definidos tras realizar numerosas pruebas sobre los dibujos disponibles.

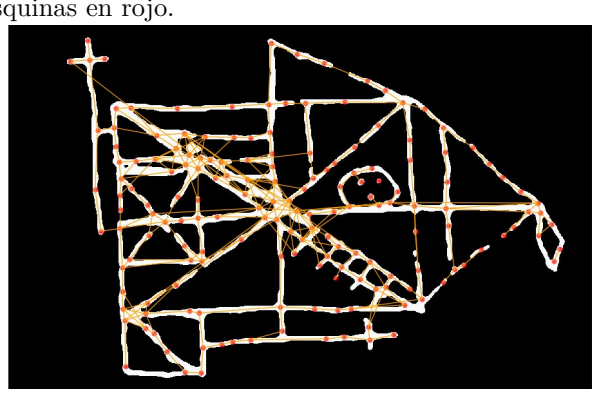
Nótese que tanto la imagen con los trazos dilatados (Fig. 9e) como la imagen con los trazos ahuecados (Fig. 9f) son necesarias para calcular los puntos y las líneas de interés. Sin la imagen con los trazos ahuecados el detector de esquinas de Harris detecta menos puntos



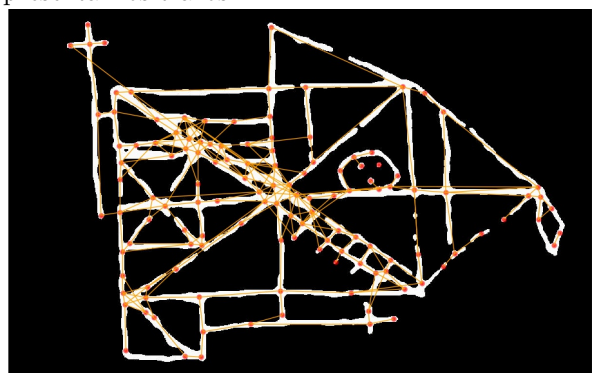
(a) Detección de esquinas en cyan y clustering de esquinas en rojo.



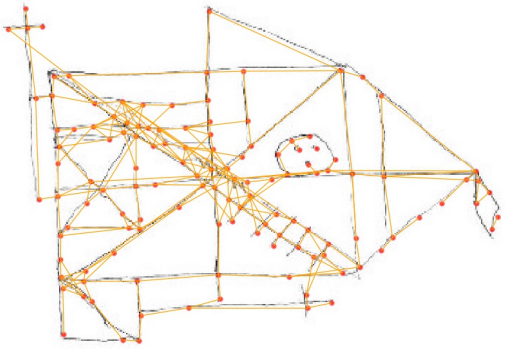
(b) Creación exhaustiva de líneas candidatas para representar los trazos.



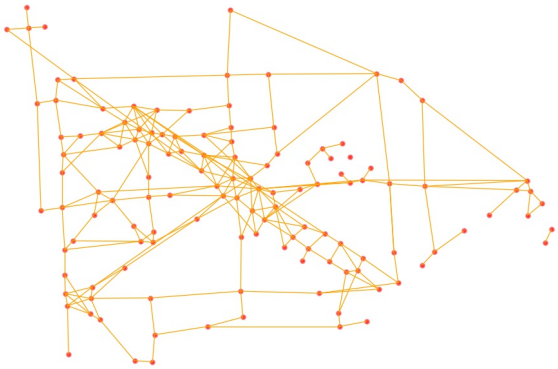
(c) Agrupamiento de líneas y eliminación de líneas redundantes.



(d) Simplificación de líneas.



(e) Superposición final de puntos y líneas de interés con el dibujo original.



(f) Los puntos y las líneas detectadas constituyen el grafo visual del dibujo original.

Fig. 10: *[Parte 2/2, ver Fig. 9]* Ejemplo de transformación de uno de los dibujos de la FCR y creación del grafo visual correspondiente. El producto final del segundo algoritmo es un conjunto de puntos de interés (en rojo) y un conjunto de líneas de interés (en naranja).

de interés y además los detecta *fuera* de los trazos, por eso es importante primero dilatarlos y después ahuecarlos. Así mismo los trazos dilatados sin ahuecar son necesarios para poder calulcar las líneas de interés que representan los trazos del sujeto.

La Fig. 10 muestra un ejemplo del funcionamiento de este algoritmo. Tomando como entrada una de las imágenes limpias producidas en la etapa anterior, el algoritmo es capaz de producir un conjunto de puntos y líneas de interés que capturan razonablemente bien la estructura del dibujo original. Como es puede observar en el ejemplo, el algoritmo no es absolutamente perfecto. Se detectan algunas líneas que no representan trazos reales en el dibujo y, al mismo tiempo se omiten ciertas líneas que sí son trazos reales. Además, todavía existen *demasiados* puntos de interés y todavía existe cierta redundancia en el conjunto de líneas de interés. Este algoritmo depende de un conjunto de parámetros que he ajustado empíricamente para que el resultado sea razonablemente bueno para la mayoría de los dibujos.

4.1.3 Creación de grafos

El tercer y último algoritmo es responsable de transformar el conjunto de puntos y líneas de interés de la etapa anterior en un único grafo que capture las propiedades más importantes del dibujo original. Puesto que no tenemos la información de cómo el sujeto dibujó los trazos de la FCR he considerado que el grafo va a ser no dirigido. En una primera instancia la construcción natural del grafo se podría hacer considerando que cada punto de interés (los puntos rojos de la figura Fig. 10f) puede ser un nodo y que dos nodos estarán conectados por un arco si existe una línea entre los dos puntos correspondientes. Sin embargo, en este tipo de figuras geométricas (y particularmente en la FCR) los trazos del dibujo tienen más importancia que los vértices ya que definen mejor la calidad del dibujo. Dado que la mayoría de la información del dibujo viene dado por las líneas de interés y en menor medida por los puntos, voy a considerar que cada línea de interés va a representar un único nodo en el grafo[6]. Dos líneas que compartan uno de sus extremos entre sí serán consideradas vecinas y se conectarán sus correspondientes nodos en el grafo por un arco. De esta manera el *grafo visual* (Fig. 11a) del dibujo es transformado en un *grafo matemático* (Fig. 11b) de manera que los nodos visuales son arcos matemáticos y los arcos visuales se transforman en nodos matemáticos.

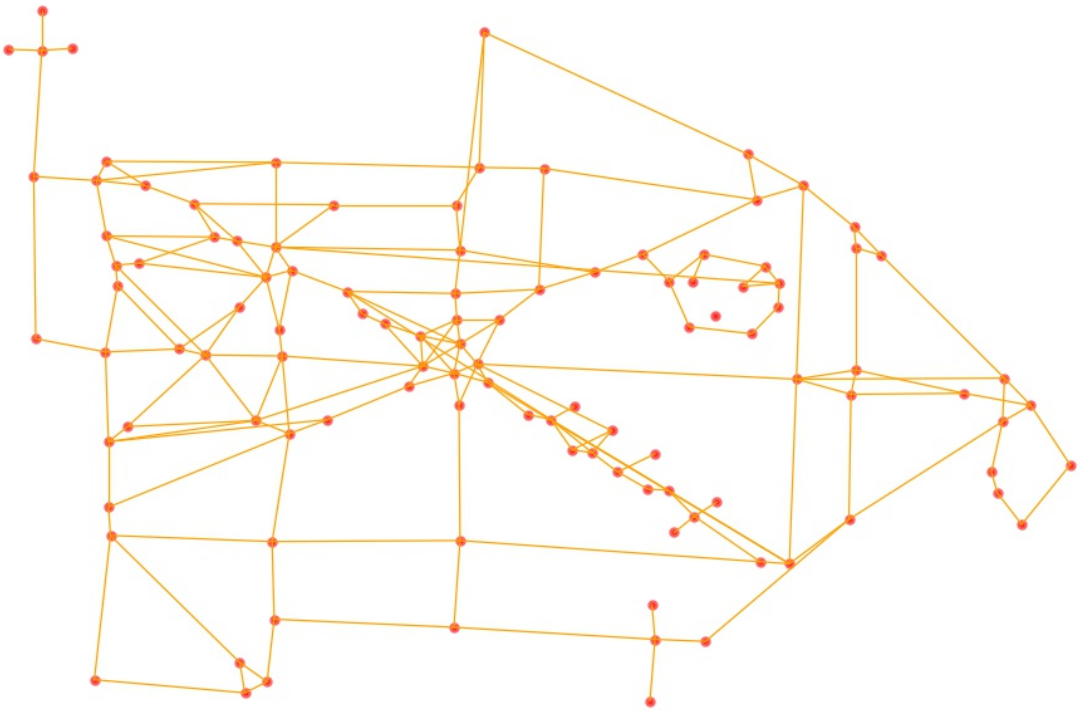
Para conseguir que los nodos de dos grafos sean *comparables* entre sí en primer lugar el algoritmo normaliza las coordenadas de cada punto y cada línea. Para ello, en primer lugar

[6]Recordemos que el Modelo propuesto en este trabajo (sección 3.2) hace uso de una red neuronal que toma como entrada un grafo en el que únicamente los nodos contienen información y que los arcos sólo sirven para unir nodos, pero no continenen información suplementaria.

se identifican los valores extremos (mínimo y máximo) de todas las coordenadas existentes (x, y) dado el conjunto de puntos y líneas de un dibujo de la FCR. A continuación se escalan por separado las componentes x e y de cada punto y de cada línea. De esta manera, todas las coordenadas de puntos y líneas se encontrarán dentro del cuadrado unidad y siempre habrá al menos un punto que toque cada uno de sus cuatro bordes. Así mismo, con esta transformación de escala se consigue que todas las posiciones y distancias de las líneas sean relativas a los tamaños característicos del dibujo del sujeto y no a las distancias de la figura escaneada. Además, esta normalización es robusta frente al algoritmo utilizado para generar los puntos y las líneas. Una vez se ha realizado esta transformación, a cada nodo se le asignan las siguientes propiedades:

- Distancia de la línea a la que representa.
- Seno y coseno del ángulo que forma esta línea con la horizontal. Estas dos propiedades capturan la orientación de la línea y son más fácilmente interpretables que el ángulo que ésta forma con la horizontal.
- Las coordenadas (x,y) que definen los dos extremos de la línea (es decir, cuatro valores reales). Las líneas están creadas de forma que el primer extremo siempre está a la izquierda del segundo extremo o por encima del segundo extremo si es una línea vertical.

Cada nodo es, por lo tanto, un objeto matemático definido por 7 variables numéricas reales. Cada arco del grafo es un objeto definido por dos nodos. El conjunto de nodos y de arcos definen completamente al grafo del dibujo.



(a) Grafo *visual*. Los puntos rojos corresponden a las esquinas del dibujo original. Las líneas naranjas corresponden a los trazos del sujeto.

(b) Grafo *matemático*. Los puntos rojos corresponden a los nodos; las líneas naranjas a los arcos.

Fig. 11: Comparación del grafo *visual* y el grafo *matemático* de un dibujo dado. Cada nodo matemático se sitúa en el punto medio de cada línea visual. Estos nodos contienen información de la línea en cuestión (distancia, ángulo, etc). Dos nodos se conectan con un arco si las líneas a las que representan tienen uno de sus extremos en común.

4.2 Entrenamiento de las redes neuronales

En la sección 3.3.1 definimos dos objetivos, por lo que he realizado dos entrenamientos de la red neuronal independientes. Para cada uno de ellos he entrenado la red neuronal con el optimizador *Adam* durante 50 épocas con una tasa de aprendizaje que varía linealmente desde el valor 10^{-4} en la primera época hasta el valor 10^{-5} en la última época. De nuevo, en ambos casos el conjunto de datos ha sido dividido aleatoriamente y se han utilizado el 80% de los datos disponibles para realizar el entrenamiento y el 20% restante como conjunto de test para evaluar el modelo.

4.2.1 Predicción de la puntuación de la FCR (0-36)

Para la predicción de la puntuación de la FCR vamos a utilizar una red neuronal con la estructura descrita en la sección 3.2 cuya última capa tendrá una única neurona con función de activación ReLU (de Rectified Linear Unit). Esta función de activación no tiene límite superior, por lo que a priori podría predecir puntuaciones superiores a 36 puntos. No obstante, ese límite superior se puede imponer a posteriori. Una alternativa sería la de normalizar las puntuaciones entre 0 y 1 (dividiendo por 36) y entrenar el modelo con una función de activación sigmoide. He intentado entrenar el modelo con una función sigmoide pero el modelo presenta muchas dificultades para predecir valores extremos[7].

La Fig. 12 muestra el proceso de entrenamiento de esta red neuronal y la evolución de la raíz cuadrada de la métrica mse (descrita en la sección 3.3.2) evaluada sobre el conjunto de datos de entrenamiento y sobre el conjunto de datos de test. Tal y como se observa 50 épocas han sido altamente suficientes para que el modelo converja. Como es lógico, la métrica es ligeramente peor cuando se evalúa sobre el conjunto de datos de test. El *mse* alcanzado para el conjunto de test es de 47.57, equivalente a $\sqrt{47.57} \approx 6.90$ puntos de diferencia de media.

La Fig. 13 muestra las predicciones del modelo entrenado evaluando los conjuntos de datos de entrenamiento y test por separado. Se trata de una figura con dos imágenes complementarias. A la izquierda se muestran los gráficos de dispersión para comparar las predicciones del modelo y compararlas con las puntuaciones reales. A la derecha se muestra un mapa de calor que muestra el número de dibujos para cada pareja de puntuaciones *real-predicha*. En general se observa que el modelo obtiene muy buenos resultados evaluando

[7] Una inicialización aleatoria de la red neuronal hace que la función sigmoide prediga puntuaciones alrededor de 0.5 (equivalentes a 18 puntos). Tras el proceso de entrenamiento dicha media se desplaza para predecir valores entre los 20 y los 25 puntos, pero la sigmoide no consigue predecir valoes muy altos y casi ningún valor por debajo de los 18 puntos.

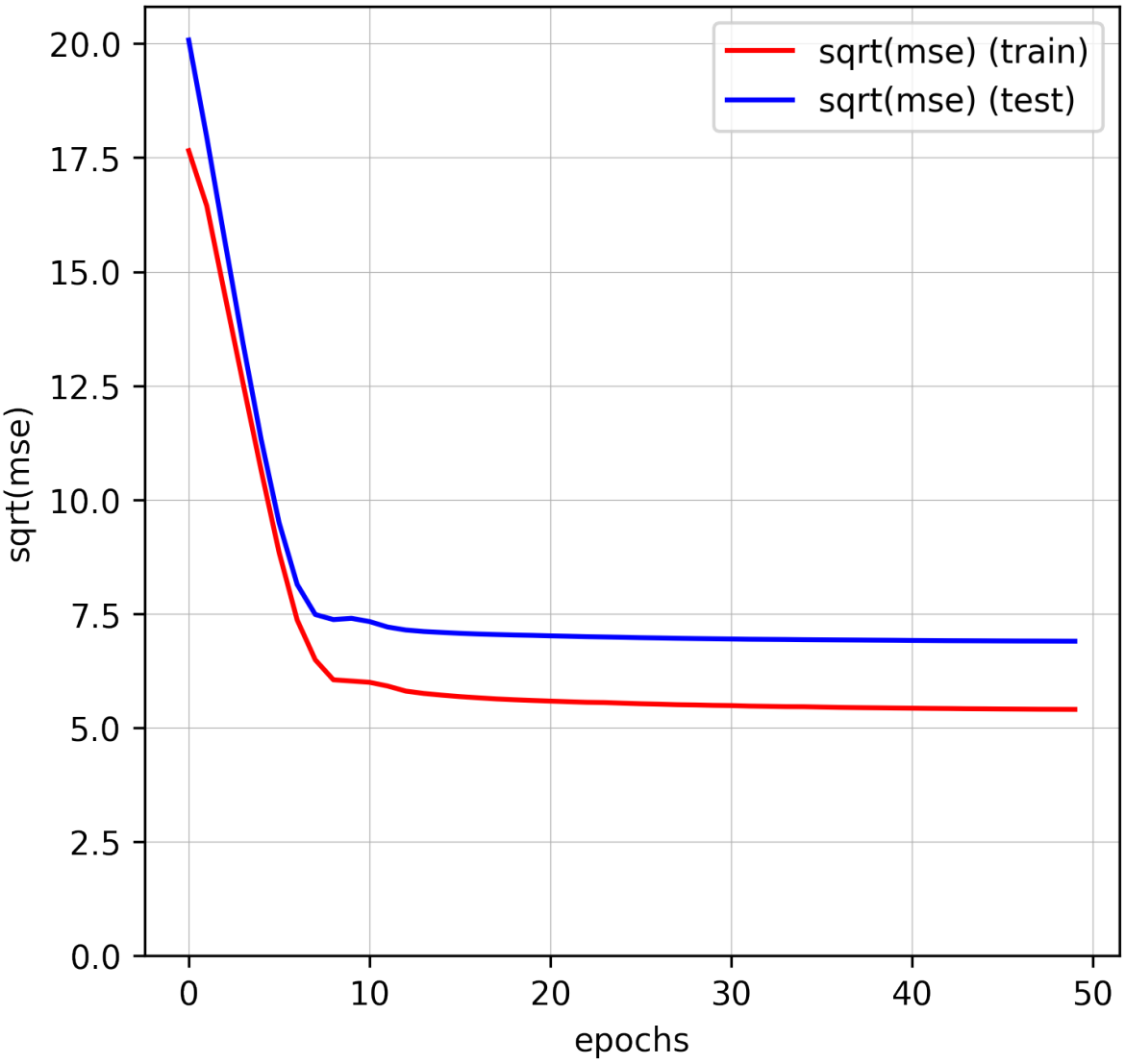
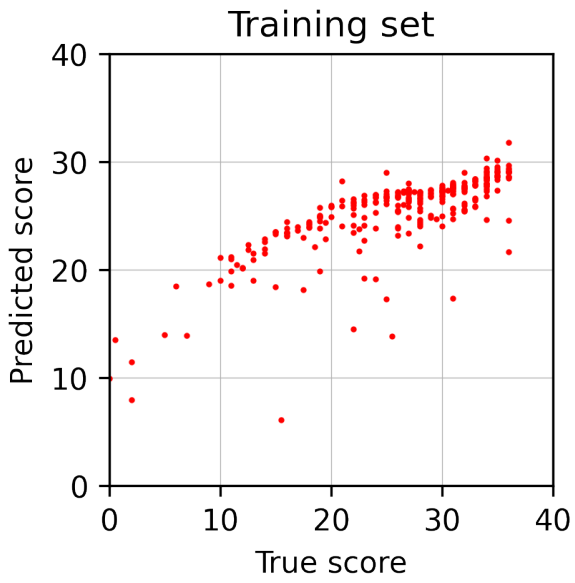


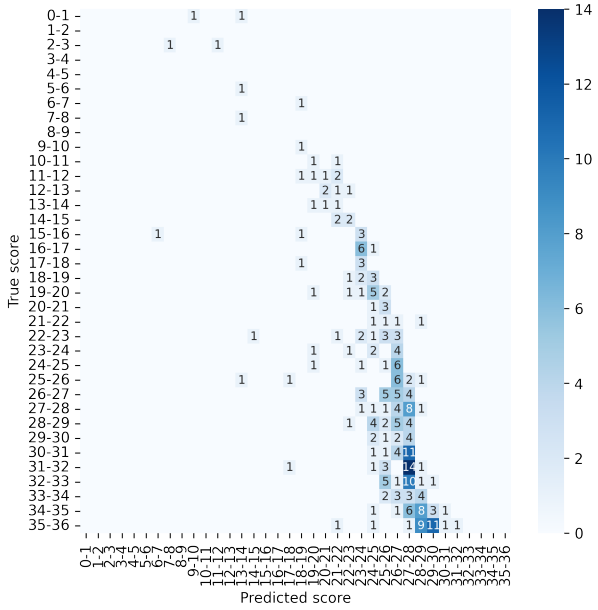
Fig. 12: Proceso de entrenamiento del modelo de predicción de la puntuación de la FCR. Evolución de la raíz cuadrada de la métrica mse con respecto a las épocas de entrenamiento. Evaluación sobre el conjunto de entrenamiento en rojo y sobre el conjunto de test en azul.

dibujos con puntuaciones entre los 20 y los 30 puntos pero presenta dificultades para evaluar dibujos con puntuaciones bajas o muy altas.

Las distribuciones de frecuencia de las puntuaciones del modelo se muestran en la Fig. 14. Estas distribuciones confirman el sesgo del modelo hacia puntuaciones en el rango de los 20-30 puntos. Modulo puntuaciones extremas, estas distribuciones de puntuaciones son similares a la distribución mostrada en la Fig. 4c. Este modelo tiene muchas dificultades

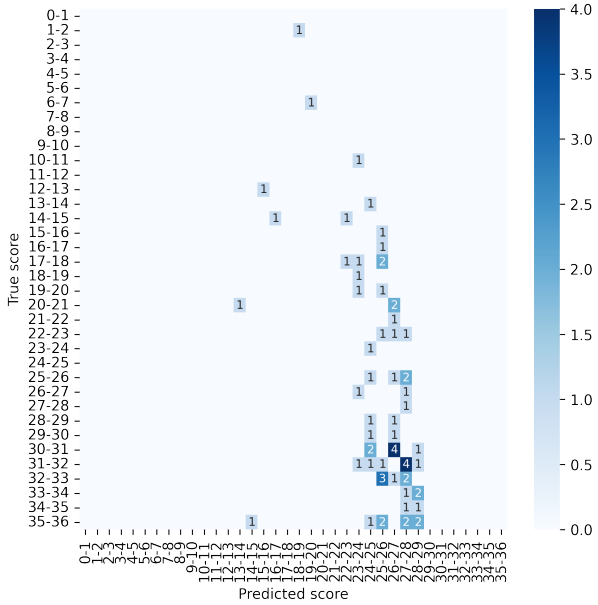


(a) Predicción de la puntuación (eje Y) *vs* puntuación original (eje X) sobre el conjunto de datos utilizado para el entrenamiento.



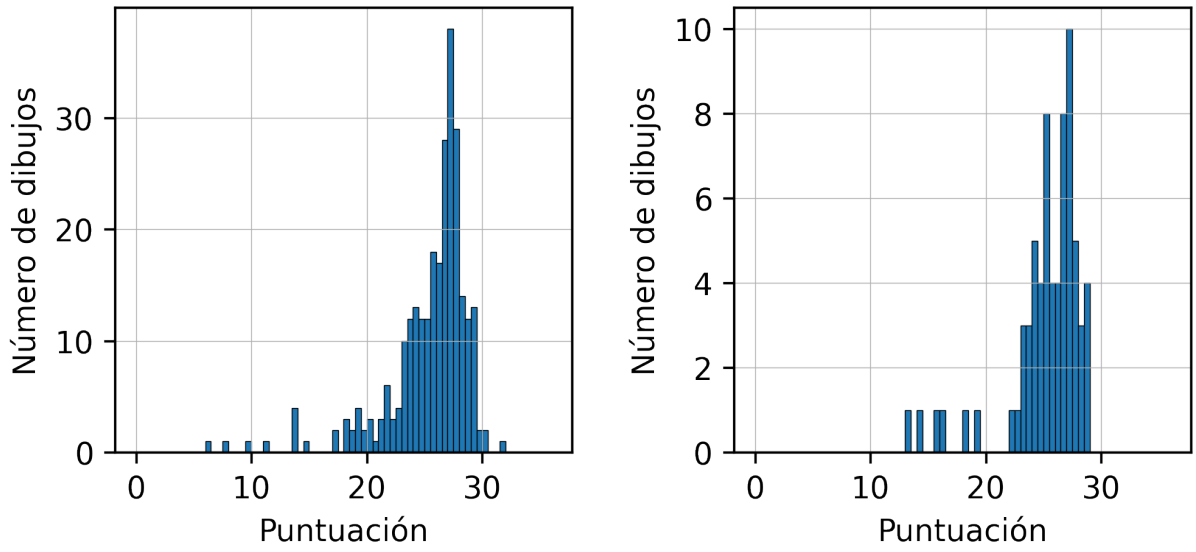
(b) Predicción de la puntuación (eje X) *vs* puntuación original (eje Y) sobre el conjunto de datos utilizado para el entrenamiento.

(c) Predicción de la puntuación (eje Y) *vs* puntuación original (eje X) sobre el conjunto de datos de test.



(d) Predicción de la puntuación (eje X) *vs* puntuación original (eje Y) sobre el conjunto de datos de test.

Fig. 13: Predicciones de las puntuaciones del modelo sobre el conjunto de datos de entrenamiento (arriba) y test (abajo).



(a) Puntuaciones sobre el conjunto de datos de entrenamiento.

(b) Puntuaciones sobre el conjunto de datos de test.

Fig. 14: Distribución de las puntuaciones otorgadas por el modelo.

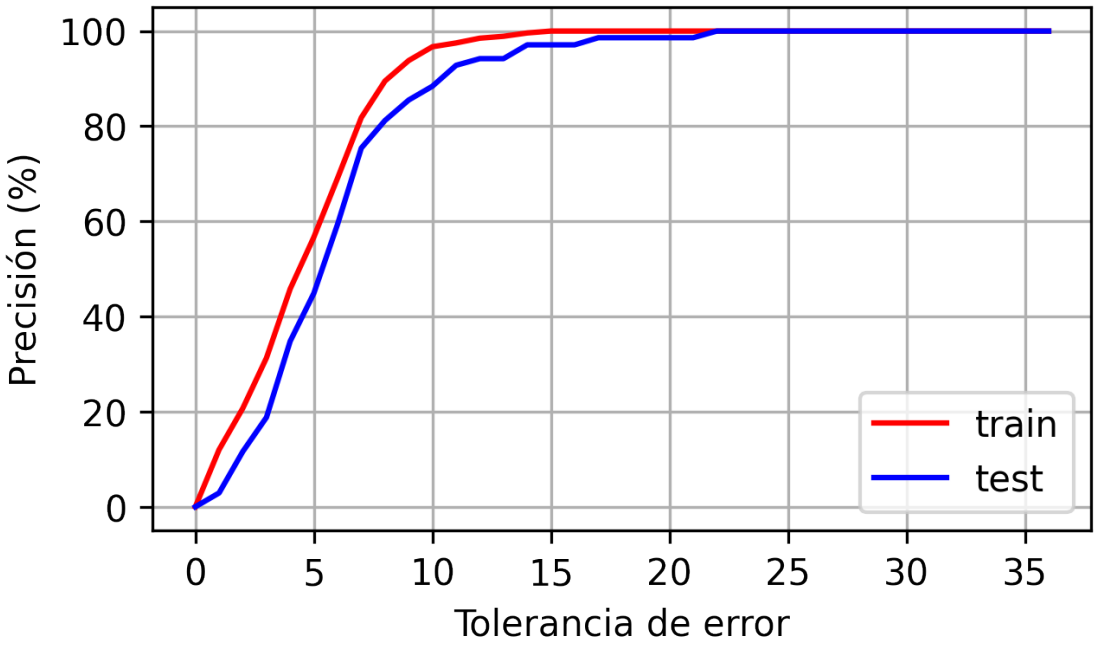


Fig. 15: Evolución de la precisión del modelo en función de la tolerancia del error.

τ	Precisión (entrenamiento)	Precisión (test)
0	0	0
1	12	3
2	21	12
3	31	19
4	46	35
5	57	45
6	69	59
7	82	75
8	90	81
9	94	86
10	97	88
11	98	93
12	99	94
13	99	94
14	100	97
15		97
16		97
17		99
18		99
19		99
20		99
21		99
22+		100

Tabla 1: Precisión (en porcentage) del modelo de regresión en función de la tolerancia de error τ.

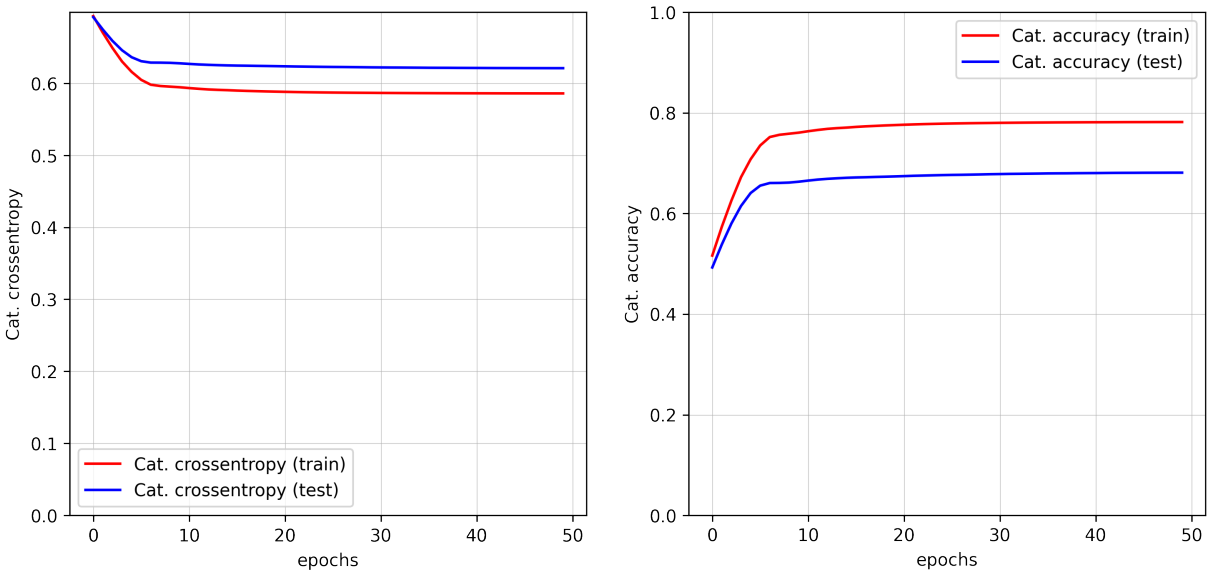
en predecir puntuaciones medias o bajas e incluso puntuaciones muy altas. El modelo va a atribuir en la mayoría de los casos una puntuación de entre 20 y 30 puntos.

Por último, la Fig. 15 y la Tabla 1 muestran la evolución de la precisión del modelo en función de la tolerancia de error τ (este concepto de error y tolerancia fue introducido en la sección 3.3.2). Cuando la tolerancia τ es cero, la precisión del modelo es cero tanto en el conjunto de entrenamiento como en el conjunto de test. Esto se debe a que ningún dibujo está *perfectamente* puntuado y el error entre la puntuación real y la puntuación de la red neuronal siempre es mayor que cero. Recordemos que la puntuación real es un número semientero, mientras que la puntuación de la red neuronal es un número real. Por lo tanto es lógico que ambas puntuaciones no coincidan nunca. Sin embargo si introducimos una pequeña tolerancia de 3 puntos de error observamos que la precisión aumenta y el 18.8% de los dibujos del set de test estarían bien puntuados. Es lógico tambien que la precisión siempre aumente con la

tolerancia del error. Tanto para el set de entrenamiento como para el set de test observamos que la precisión del modelo aumenta linealmente hasta una tolerancia de 6 puntos, para la que se consigue una precisión del 69.1% en el conjunto de entrenamiento y del 59.4% en el conjunto de test. Esta precisión ya es relativamente alta para una tolerancia relativamente baja. A partir de este momento la precisión comienza a acercarse al *plateau* del 100% de precisión. Se necesita una tolerancia de 8 puntos para superar una precisión del 80% en el conjunto de test y una tolerancia de 11 puntos para superar el 92%. La precisión del 100% se alcanza cuando la tolerancia es de 15 puntos en el set de entrenamiento y de 22 puntos en el set de test, que son márgenes de error muy elevados. Esto se debe principalmente a la dificultad de la red para predecir puntuaciones muy bajas.

4.2.2 Predicción de la clasificación del sujeto (sano, DCL)

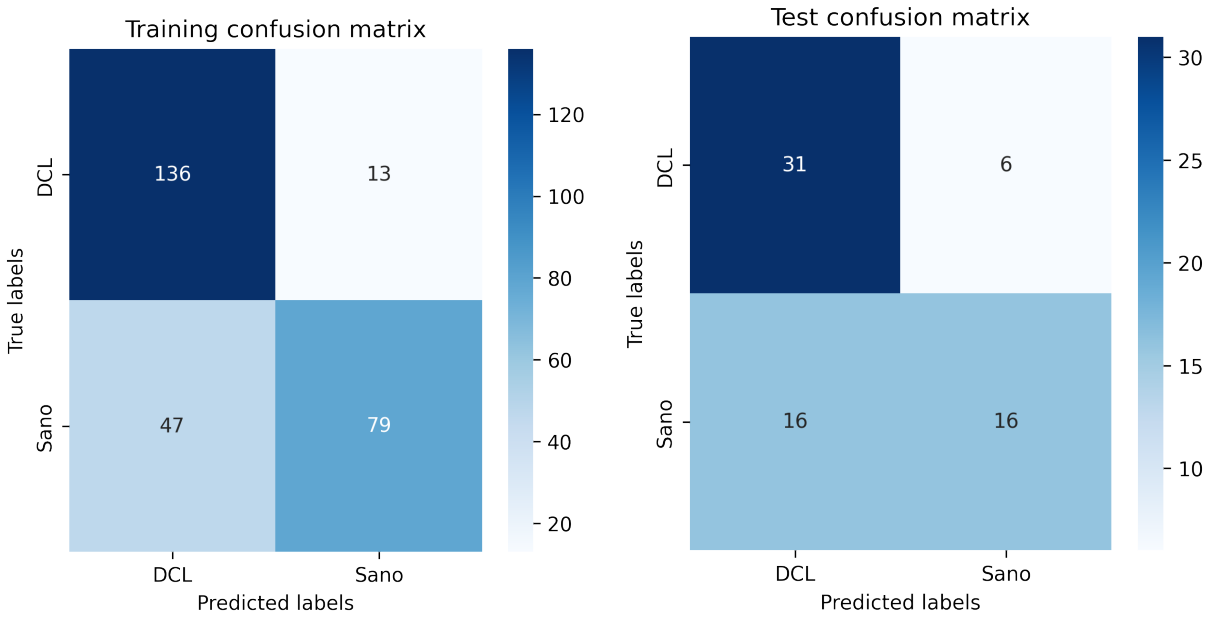
De nuevo, en este caso la última capa de la red neuronal con la estructura descrita en la sección 3.2 tendrá dos neuronas con función de activación *softmax*; cada una de estas neuronas ofrecerá la probabilidad de uno de los dos posibles diagnósticos del paciente en los que se han agrupado los tres tipos de DCL.



(a) Evolución de la función de pérdida cce.

(b) Evolución de la precisión (categorical accuracy).

Fig. 16: Proceso de entrenamiento del modelo de predicción del diagnóstico. Evolución de la función de pérdida cce y la precisión con respecto a las épocas de entrenamiento. Evaluación sobre el conjunto de entrenamiento en rojo y sobre el conjunto de test en azul.



(a) Datos usados para entrenar el modelo.

(b) Datos de test.

Fig. 17: Matriz de confusión de las predicciones del modelo sobre el conjunto de datos de entrenamiento y de test.

La Fig. 16 muestra el proceso de entrenamiento de la red neuronal. De nuevo se observa que el proceso de entrenamiento converge como es debido y que 50 épocas son suficientes. Como es de esperar, la función de pérdida del conjunto de datos utilizados en el entrenamiento es ligeramente inferior a la función de pérdida evaluada sobre el conjunto de test. Así mismo, la precisión en el conjunto de entrenamiento es superior a la precisión evaluada sobre el conjunto de test. La función de pérdida al principio del entrenamiento está cerca del valor 0.69. Esto se debe a que la red neuronal ha sido inicializada aleatoriamente y por lo tanto la salida aleatoria de las dos neuronas de la última capa oscila alrededor de 0.5, y por lo tanto el valor esperado de la función de pérdida es

$$cce_{esperado} \approx -(0 \cdot log(0.5) + 1 \cdot log(0.5)) = log(2) \approx 0.693$$

Así mismo la precisión inicial oscila alrededor de 0.5 ya que los dibujos son clasificados de forma aleatoria, y la mitad de los dibujos estarán bien clasificados y la otra mitad mal. A medida que el proceso de entrenamiento converge, la precisión sobre el conjunto de entrenamiento alcanza el 78.2% y la precisión sobre el conjunto de test el 68.1%.

La Fig. 17 muestra las matrices de confusión del set de entrenamiento y del set de test. Tal y como se puede observar las predicciones del modelo en ambos casos son relativamente

razonables. Sobre el conjunto de entrenamiento 136 de los 149 dibujos de pacientes con DCL han sido bien clasificados, al igual que 79 de los 126 dibujos de pacientes sanos. Esto se traduce en una precisión del $\frac{136+79}{136+79+13+47} \approx 78.2\%$. Sobre el conjunto de test 31 de los 37 dibujos de pacientes con DCL han sido bien clasificados, pero solo la mitad de los 32 dibujos de pacientes sanos han sido clasificados como sanos. Esto se traduce en una precisión del $\frac{31+16}{31+16+6+16} \approx 68.1\%$.

5 Conclusiones

En este trabajo se han presentado una serie de algoritmos para transformar imágenes geométricas (notablemente la FCR) en grafos y una estructura de redes neuronales capaz de tomar dichos grafos como entrada y realizar tareas de regresión y clasificación. Se ha presentado en detalle todos los pasos de los algoritmos generados y se ha justificado el valor de sus parámetros numéricos que definen su comportamiento. Así mismo se han presentado los resultados sobre la capacidad de las redes neuronales para puntuar automáticamente la FCR y para diagnosticar a los sujetos que la dibujaron.

En este trabajo se han considerado ambas tareas por separado y se ha realizado un entrenamiento independiente. Una alternativa y posible mejora podría haber sido la de generar dos capas de salida, una para la puntuación del dibujo con una neurona y otra para la clasificación del sujeto con dos neuronas, y entrenar una única red para realizar ambas tareas simultáneamente.

Otra potencial mejora a nivel práctico sería la modificación de las métricas para evaluar las funciones de pérdida de la red neuronal. En este trabajo se han tratado las clases *sano* y *DCL* democráticamente y ambas tenían el mismo peso. No obstante es posible que en la realidad sea mucho más grave catalogar un paciente como sano si este está enfermo (y por lo tanto necesita ayuda) que catalogar un paciente sano como enfermo, ya que en este caso simplemente se realizarían más pruebas para detectar el falso positivo y no habría repercusiones graves para el paciente.

Por último cabe destacar que tal y como se explicó en la sección 3.1, el diagnóstico de un sujeto depende de la evolución de sus dibujos, del rendimiento de otros sujetos que compartan elementos sociodemográficos y del rendimiento en otras pruebas que no tienen nada que ver con la FCR. Por lo tanto, los dibujos *aislados* no tienen por qué contener la información suficiente como para diagnosticar al paciente y es posible que el modelo no consiga aprender de los datos como es debido; hay pacientes sanos que realizan dibujos con muy baja puntuación y pacientes con DCL que realizan dibujos muy precisos.

Referencias

[1] O. Hijano Cubelos, T. Balezeau and J. Guerin, *The champollion project: Automatic structuration of clinical features from medical records*, in *Artificial Intelligence in Medicine: 19th International Conference on Artificial Intelligence in Medicine, AIME 2021, Virtual Event, June 15–18, 2021, Proceedings*, (Berlin, Heidelberg), p. 452–456, Springer-Verlag, 2021. DOI.

[2] S. Rabinovici-Cohen, T. Tlusty, A. Abutbul, K. Antila, O. Hijano Cubelos, X. Fernandez et al., *Radiomics for predicting response to neoadjuvant chemotherapy treatment in breast cancer*, in *Medical Imaging 2020: Imaging Informatics for Healthcare, Research, and Applications* (P.-H. Chen and T. M. Deserno, eds.), vol. 11318, pp. 333 – 341, International Society for Optics and Photonics, SPIE, 2020. DOI.

[3] S. Rabinovici-Cohen, A. Abutbul, X. Fernández, O. Hijano Cubelos, S. Perek and T. Tlusty, *Multimodal prediction of breast cancer relapse prior to neoadjuvant chemotherapy treatment*, in *Predictive Intelligence in Medicine: Third International Workshop, PRIME 2020, Held in Conjunction with MICCAI 2020, Lima, Peru, October 8, 2020, Proceedings*, (Berlin, Heidelberg), p. 188–199, Springer-Verlag, 2020. DOI.

[4] J. López Miquel and G. Martí Agustí, *Mini-Examen Cognoscitivo (MEC)*, *Revista Espanola de Medicina Legal* (2011) .

[5] J. Peña-Casanova, *Programa integrado de exploración neuropsicológica - Test Barcelona*, *"Revista de Logopedia, Foniatria y Audiologia"* (1991) .

[6] M. D. Lezak, D. B. Howieson, D. W. Loring, J. H. Hannay and J. S. Fischer, *Neuropsychological Assessment. Oxford University Press, New York* (2004) .

[7] S. Rosenblum and G. Luria, *Applying a Handwriting Measurement Model for Capturing Cognitive Load Implications Through Complex Figure Drawing*, *Cognitive Computation* (2016) .

[8] D. R. Coates, J. Wagemans and B. Sayim, *Diagnosing the periphery: Using the Rey-Osterrieth Complex Figure drawing test to characterize peripheral visual function*, *i-Perception* (2017) .

[9] S. García-Herranz, M. C. Díaz-Mardomingo and H. Peraita, *Neuropsychological predictors of conversion to probable Alzheimer disease in elderly with mild cognitive impairment*, *Journal of Neuropsychology* (2016) .

[10] R. O. Canham, S. L. Smith and A. M. Tyrrell, *Automated scoring of a neuropsychological test: The Rey Osterrieth complex figure*, in *Conference Proceedings of the EUROMICRO*, 2000. DOI.

[11] P. A. Osterrieth, *Le test de copie d'une figure complexe; contribution à l'étude de la perception et de la mémoire.*, *Archives de Psychologie* (1944) .

[12] J. Liberman, W. Stewart, O. Seines and B. Gordon, *Rater agreement for the Rey-Osterrieth Complex Figure Test*, *Journal of Clinical Psychology* (1994) .

[13] H. C. Shin, H. R. Roth, M. Gao, L. Lu, Z. Xu, I. Nogues et al., *Deep Convolutional Neural Networks for Computer-Aided Detection: CNN Architectures, Dataset Characteristics and Transfer Learning*, *IEEE Transactions on Medical Imaging* (2016) , [1602.03409].

[14] R. R. Zeina and J. Ramel, *International master of research in computer science : Computer aided decision support graph for pattern recognition*, 2013.

[15] J. Liu, M. Li, Q. Liu, H. Lu and S. Ma, *Image annotation via graph learning*, *Pattern Recognition* (2009) .

[16] P. F. Felzenszwalb and D. P. Huttenlocher, *Efficient graph-based image segmentation*, *International Journal of Computer Vision* (2004) .

[17] F. Feng, W. Li, Q. Du and B. Zhang, *Dimensionality reduction of hyperspectral image with graph-based discriminant analysis considering spectral similarity*, *Remote Sensing* (2017) .

[18] B. Peng, L. Zhang and D. Zhang, *A survey of graph theoretical approaches to image segmentation*, *Pattern Recognition* (2013) .

[19] Roberto Cerrillo Ayuso, *Evaluación automática de test neuropsicológicos utilizando técnicas de graph matching*, *UNED: Trabajo Fin de Máster* .

[20] S. B. Kotsiantis, *Supervised machine learning: A review of classification techniques*, 2007. 10.31449/inf.v31i3.148.

[21] Eladio Estella Nonay, *Diagnóstico automático de la figura compleja de rey mediante redes siamesas*, *UNED: Trabajo Fin de Máster* (2020) .

[22] M. Rincón Zamorano, *An open dataset for automatic drawing analysis of figures included in neuropsychological tests for assessment and diagnosis of mild cognitive impairment*, *(To be published)* .

[23] L. Martin, B. Muller, P. J. Ortiz Suárez, Y. Dupont, L. Romary, É. de la Clergerie et al., *CamemBERT: a Tasty French Language Model*, 2020. 1911.03894. DOI.

[24] T. N. Kipf and M. Welling, *Semi-supervised classification with graph convolutional networks*, in *5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings*, 2017. 1609.02907.

[25] M. Zhang, Z. Cui, M. Neumann and Y. Chen, *An end-to-end deep learning architecture for graph classification*, in *32nd AAAI Conference on Artificial Intelligence, AAAI 2018*, 2018.

[26] M. Benedet and M. Alejandre, *TAVEC: test de aprendizaje verbal España-Complutense.* 2014.

[27] M. Ester, H.-P. Kriegel, J. Sander and X. Xu, *A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise*, in *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining*, 1996.

[28] C. Harris and M. Stephens, *A combined edge and corner detector*, *Proc 4th Alvey Vision Conference* (1988) .

Otros trabajos

En las secciones anteriores he descrito la integralidad de mi Trabajo de Fin de Máster desarrollado con mi tutor Mariano Rincón Zamorano. En esta última sección adicional quiero describir brevemente dos trabajos que he hecho en paralelo al TFM sobre aplicaciones de la Inteligencia Artificial y que han resultado en cuatro publicaciones científicas y un software que hoy en día se encuentra en producción en el mayor centro de tratamiento contra el cáncer de mama de Europa. Estos trabajos son parte de mi producción profesional y no están directamente relacionados con el máster o con la UNED.

Predicción de la respuesta al tratamiento neoadjuvante de cánceres de mama

En este estudio combinamos imágenes de resonancia magnética con otros datos clínicos (tipo de cáncer, resultados de biopsias, etc) para intentar predecir la respuesta al tratamiento neoadjuvante en algunos cánceres de mama y para calcular la probabilidad de aparición de metástasis en el futuro, entre otros. La Fig. 18 muestra los resúmenes y algunas figuras de dos de las tres publicaciones resultantes (la tercera está en proceso de publicación). Estas aplicaciones son muy útiles como apoyo a los clínicos y a los radiólogos que evalúan la imágenes manualmente y como generadores de *alertas* en los casos más severos.

Análisis de texto para la estructuración de datos a partir de informes clínicos

Uno de los problemas a los que se enfrentan la mayoría de los hospitales hoy en día es que la mayoría de los informes clínicos escritos por los doctores se encuentran en forma de texto libre digitalizado, pero no existen bases de datos que recojan los datos de forma estructurada. En este proyecto (llamado Champollion en honor a Jean-Francois Champollion por descifrar la escritura jeroglífica) presentamos una aplicación de IA para el análisis automático de textos y la estructuración automática de datos. La Fig. 19 muestra algunas capturas de pantalla de la publicación resultante. En este caso el producto final no sólo es una publicación para mostrar lo que se puede hacer, sino que el sistema está implementado realmente y funciona en producción a día de hoy en el Instituto Curie en Francia, y es altamente utilizado por los clínicos y los investigadores de dicha institución para llevar a cabo sus tareas de tratamiento e investigación del cáncer.

Multimodal Prediction of Breast Cancer Relapse Prior to Neoadjuvant Chemotherapy Treatment

Simona Rabinovici-Cohen[1](✉), Ami Abutbul[1], Xosé M. Fernández[2], Oliver Hijano Cubelos[2], Shaked Perek[1], and Tal Tlusty[1]

[1] IBM Research – Haifa, Mount Carmel, 3498825 Haifa, Israel
simona@il.ibm.com
[2] Institut Curie, 26 Rue d'Ulm, 75005 Paris, France

Abstract. Neoadjuvant chemotherapy (NAC) is one of the treatment options for women diagnosed with breast cancer, in which chemotherapy is administered prior to surgery. In current clinical practice, it is not possible to predict whether the patient is likely to encounter a relapse after treatment and have the breast cancer reoccur in the same place. If this outcome could be predicted prior to the start of NAC, it could inform therapeutic options. We explore the use of multi-modal imaging and clinical features to predict the risk of relapse following NAC treatment. We performed a retrospective study on a cohort of 1738 patients who were administered with NAC. Of these patients, 567 patients also had magnetic resonance imaging (MRI) taken before the treatment started. We analyzed the data using deep learning and traditional machine learning algorithms to increase the set of discriminating features and create effective models. Our results demonstrate the ability to predict relapse prior to NAC treatment initiation, using each modality alone. We then show the possible improvement achieved by combining MRI and clinical data, as measured by the AUC, sensitivity, and specificity. When evaluated on holdout data, the overall combined model achieved 0.735 AUC and 0.438 specificity at a sensitivity operation point of 0.95. This means that almost every patient encountering relapse will also be correctly classified by our model, enabling the reassessment of this treatment prior to its start. Additionally, the same model was able to correctly predict in advance 44% of the patients that would not encounter relapse.

(a) Título y resumen del primer artículo [3].

Radiomics for predicting response to neoadjuvant chemotherapy treatment in breast cancer

Simona Rabinovici-Cohen[a], Tal Tlusty[a], Ami Abutbul[a], Kari Antila[c], Xosé Fernandez[b], Beatriz Grandal Rejo[b], Efrat Hexter[a], Oliver Hijano Cubelos[b], Abed Khateeb[a], Juha Pajula[c], Shaked Perek[a]
[a]IBM Research – Haifa, Mount Carmel, Haifa 3498825, Israel
[b]Institut Curie, 26 Rue d'Ulm, 75005 Paris, France
[c]VTT Technical Research Centre, Vuorimiehentie 3, Espoo, Finland

ABSTRACT

Women who are diagnosed with breast cancer are referred to Neoadjuvant Chemotherapy Treatment (NACT) before surgery when treatment guidelines indicate that. Achieving complete response in this treatment is correlated with improved overall survival compared with those experiencing a partial or no response at all. In this paper, we explore multi modal clinical and radiomics metrics including quantitative features from medical imaging, to assess in advance complete response to NACT. Our dataset consists of a cohort from Institut Curie with 1383 patients; from which 528 patients have mammogram imaging. We analyze the data via image processing, machine learning and deep learning algorithms to increase the set of discriminating features and create effective models. Our results show ability to classify the data in this problem settings, using the clinical data. We then show the possible improvement we may achieve in combining clinical and mammogram data measured by the AUC, sensitivity and specificity. We show that for our cohort the overall model achieves sensitivity 0.954 while keeping good specificity of 0.222. This means that almost all patients that achieved pathologic complete response will also be correctly classified by our model. At the same time, for 22% of the patients, the model could correctly predict in advance that they won't achieve pathologic complete response, enabling them to reassess in advance this treatment. We also describe our system architecture that includes the Biomedical Framework, a platform to create configurable reusable pipelines and expose them as micro-services on-premise or in-the-cloud.

(b) Título y resumen del segundo artículo [2].

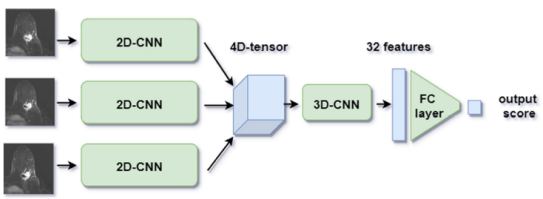


Fig. 1. MRI CNN Architecture. Three adjacent MRI slices (pre-significant, significant, post-significant) form the input to three 2D-CNN that have the same weights. The features are aggregated into a 3D-CNN followed by an average global pooling layer and a fully connected layer that outputs the probability of a patient having a relapse.

(c) Estructura de la red neuronal para analizar resonancias magnéticas.

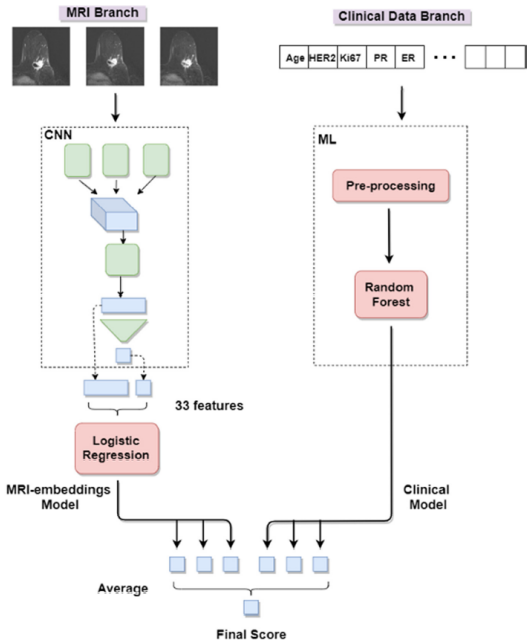


Fig. 2. Ensemble Model Architecture. (left) MRI model branch with three slices as input, (right) clinical model branch with 26 clinical features as input, and (bottom) merge of the two branches into one ensemble. The final score is the average of three scores from MRI-embeddings model variations and three scores from clinical model variations.

(d) Combinación de imágenes con datos clínicos.

Fig. 18: Predicción de la respuesta al tratamiento neoadjuvante de cánceres de mama.

The Champollion Project: Automatic Structuration of Clinical Features from Medical Records

O. Hijano Cubelos[1], T. Balezeau[1], and J. Guerin[1]

[1] Institut Curie, 26, rue d'Ulm, 75005 Paris, France
oliver.hijano-cubelos@curie.fr

Abstract. Cancer is one of the leading causes of mortality worldwide and as populations age, the burden is growing. Treating increasing numbers of patients enables us to gather detailed medical records. Databases with exhaustive, high quality structured data are thus an essential resource for cancer researchers and provide invaluable information to clinicians whenever they need to treat their patients. In addition, these databases fuel our data strategy as the cornerstone of our digital healthcare ecosystem and they provide crucial support for the development of Artificial Intelligence-related projects. Feeding such databases and registries requires manual curation to ensure their quality over time. Finding alternatives to manual structuration is essential because around 80% of the relevant clinical information is contained in open text and it is costly to maintain teams of curators given the growing volumes of data generated every year. In this article we describe an Artificial Intelligence system developed at Institut Curie, capable of structuring clinical features from unstructured Electronic Health Records. Our system allows us to structure clinical data with reduced manual labor and with accuracy comparable to that of expert clinicians, empowering our data ecosystem and improving the support we can give to clinicians and researchers.

(a) Título y resumen del artículo [1].

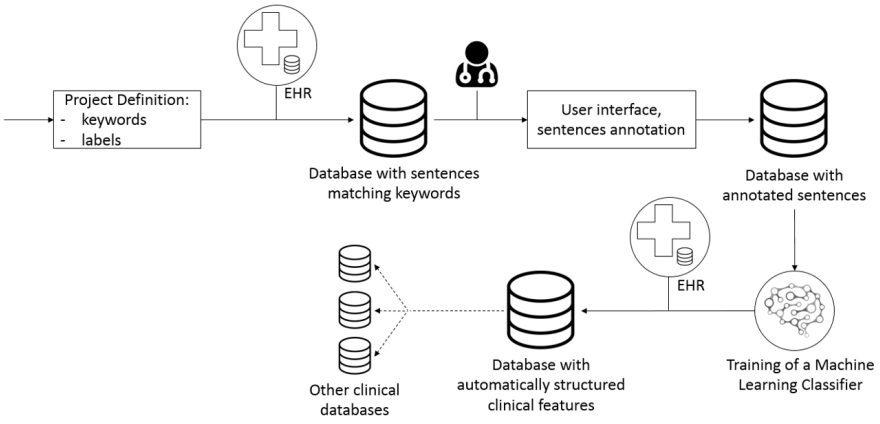


Fig. 1. General overview of the Champollion pipeline. A project is defined by a list of keywords and a list of labels. A random sample of sentences is extracted from the whole corpus of Electronic Health Records (EHR) and stored in a database. A web user interface enables clinicians to annotate sentences manually using a set of labels previously defined. Annotated sentences are stored in another database and used as a Training Dataset to optimize classifier models. Trained models are then used on the whole corpus of text to classify new sentences and store clinical features in a clinical database. Finally, extracted features can be compared and validated with other databases when available.

(b) Representación de las etapas implementadas en el desarrollo del proyecto.

Fig. 19: Análisis de texto para la estructuración de datos a partir de informes clínicos.